A Variable-Radix Digit-Serial Architecture

M.P. Leong and P.H.W. Leong
{mpleong@cse.cuhk.edu.hk, phwl@cse.cuhk.edu.hk}
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong

Abstract

A variable-radix digit-serial design architecture is presented in this paper. This architecture was applied to the computation of the discrete cosine transform (DCT). Results show significantly improved area efficiency over a standard fixed-radix digit-serial implementation.

1 Introduction

Digit-serial architectures are characterized by the property that the entire word of each variable is divided into digits, each digit being processed in a cycle and the entire operation being carried out in multiple cycles. Since the hardware can be time-multiplexed, digit-serial architectures offer reduced area over bit-parallel implementations. In addition, since the number of bits to be processed in a clock cycle are reduced, a higher clock rate can be achieved. Traditional digit-serial architectures have the following properties: the size of a digit-serial operator is strongly dependent on its radix and the area requirement of the operator increases with the radix r; an n-digit variable requires at least n clock cycles for its two's complement representation to be transmitted over the datapath, and n clock cycles are required to complete an operation.

In bit parallel systems, different wordlengths are used to represent each signal in a design. Since the area of an operator depends on its bit width, the area of a multiple wordlength design can be minimized if the wordlengths of its variables and hence the bit widths of operators are optimized according to precision requirements. If two's complement fractions are used, the fractional wordlength f and total (integer and fractional) wordlength w of each variable can be independently determined. Methodologies have been proposed to determine the appropriate wordlengths of variables, such as by simulation [1] or mathematical analysis.

Unfortunately, some problem arise when the above two techniques are applied simultaneously. Firstly, suppose that two operators A and B have a common radix but their bit widths are w_A and w_B respectively. If $w_A \gg w_B$, operator A takes significantly more clock cycles (n_A) to complete its calculation than operator B (n_B) ($n_A \gg n_B$). Unless all operators have a common bit width (as in a fixed wordlength implementation), there will be some idle cycles, and this effect is more pronounced if the bit widths of operators are very different. Secondly, if the radix is fixed, the operator (C) with the largest bit width, w_C, has the largest number of digits, n_C, and the throughput of such a system is limited to n_C. In order to increase throughput while preserving the precision, the radix can be increased so that $n_C = \lceil w_C/r \rceil$ is reduced. However, this increases the area requirements of all operators because the radix is a common parameter. Therefore, the associated area overhead is very large.

2 Variable-Radix Digit-Serial Architecture

To overcome the abovementioned problems, an architecture where each variable can have a different wordlength as well as a different radix is proposed. In this approach, the number of digits n is the same for all variables and operators in the implementation. The format of a variable is parameterized by a triplet (w, f, d), w and f being the total wordlength and the fractional wordlength respectively and d the radix of the variable. In general, $w \geq 0$, f is an integer and $d \geq \lceil w/n \rceil$. When $d > w/n$, sign-extension is carried out on the variable concerned.

As an example, the representation of a variable with precision format $(16, 5, 6)$ and $n = 3$ is

$$\overbrace{\underbrace{b_{10}b_{10}}_{\text{sign extension}}b_{10}b_9b_8b_7}^{\text{digit 2}}\ \overbrace{b_6b_5b_4b_3b_2b_1}^{\text{digit 1}}\ \overbrace{\underbrace{b_0.b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}}_{\text{fractional part}}}^{\text{digit 0}}$$

The subscripts of b refer to their exponents in base 2. For instance, if the most significant bit of digit 1 is one, its partial value is $2^6 = 64$. Two sign-extension bits are inserted to expand the wordlength of the variable from 16 to 18 bits (3 digits × 6 bits).

This variable-radix digit-serial architecture has the following features. Firstly, the number of digits n is common to all variables and operators throughout the system. This ensures all arithmetic operators have the same throughput, hence maximal resource utilization is guaranteed. Secondly, since all the operators have the same number of digits, they take the same number of cycles to complete, have the same throughput and hence no bottleneck can occur. Thirdly, unlike the traditional digit-serial architecture, radices of individual variables can be increased or decreased independently and do not affect those of other variables. The area overhead associated with increasing the wordlength of a variable is hence localized. This is in contrast with the traditional digit-serial

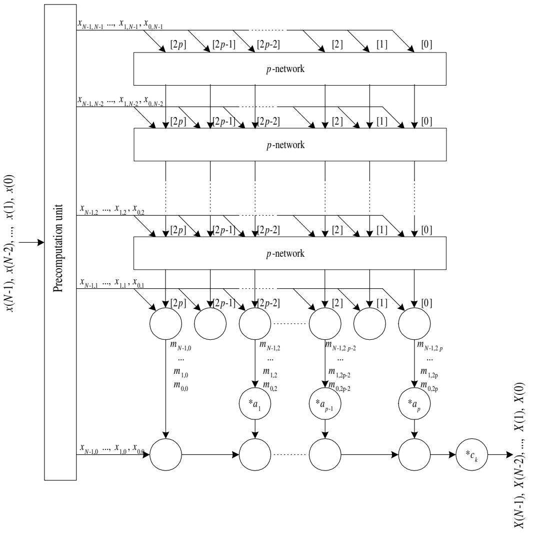
$x(N\text{-}1),\ x(N\text{-}2),\ ...,\ x(1),\ x(0)$
Precomputation unit
$x_{N-1,N-1}$ ..., $x_{1,N-1}$, $x_{0,N-1}$
$x_{N-1,N-2}$ ..., $x_{1,N-2}$, $x_{0,N-2}$
$x_{N-1,2}$ ..., $x_{1,2}$, $x_{0,2}$
$x_{N-1,1}$ ..., $x_{1,1}$, $x_{0,1}$
$x_{N-1,0}$ ..., $x_{1,0}$, $x_{0,0}$
[2p]
[2p-1]
[2p-2]
[2]
[1]
[0]
p-network
p-network
p-network
$m_{N-1,0}$ ... $m_{1,0}$ $m_{0,0}$
$m_{N-1,2}$ ... $m_{1,2}$ $m_{0,2}$
$m_{N-1,2p-2}$ ... $m_{1,2p-2}$ $m_{0,2p-2}$
$m_{N-1,2p}$ ... $m_{1,2p}$ $m_{0,2p}$
$*a_1$
$*a_{p-1}$
$*a_p$
$*c_k$
$X(N\text{-}1),\ X(N\text{-}2),\ ...,\ X(1),\ X(0)$

