HARDWARE IMPLEMENTATION OF SOBEL-EDGE DETECTION DISTRIBUTED ARITHMETIC DIGITAL FILTER

Sorawat CHIVAPREECHA and Kobchai DEJHAN
Faculty of Engineering and Research Center for Communication and Information Technology
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
Tel. 66-2326-4238, 66-2326-4242, Fax. 66-2326-4554
e-mail : {sorawat, kobchai}@telecom.kmitl.ac.th
Chanchai PIENVIJARNPONG
Geo-Informatics and Space Technology Development Agency (GISTDA)
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand
e-mail : chanchai@gistda.or.th

KEY WORDS: Sobel-Edge Detection, Digital Filter, Distributed Arithmetic, FPGA

ABSTRACT This paper proposes a design of two-dimensional (2-D) digital filter for real-time image processing. The filter structure is based on Distributed Arithmetic (DA), the obtained digital filter will be multiplierless digital filter. In addition, the hardware design uses VHDL and synthesis using FLEX10K Altera FPGA as target technology and use MAX+plusII program for overall development. Results of design are shown the speed performance and used area of FPGA. Finally, Sobel-edge detection images of size 256x256 pixels are shown when compared with simulation the results from Matlab program.

1. INTRODUCTION

At present, the real-time image processing using 2-D digital filter has become rapidly growing in many applications such as geophysical and biomedical image processing. However, the designs of 2-D digital filter need many multiplications for one output sample because there are many non-zero elements in the filter coefficient. The number of multipliers are decreased the speed of processing time since the multiplying stage has a lot of consumption process and uses large area in design. Also, this paper proposes a design of distributed arithmetic structure for 2-D digital filter.

Distributed arithmetic (DA) was proposed by (Peled and Liu, 1974) at first. The DA is a direct method for sum of products operations, partial products can pre-compute by difference equation and stored in look-up table contained in memory, input signals can be used for addressing. The product can be computed by scaling accumulate of partial products from memory, therefore, multipliers don't necessary for this method, the obtained 2-D digital filter is multiplierless. In previous papers (Jaggernauth et al, 1985, Thongplew and Dejhan, 1994), 2-D digital filter was developed without multipliers using standard integrated circuits and memory devices which has large area and power consumption. This paper uses FPGA (Field Programmable Gate Array) in the design. Consequently, it can be obtained the single chip 2-D digital filter and reduces the power consumption.

2. REALIZATION FOR 2-D DISTRIBUTED ARITHMETIC DIGITAL FILTER

Consider, the 2-D digital filter is described by linear difference equation as follow,

$$y(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}x(m-k,n-l)\,h(k,l) \tag{1}$$

where $x(m,n)$ and $y(m,n)$ are the image input and output, respectively when $h(k,l)$ are filter coefficients of mask size 3x3.

Assume, all signals are bounded by ± 1 and define the input signal in two's complement format, B bits accuracy including sign bit by

$$x(m,n)=\sum_{j=1}^{B-1}x_j(m,n)\,2^{-j}-x_0(m,n) \tag{2}$$

where $x_j(m,n)$ are $x(m,n)$ in bit level. Thus, eq. (1) can be rewritten as

$$y(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}h(k,l)\left(\sum_{j=1}^{B-1}x_j(m-k,n-l)\,2^{-j}-x_0(m-k,n-l)\right) \tag{3}$$

Rearrange the summations in eq. (3) yields

$$y(m,n)=\sum_{j=1}^{B-1}\left[\sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)x_j(m-k,n-l)\,2^{-j}\right)\right]-\sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)x_0(m-k,n-l)\right) \tag{4}$$

Define the function $F(\cdot)$ as follow,

$$\begin{aligned}F_j\left(x_j(m,n),x_j(m,n-1),\cdots,x_j(m-2,n-2)\right)\\ =h(0,0)x_j(m,n)+h(0,1)x_j(m,n-1)+\cdots+h(2,2)x_j(m-2,n-2)\end{aligned} \tag{5}$$

Since each $x_j(m-k,n-l)$, where $j = 0,1,\ldots, B-1$ is only 0 or 1, we also pre-compute the values of this function that called partial products stored in memory and using the input signals for addressing. Therefore, the output can be computed by shifting and adding of partial products as in eq. (6).

$$y(m,n)=\sum_{j=1}^{B-1}F_j(\cdot)2^{-j}-F_0(\cdot) \tag{6}$$

This equation can be realized to be hardware by using the distributed arithmetic and can be shown that the multiplier has not used in this structure.

