Recurrently Decomposable 2-D Convolvers for FPGA-Based Digital Image Processing

Zhao-Bin Ma, Yang Yang, Yun-Xia Liu, and Anil Anthony Bharath

Abstract—Two-dimensional (2-D) convolution is a widely used operation in image processing and computer vision, characterized by intensive computation and frequent memory accesses. Previous efforts to improve the performance of field-programmable gate array (FPGA) convolvers focused on the design of buffering schemes and on minimizing the use of multipliers. A recently proposed recurrently decomposable (RD) filter design method can reduce the computational complexity of 2-D convolutions by splitting the convolution between an image and a large mask into a sequence of convolutions using several smaller masks. This brief explores how to efficiently implement RD based 2-D convolvers using FPGA. Three FPGA architectures are proposed based on RD filters, each with a different buffering scheme. The conclusion is that RD based architectures achieve higher area efficiency than other previously reported state-of-the-art methods, especially for larger convolution masks. An area efficiency metric is also suggested, which allows the most appropriate architecture to be selected.

Index Terms—Field-programmable gate arrays (FPGAs), large convolution mask, recurrently decomposable, two-dimensional (2-D) convolution.

I. INTRODUCTION

TWO-DIMENSIONAL (2-D) convolution is widely used in image analysis and computer vision. It is also increasingly found in the latest generation of deep-learning systems for image categorization [1]. Convolution can be computationally intensive: for a $R \times S$ convolution mask, $R \times S$ multiplications, $R \times S - 1$ additions and $R \times S$ accesses to the input data are required for processing each pixel. In modern computer vision, and deep convolutional networks, dozens to hundreds of convolution masks may be required.

Field-programmable-gate-array (FPGA) represents a good choice of device for performing 2-D convolution, because of the ability to fully exploit the inherent parallelism involved in this spatial operation. Designs for efficient FPGA implementations – achieving, say, a throughput of 1 pixel/clock mainly focus on two aspects: the design of the buffering scheme and improving the convolution kernel module.

This work is supported by the National Natural Science Foundation of China (No. 61203239, 61305015, 61375084, 11474185).

Z. Ma and Y. Yang are with the School of Information Science and Engineering, Shandong University, Jinan 250100, China (e-mail: yyang@sdu.edu.cn).

Y. Liu is with the School of Control Science and Engineering, Shandong University, Jinan 250014, China.

A.A. Bharath is with the Department of Bioengineering, Imperial College London, London, SW7 2AZ, UK.

A good buffering scheme [2-3] can lead to reduced on-chip resources by limiting the number of input buffers based on an acceptable external memory bus bandwidth. Specifically, a full buffering (FB) scheme [2] would have an optimal external memory bus bandwidth of 1 pixel/clock, whilst a single-window partial buffering (SWPB) scheme [2] requires the least amount of on-chip resources; a multi-window partial buffering (MWPB) scheme [3] is a tradeoff between these two extremes.

A diverse set of ideas has been explored to reduce the complexity of the convolution kernel module, because this can account for a significant part of the FPGA resources. Bosi *et al.* [2] devised a 2-D convolver with multiplexed 1-D convolution modules or pixels interlacing strategy. These schemes sacrifice processing speed in order to reduce hardware consumption. A second category of methods replaces the multipliers with alternatives, such as shift-and-accumulation (SA) operations [4], look up table (LUT) [5], and $\log_2$ and inverse- $\log_2$ approximations [6]. However, the total number of multiplications or substitute algorithms (MSAs) remains untouched in these multiplier-less implementations. Improved area efficiency was achieved by combining $\log_2$ and inverse- $\log_2$ approximations with folding operations in [7]. These folding operations rely on symmetry in the convolution masks, but mask symmetries vary greatly, particularly when the goal of convolution is to produce rich feature spaces [1].

Recurrently decomposable (RD) filters [8] provide a feasible way of reducing MSAs for arbitrary 2-D filters. The complexity is reduced at the software level by separating the convolution mask into a series of convolutions using smaller masks. In this brief, the FPGA implementations of RD based 2-D convolvers are studied. Three architectures adopting FB, SWPB and MWPB buffering schemes are presented, and for each, we study bandwidth requirement and area utilization. The performance of the resulting FPGA implementations is compared with other state-of-the-art approaches. A metric that integrates both required bandwidth and resources utilization is proposed for selecting the most appropriate architecture to meet design considerations.