Figure 1: The systolic array for 1D N-point DCT, with p-network blocks and blank nodes represent additions.

architecture in which a global area overhead is imposed if the throughput of the implementation must be retained. Thus a more cost-effective tradeoff between performance and resource requirements may possibly be achieved.

A possible drawback of the variable-radix architecture is that extra hardware is needed to convert variables from one radix to another whereas in the traditional approach, operands have the same radix.

3 Systolic Computation of the DCT

Liu et. al. [2] proposed an approach to compute the DCT by transforming its computation into a systolic structure involving discrete moments (DM). The bulk of the computation is performed using only additions and constant-cofficient multiplications. The algorithm centers around setting up the relationship between the DCT and DM, in which the N-point DCT, $X(k)$, is equal to

$$c_k\left(x_{k,0}+\sum_{r=0}^{p}a_r m_{k,2r}\right)+R_p,\quad 0\le k\le N-1,$$

where $a_r=\frac{(-1)^r\pi^{2r}}{(2N)^{2r}(2r)!}$ and $m_{k,2r}=\sum_{i=1}^{N-1}x_{k,i}i^{2r}$. $m_{k,2r}$ is the $2r$-order DM and R_p is a Taylor remainder term. As proposed by Liu et. al. [2], to compute $m_{k,2r}$, an architecture known as the p-network which resembles a Pascal's triangle can be used. The p-network consists only of adders and carries out the transformation

$$\begin{aligned}&F_p(1,x,x^2,\ldots,x^{p-1},x^p)\\&=(1,(1+x),(1+x)^2,\ldots,(1+x)^{p-1},(1+x)^p).\end{aligned}$$

The resultant systolic structure is depicted in Figure 1.

In our experiments $N=8$ and $p=3$. The systolic structure thus generated consists of 157 adders and 5 constant-coefficient multipliers. An optimization approach [1] was used to determine the wordlengths and

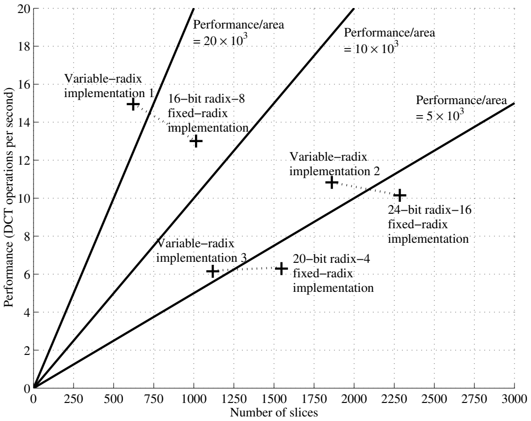

Figure 2: Comparison of performance and area between variable-radix and fixed-radix implementations.

radices of variables such that the area (including the area used by radix conversion described in Section 2) is minimized while the overall output error is within an error bound. The target FPGA device was a Xilinx Virtex XCV1000-6 FPGAs and all the implementations mention below were tested on an Annapolis "Wildstar" platform.

4 Results

Variable-radix DCT implementations were compared with those implemented using a traditional fixed-radix digit-serial architecture. A comparison of performance and area between the two implementations are shown in Figure 2. In this figure, three pairs of implementations were compared (indicated by dotted lines) and three equi-performance to area ratio lines (the larger performance to area ratio, the better the area efficiency) are also shown. In all the above cases, the performance to area ratio of the variable-radix digit-serial implementations were better than fixed-radix ones. An 87% improvement was seen in variable-radix implementation 1, and the two others have 31% and 37% improvements respectively.

5 Conclusion

This paper presented a variable-radix digit-serial architecture which combines the advantages of both digit-serial and multiple wordlength designs. Although this paper only described the application of this architecture to the DCT, it is envisaged that the same methodology can also be applied to many other applications.

References

[1] M. P. Leong, M. Y. Yeung, C. K. Yeung, C. W. Fu, P. A. Heng, and P. H. W. Leong, "Automatic floating to fixed point translation and its application to post-rendering 3D warping," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 240–248, April 1999.

[2] J. G. Liu, H. F. Li, F. H. Y. Chan, and F. K. Lam, "Fast discrete cosine transform via computation of moments," *Journal of VLSI Signal Processing*, vol. 19, pp. 257–268, 1998.