3. HARDWARE ARCHITECTURAL DESIGN

Distributed arithmetic realization of Sobel-edge detection digital filter can be derived from eq. (6). The major hardware consists of image data arrangement part for image size 256x256 pixels and processing part. The structures has many components, as shown in Fig. 1.

The operation can be described by 4 steps as follow,

1. Signal ***lr_img*** will be loaded as input image data $x(m,n)$ 8 bits to buffer, in the same time each pixels in same column will be shifted by PIPO (parallel in parallel out shift register) to generate the data in time index n-1 and n-2, respectively. Delay for 256 steps is used for delay 256 samples to generate data in time index m-1 and m-2, respectively. This function can operate by image data arrangement part, which called frame manager.

2. Signal ***lr*** will be loaded as all of data from frame manager part to bank of PISO (parallel in serial out shift register)
3. Signal ***clk*** will be shifted data in each PISO to bit level, output of each PISO for each time of shifting used for two ROM addressing, output of each ROM will be added to scaled value from ACC using scaling accumulator that control by signal ***s/a***. Result will be loaded to ACC by signal ***lacc***.
4. Repeat step 3, until eighth bit, output of each ROM will be subtracted from scaled value from ACC. Result will be loaded to buffer by using signal ***lr***, then the signal ***clacc*** will clear ACC, and repeat step 1-4, respectively.

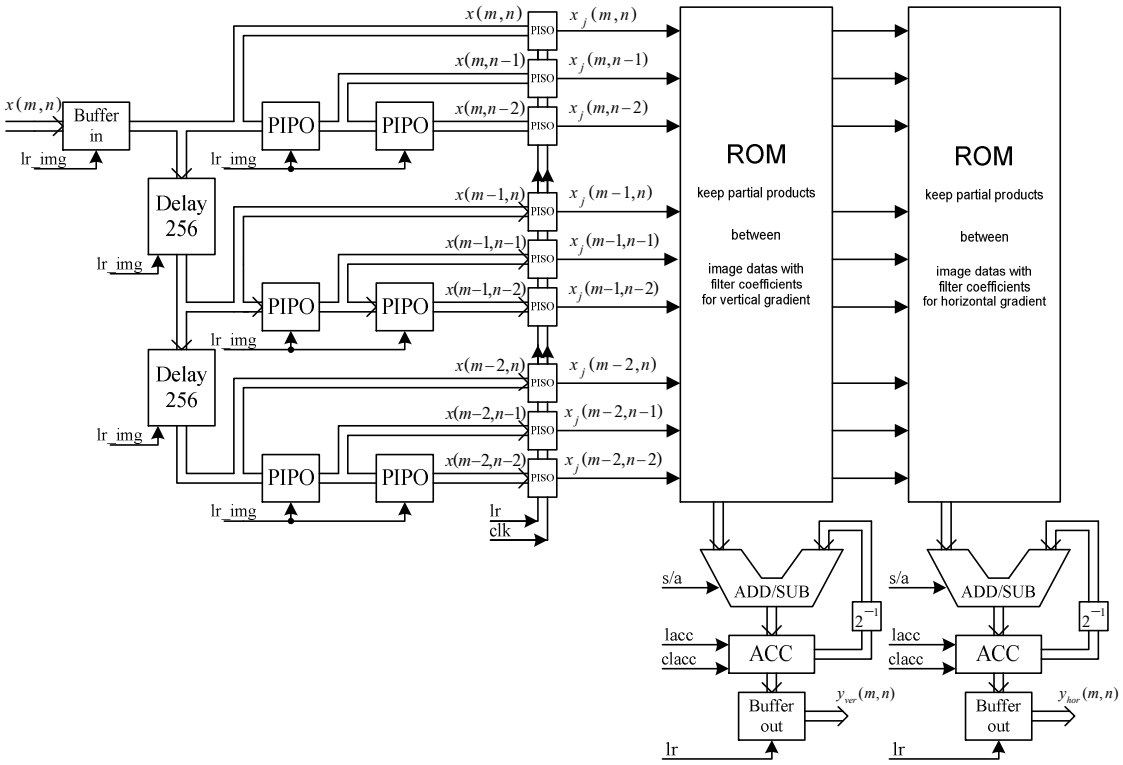

Figure 1 Hardware architecture of Sobel-edge detection Distributed Arithmetic digital filter

All components can be modeled using VHDL which delay for 256 steps obtained from LPM (library of parameterized modules) in MAX+plusII program. Delay for 256 steps can be replaced with LPM_RAM, MAX+plusII compiler automatically implements and suitable portions of this function in EABs (embedded array blocks) in FLEX10K devices, it also does not use logic cells for design. The modified frame manager is shown in Fig. 2.