In Section II we briefly describe the RD approach for 2-D filter implementation, the foundation of all convolution design improvements in this brief. The three proposed RD based convolvers are provided in Section III, with the comparisons of FPGA implementations given in Section IV. Finally, Section V concludes this brief.

II. RECURRENTLY DECOMPOSABLE 2-D FILTERS

Conventionally, 2-D filters are defined either as separable or non-separable according to whether or not they can be decomposed into the tensor product of two 1-D filters. In previous work [8], we showed that the computation efficiency of 2-D convolution can be improved if a 2-D filter can be separated into the convolution of two smaller 2-D filters. We termed filters that satisfied this *recurrently decomposable filter*.

Consider the case where a $R \times S$ mask ω is decomposed into the convolution of a $P \times Q$ sized small mask φ , and $J \times K$ sized small mask ψ :

$$\omega_{R \times S} = \varphi_{P \times Q} * \psi_{J \times K}, \quad (1)$$

where $*$ denotes the convolution operation, we have

$$\begin{cases} R = P + J - 1 \\ S = Q + K - 1 \end{cases} \quad 2 \leq P, J \leq R-1, 2 \leq Q, K \leq S-1. \quad (2)$$

Thus for each pixel, the total MSAs saving is $R \times S - (P \times Q + J \times K) = (P-1) \times (K-1) + (Q-1) \times (J-1) - 1$, which is of quadratic order in mask size. For example, if a 17×17 mask can be separated into the convolution of two 9×9 masks, or a 16×2 mask and a 2×16 mask, the number of MSAs is reduced by 127 and 225, respectively. Comparing to the original number of 289 MSAs per pixel, this represents a significant (43.94% and 77.85%, respectively) saving in computation. These two cases correspond to the worst case and the best case in terms of improvement in computational complexity among all possible decompositions.

In real applications, there will be a trade-off between accuracy and efficiency. The decomposition is conducted by solving the unconstrained optimization problem

$$\begin{aligned} \arg \min_{\varphi, \psi} \sum_i \sum_j \|\omega_{i,j} - (\varphi * \psi)_{i,j}\|^2 \\ \text{s.t.} \quad \sum_i \sum_j \|\omega_{i,j} - (\varphi * \psi)_{i,j}\|^2 \leq \varepsilon \quad \forall (i,j) \in R \times S \end{aligned} \quad (3)$$

where $(\cdot)_{i,j}$ denotes the kernel weight and ε is an acceptable upper bound on error tolerance based on the application. Practically speaking, the approximation error between the original kernel ω and the convolution of lower ones φ and ψ (i.e. $\varphi * \psi$) tends to be small, so we can often preserve improved efficiency with negligible effect on numeric performance. Readers are referred to [8] for more discussion on error analysis.

For a given 2-D convolution mask, we can select freely from all possible decompositions that meet the error tolerance requirement. Once the decomposition scheme ($\varphi_{P \times Q}$ and $\psi_{J \times K}$) is determined, we seek an efficient FPGA implementation. Although φ or ψ can sometimes be further decomposed, we limit our discussion in this brief to one-layer decompositions, which is sufficient for illustration.

III. RD BASED ARCHITECTURE DESIGN

In this section, we describe three FPGA architectures of RD based 2-D convolvers, each supported by different buffering

Fig. 1. FB architecture based on recurrent decomposition.

schemes. We provide comparisons of area utilization and bandwidth requirement with conventional schemes.

The RD based architecture consists of a cascade of convolvers that correspond to the smaller sized masks, with buffering between them. In the proposed RD based architectures (See Fig.1-3), Convolver I and II implement $\varphi_{P \times Q}$ and $\psi_{J \times K}$ respectively, where the size constraint in Eq. (2) is satisfied. All discussions are based on the scenario that an input $M \times N$ image is convolved with a $R \times S$ convolution mask without boundary extension. The output image will be $(M - 2 \times \lfloor R/2 \rfloor) \times (N - 2 \times \lfloor S/2 \rfloor)$ in size, where $\lfloor X \rfloor$ denotes the largest integer less than or equal to X .

A. RD Based FB Architecture

In the proposed RD based FB architecture, pixels are fetched from external memories and buffered on-chip in Convolver I as depicted in Fig. 1. A first in first out (FIFO) with depth T is introduced to balance the clock frequency and data width between the external memory bus and the convolver. The FIFO in Convolver II is saved and the output of Convolver I is buffered by a combination of $J-1$ line-buffers whose length is $N - 2 \times \lfloor Q/2 \rfloor - K$ and J sets of register arrays in Convolver II. The $J \times K$ convolution in Convolver II begins when these buffers are filled up.