The timing diagram of control unit is used for the proposed hardware architecture with modified frame manager and can be shown in Fig. 3.

Each ROM keeps the partial products between image data with filter coefficients of 3x3 mask size can be used LPM_ROM for design. Also, it must be used 9 inputs for each ROM addressing. The values content in each ROM and can be found using eq. (5). Filter coefficients are shown in Fig. 4

All possible values containing in ROM are equal to 512 values, each value will be converted to 8 bits two's complement format before stored in ROM as shown in Table 1.

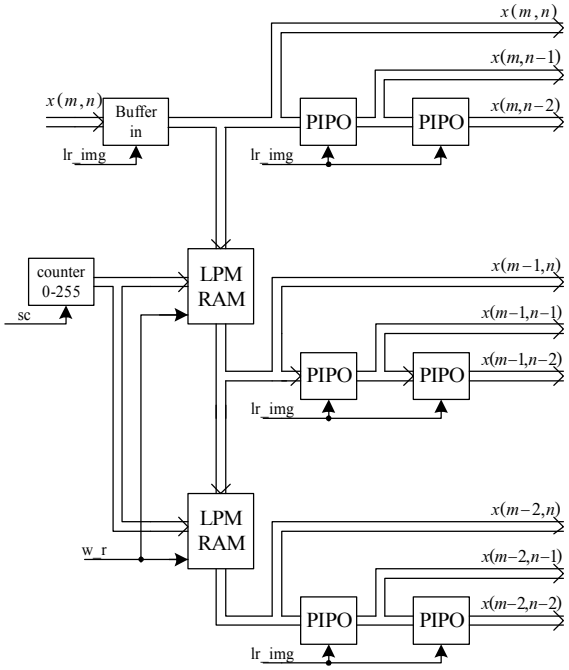

Figure 2 Modified frame manager

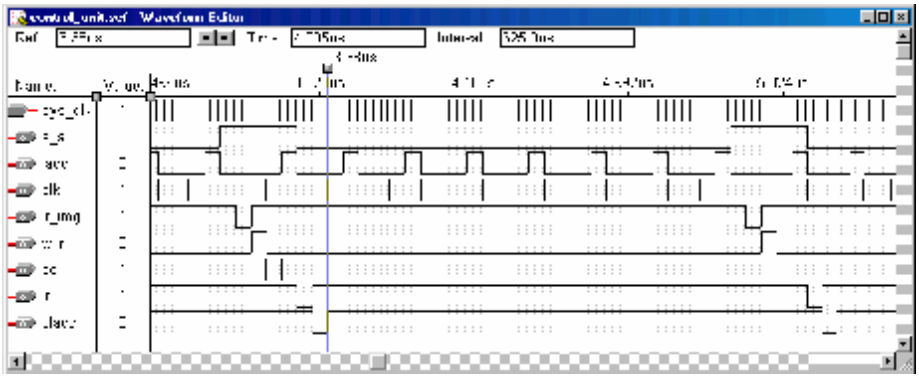

Figure 3 Timing diagram of control unit

$h(0,0)$	$h(0,1)$	$h(0,2)$
$h(1,0)$	$h(1,1)$	$h(1,2)$
$h(2,0)$	$h(2,1)$	$h(2,2)$

Figure 4 Filter coefficients in mask size 3x3

Table 1 Evaluation of values content in each ROM

Addresses									Values content in ROM
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	h(2,2)
0	0	0	0	0	0	0	1	0	h(2,1)
0	0	0	0	0	0	0	1	1	h(2,1)+h(2,2)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	0	0	h(0,0)+h(0,1)+h(0,2)+h(1,0)+h(1,1)+h(1,2)+h(2,0)
1	1	1	1	1	1	1	0	1	h(0,0)+h(0,1)+h(0,2)+h(1,0)+h(1,1)+h(1,2)+h(2,0)+h(2,2)
1	1	1	1	1	1	1	1	0	h(0,0)+h(0,1)+h(0,2)+h(1,0)+h(1,1)+h(1,2)+h(2,0)+h(2,1)
1	1	1	1	1	1	1	1	1	h(0,0)+h(0,1)+h(0,2)+h(1,0)+h(1,1)+h(1,2)+h(2,0)+h(2,1)+h(2,2)