The proposed RD based architecture inherits the advantages of the FB scheme in that it only requires an external memory bus bandwidth of 1 pixel/clock. Comparing with the conventional FB scheme, which implements the $R \times S$ mask as a whole, there is a reduction of $(P-1) \times (K-1) + (J-1) \times (Q-$

TABLE I
AREA UTILIZATION AND BANDWIDTH REQUIREMENT OF DIFFERENT 2-D CONVOLVER ARCHITECTURES

Scheme	Area Utilization		Bandwidth (pixels/clock)
	Buffered Pixels	MSAs	
FB [2]	$(R-1) \times N + S + T$	$R \times S$	1
FB(RD)	$(R-1) \times N + S + 1 - 2 \times \lfloor Q/2 \rfloor \times (J-1) + T$	$P \times Q + J \times K$	1
SWPB [2]	$R \times (T+S)$	$R \times S$	R
SWPB(RD)	$P \times Q + J \times K + (R+1) \times T$	$P \times Q + J \times K$	$R+2$
MWPB [3]	$(R+S-1) \times (S+T)$	$R \times S$	$(R+S-1)/S$
MWPB(RD)	$(P+Q-1) \times Q + (J+K-1) \times K + (R+S) \times T$	$P \times Q + J \times K$	$(R+S+1)/S$

Fig. 2. Architecture based on SWPB recurrent decomposition.

1) – 1 MSAs: a significant saving in hardware resources. There is also a slight saving of $2 \times \lfloor Q/2 \rfloor \times (J-1) - 1$ in the number of shift registers, which is negligible compared to the total resources used.

B. RD Based SWPB Architecture

The SWPB scheme [2] was proposed to reduce the number of buffers under a certain external memory bus bandwidth requirement. Unlike FB scheme, the $P \times Q$ shift registers in Convolver I directly receive data from FIFOs and then the convolution window moves to the next position with P pixels shifted in. An external memory of at least $(N - 2 \times \lfloor Q/2 \rfloor) \times J$ in size is adopted to store the interim convolution results of Convolver I. These results are fed to a $J \times K$ SWPB convolver (Convolver II) for further processing. On-chip resources reduction is obtained at the price of increased external memory bus bandwidth.

As compared with the conventional SWPB scheme [2], $(P-1) \times (K-1) + (J-1) \times (Q-1) - 1$ MSAs and shift-registers are saved, but one more FIFO is needed. The resources difference

Fig. 3. Architecture based on MWPB recurrent decomposition.

in buffering pixels is negligible comparing with those consumed by the whole convolution module. Considering also the external temporary memory, the bandwidth demanded by the RD based architecture is raised by 2 pixels/clock.

C. RD Based MWPB Architecture

To balance the on-chip resources utilization and external memory bus bandwidth, the MWPB scheme [3] was proposed to reuse data that have already been stored in internal buffers. Fig.3 illustrates the proposed RD based MWPB architecture as a cascade of two MWPB convolvers. An external memory of at least $(M - 2 \times \lfloor P/2 \rfloor) \times K$ in size is utilized to store interim results, and as the processing is in column-major scan format, column to row scan format adaptation could be performed to ensure data consistency whenever necessary [3].

Compared with the conventional MWPB scheme, an RD based architecture reduces the number of required registers by $(K-1) \times (P+Q-2) + (Q-1) \times (J+K-2) - 1$. There is one FIFO increase and a saving of $(P-1) \times (K-1) + (J-1) \times (Q-1) - 1$ MSAs. The external memory bus bandwidth that the RD

TABLE II
HARDWARE RESOURCES UTILIZATION FOR DIFFERENT 17×17 CONVOLUTION ARCHITECTURES

Scheme	FB [2]	FB(RD)		SWPB [2]	SWPB(RD)		MWPB [3]	MWPB(RD)	
		Worst Case	Best Case		Worst Case	Best Case		Worst Case	Best Case
Input Buffer	6634	6714	6784	1016	1076	1051	2073	2228	2207
Convolution Kernel	12764	7192	2800	12764	7192	2800	13046	7472	3012
Output Buffer	43	75	76	42	76	76	49	82	81
Complete Design	19439	13976	9659	13821	8338	3935	15160	9762	5289
Decrease in Convolution Kernel		43.65%	78.06%		43.65%	78.06%		42.71%	74.91%
Decrease of Complete Design		28.10%	50.31%		39.67%	71.53%		35.61%	65.11%

based MWPB architecture needed is $(R + S + 1) / S$ pixels/clock.

Table I summarizes the main features of different architectures, when the throughput is all fixed to 1 pixel/clock. Area utilization is measured by the total number of pixels the convolver needs to buffer and the amount of MSAs that the 2-D convolution kernel module demands, where T represents the depth of the FIFO. External memory bus bandwidth is given in terms of pixels/clock.

As shown in Table I, the RD based architectures consume almost the same resources (depending on depth value T of FIFOs) in buffered pixels, but have great benefits in saving MSAs for all three buffering schemes. In terms of bandwidth requirement, the RD based full buffering architecture maintains the 1 pixel/clock external memory bus bandwidth. The $2/S$ slight increase hardly influences the bandwidth of the MWPB architecture (around 2 pixels/clock, if a square shaped convolution mask assumed), especially for larger mask sizes. For the RD based SWPB architecture, the required bandwidth is raised by 2 pixels/clock. A trade-off must be made between the MSAs saved and the increased bandwidth for small mask sizes (See more discussion on architecture selection in section IV).

IV. PERFORMANCE ANALYSIS

A. FPGA Implementation

In this experiment, we consider an input image of size 1024×1024 which is to be convolved with a 17×17 mask. The FPGA implementations were realized on a low-cost XILINX Spartan-3 XC3S4000 FPGA using the Integrated Software Environment (ISE) 14.7. The data-width of each pixel in the input image was 8 bits, while the FIFO's depth was set to 8. MSAs were realized with multipliers. All of the FIFOs, line-buffers, multipliers and adders were instantiated from the IP cores provided by ISE. Specifically, the FIFOs were implemented from shift registers, while line-buffers were implemented using RAM-based shift registers. All these designs were written using Verilog and were synthesized using XILINX Synthesis Tool (XST). Table II summarizes the hardware resources utilization in terms of slices for the input buffer module, the convolution kernel module and the output buffer module, respectively. The resources decrease in the kernel module and the proposed RD architectures are also given in percentage with respect to conventional ones, respectively. A throughput of 1 pixel/clock is assumed for all architectures.

TABLE III
BANDWIDTH AND MSAs REQUIREMENT OF DIFFERENT ARCHITECTURES

SCHEME		Bandwidth Required	Number of MSAs
FB [2]		1	$R \times S$
FB (RD)	Best Case	1	$(R - 1) \times 2 + 2 \times (S - 1)$
	Worst Case	1	$[R/2] \times [S/2] + (R + 1 - [R/2]) \times (S + 1 - [S/2])$
SWPB [2]		R	$R \times S$
SWPB (RD)	Best Case	$R + 2$	$(R - 1) \times 2 + 2 \times (S - 1)$
	Worst Case	$R + 2$	$[R/2] \times [S/2] + (R + 1 - [R/2]) \times (S + 1 - [S/2])$
MWPB [3]		$(R + S - 1) / S$	$R \times S$
MWPB (RD)	Best Case	$(R + S + 1) / S$	$(R - 1) \times 2 + 2 \times (S - 1)$
	Worst Case	$(R + S + 1) / S$	$[R/2] \times [S/2] + (R + 1 - [R/2]) \times (S + 1 - [S/2])$

A significant reduction in hardware utilization can be seen in Table II, for all recurrently decomposable architectures. For the best and worst cases of the three RD based architectures, the real implementations show reductions in hardware resources required by convolution kernel module to around 42% and 78%, respectively. They are very close to their theoretical values (43.94% and 77.85%), due to the amount of control logic used for the pipeline controls in different buffering schemes. The savings in resources can be up to 71.53% for SWPB scheme; the benefit was smaller for FB scheme (28.10% for the worst case). This is because the input buffers account for a large part in the complete design. Note that the benefits apply to all MSAs types (e.g. SA [4], LUT [5], and $\log_2$ and inverse- $\log_2$ approximations [6]), as their computational complexities all reduce proportionally with the number of MSAs.

B. Architecture Selection

Bandwidth requirement is a key factor in architecture selection for buffering schemes. Generally speaking, FB and MWPB architectures are widely adopted, while the high bandwidth requirement of SWPB architecture may not be satisfied in practical applications for a low-cost FPGA implementation.

To complete the architecture selection, we have to determine whether a recurrent decomposition should be performed for a specific design point. Prior performance metrics [3][9] do not incorporate the hardware consumption that is required by the convolution kernel module, which is a key issue that we address, and which is reflected through proposing a performance metric.