4. RESULTS

An EPF10K20RC240-4 device in FLEX10K device family used for circuit synthesis. From device summary of Sobel-edge detection distributed arithmetic digital filter, 9 input pins used for 8 bits input image data and one for system clock, 16 output pins for 8 bits two output image data, 12288 memory bits used for implementing LPM_ROM and LPM_RAM into EABs because 1 EAB equivalent to 2048 memory bits. Also, total 6 EABs are used. Others component which using VHDL for design will use 280 LCs (logic cells) for implementing. Timing summary shows the maximum frequency of synthesized circuits is about 37 MHz.

In experiment, image data of 256 levels gray scale image can be stimulated from computer. The operation of Sobel-edge detection can be shown in Fig. 5.

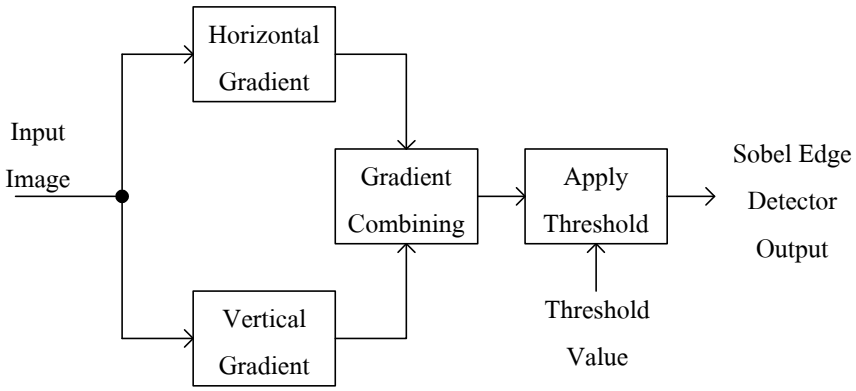

Figure 5 Operation of Sobel-edge detection

Filter coefficients are used for testing as shown in 2 cases as vertical edge detector and horizontal edge detector. The 3x3 mask of both vertical gradient and horizontal gradient operators that use in experiment (Gonzalez and Woods, 2002) is shown in Fig. 6.

1	0	-1
2	0	-2
1	0	-1

(a) Vertical edge detector

1	2	1
0	0	0
-1	-2	-1

(b) Horizontal edge detector

Figure 6 Convolution masks of Sobel-edge detection

Gradient combining obtained from absolute values between vertical gradient output and horizontal gradient output as follow

$$y(m,n) = \sqrt{y_{ver}(m,n)^2 + y_{hor}(m,n)^2} \quad (7)$$

Also, apply threshold to gradient combining output for adjust image results to 2 levels intensity. In Fig. 7 shows the simulation results that obtained from gradient combining and after applying threshold, respectively and Fig. 8 shows the proposed hardware results that obtained from gradient combining and after applying threshold, respectively.

(a) Gradient combining (b) Applying threshold

Figure 7 Sobel-edge detection outputs from Matlab

(a) Gradient combining (b) Applying threshold

Figure 8 Sobel-edge detection outputs from proposed hardware

From Fig. 7 and 8 show that the experimental results, are almost same as simulation results from Matlab program. It can be ensured that the proposed hardware can operate correctly.

5. CONCLUSIONS

From the proposed method in this paper, it can obtain the multiplierless 2-D digital filter using distributed arithmetic realization. The hardware is suitable for implementing on FPGA to obtain single chip 2-D digital filter, high processing speed and reduce power consumption when compared with discrete components in implementation. Consequently, the obtained 2-D digital filter can be used with real-time applications.

References

Peled, A. and Liu, B., 1974. A New Hardware Realization of Digital Filters. IEEE Trans. On ASSP., Vol. ASSP-22, No. 6, pp. 456-462.

Jaggernauth, J., Loui, A. C. P. and Venetsanopoulos, A. N., 1985. Real-Time Image Processing by Distributed Arithmetic Implementation of Two-Dimensional Digital Filters. IEEE Trans. on ASSP., Vol. ASSP-33, No. 6, pp. 1546-1555.

Thongplew, S. and Dejhan, K., 1994. Implementation of 2-D Digital Filter-based Distributed Arithmetics. Ladkrabang Engineering Journal., Vol. 11, No. 1, pp. 1-6.

Gonzalez, R. C. and Woods, R. E., 2002. Digital Image Processing, Prentice-Hall.