Fig. 4. Bar diagram comparing the area efficiency metric of recurrently decomposable 2-D architectures with conventional convolvers for varying convolution mask sizes. The lower the bar, the more efficient.

Suppose the maximum computing efficiency is achieved (throughput is fixed to 1 pixel/clock), we may define the optimum architecture as that which minimizes the area efficiency metric, ξ :

$$\xi = BW_{in} \times N_{MSA}, \quad (4)$$

where BW_{in} denotes the required external memory bus bandwidth, and N_{MSA} denotes the total number of MSAs involved in the convolution kernel module. We consider N_{MSA} as a good indicator of resources consumption of a candidate architecture, without going into further details about MSAs type and different component libraries. The lower the metric value ξ , the more efficient the architecture is for a particular design point.

Table III provides a comparison of the required bandwidth and number of MSAs for six architectures. $[X]$ denotes the smallest integer greater than or equal to X . The results of best and worst cases can be used as upper and lower bounds for area efficiency improvements in hardware resources. Fig. 4 visualizes the proposed metric ξ on log scale with respect to varying convolution mask sizes for different buffering

schemes, where three mask sizes are selected as representatives for convolution masks in three different size ranges; these correspond to small (3×3 , 5×5 , 7×7), medium (13×13 , 17×17 , 21×21) and large (33×33 , 41×41 , 49×49) mask sizes. As shown in Fig. 4, RD based FB architectures are always more efficient for all mask sizes. While for partial buffering architectures, the mask sizes should be greater than 5×5 . This is agreement with the qualitative analysis in Section III. Furthermore, the improvements in area efficiency brought by recurrent decompositions become more obvious with the increase of mask size, for all buffering schemes.

V. CONCLUSION

In this brief, we presented three RD based FPGA architectures adopting FB, SWPB, and MWPB schemes, respectively. By recurrent decomposition, resources consumed by the convolution kernel module are greatly reduced. FPGA implementations demonstrate better area efficiency, especially for large convolution kernels. An area efficiency metric is suggested to guide architecture selection.

We have restricted this brief to considering the buffering schemes of basic architectures of RD based convolvers. For substitute algorithms that improve the efficiency of convolution kernel modules [4-7], the computational complexity and hardware consumption will shrink in proportion with the number of MSAs by applying a recurrent decomposition. Thus, these schemes can benefit from RD architectures.

ACKNOWLEDGEMENT

The authors would like to thank the associate editor and the reviewers for helpful comments that greatly improved this brief.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G.E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, May. 2015.
- [2] B. Bosi, G. Bois, and Y. Savaria, "Reconfigurable pipelined 2-D convolvers for fast digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 3, pp. 299-308, Sep. 1999.
- [3] H. Zhang, M. Xia and G. Hu, "A multiwindow partial buffering scheme for FPGA-based 2-D convolvers," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 54, no. 2, pp. 200-204, Feb. 2007.
- [4] M. H. Sunwoo and S. K. Oh, "A multiplierless 2-D convolver chip for real-time image processing," *J. VLSI Signal Processing Systems for Signal Image and Video Technology*, vol. 38, no. 1, pp. 63-71, Aug. 2004.
- [5] F. J. Toledo-Moreo, J. J. Martínez-Alvarez, J. Garrigós-Guerrero, and J.M. Ferrández-Vicente, "FPGA-based architecture for the real-time computation of 2-D convolution with large kernel size," *J. Systems Architecture*, vol. 58, no. 8, pp. 277-285, Sep. 2012.
- [6] M. Z. Zhang, H. T. Ngo, and V. K. Asari, "Multiplier-less VLSI architecture for real-time computation of multi-dimensional convolution," *Microprocessors and Microsystems*, vol. 31, no. 1, pp. 25-37, Feb. 2007.
- [7] M. Z. Zhang, and V. K. Asari, "An efficient multiplier-less architecture for 2-D convolution with quadrant symmetric kernels," *The VLSI Journal*, vol. 40, no. 4, pp. 490-502, Jul. 2007.
- [8] Y. X. Liu, Y. Yang and A. A. Bharath, "Recurrently decomposable 2-D filters," *J. Computational Information System*, vol. 11, no. 5, pp. 1773-1779, Mar. 2015.
- [9] F. Cardells-Tormo and P.-L. Molinet, "Area-efficient 2-D shift-variant convolvers for FPGA-based digital image processing," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 53, no. 2, pp. 105-109, Feb. 2006.