International Journal of Research

Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

Vlsi Design for Carry-Protect Formatted Data

[1]SK. MUJEEB, [2]B.BALA KRISHNA, [3] Dr.B.S.R.MURTHY

[1] M.Tech Student,DEPT OF ELECTRONICS & COMMUNICATION ENGINEERING. GANDHI ACADEMY OF TECHNICAL EDUCATION, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206

[2]M.TECH,Assistant Professor, HOD,DEPT OF ELECTRONICS & COMMUNICATION ENGINEERING. GANDHI ACADEMY OF TECHNICAL EDUCATION, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206

[4]Phd,Professor, & Principal. GANDHI ACADEMY OF TECHNICAL EDUCATION, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206

ABSTRACT:

However, research activities have proven the arithmetic optimizations at greater abstraction levels compared to structural circuit one considerably effect on the datapath performance. CS representation continues to be broadly accustomed to design fast arithmetic circuits because of its natural benefit of getting rid of the big carry-propagation chains. Hardware acceleration continues to be demonstrated a very promising implementation technique for digital signal processing (DSP) domain. Instead of adopting a monolithic application-specific integrated circuit design approach, within this brief, we present a manuscript accelerator architecture composed of flexible computational models that offer the execution of a big group of operation templates present in DSP popcorn kernels. Extensive experimental evaluations reveal that the suggested accelerator architecture provides average gains as high as 61.91% in area-delay product and 54.43% in energy consumption in comparison using the condition-of-art flexible datapaths. We differentiate from previous creates flexible accelerators by enabling computations to become strongly carried out with carry-save (CS) formatted data. Advanced arithmetic design concepts, i.e., recoding techniques, are employed enabling CS optimizations to become carried out inside a bigger scope compared to previous approaches.

Keywords: Carry-save (CS) form, datapath synthesis, flexible accelerator, operation chaining.

I. INTRODUCTION

The incorporation of heterogeneity through specialized hardware accelerators improves

performance and reduces energy consumption. Modern embedded systems target high-finish application domain names needing efficient implementations of computationally intensive digital signal processing (DSP) functions. Many scientists have suggested using domain-specific coarse-grained reconfigurable accelerators, to be able to increase ASICs' versatility without considerably compromising their performance [1]. Although application-specific integrated circuits (ASICs) make up the ideal acceleration solution when it comes to performance and power, their inflexibility results in elevated plastic complexity, as multiple instantiated ASICs are necessary to accelerate various popcorn kernels. High-performance flexible datapaths happen to be suggested to efficiently map primitive or chained procedures based in the initial data-flow graph (DFG) of the kernel. The templates of complex chained procedures are generally removed from the kernel's DFG or specified by a predefined behavior template library. Design choices around the accelerator's datapath highly impact its efficiency. Existing creates coarse-grained recon-figural datapaths mainly exploit architecture-level optimizations, e.g., elevated instruction-level parallelism (ILP). The domain-specific architecture generation calculations and vary the kind and quantity of computation models achieving a personalized design structure. Flexible architectures were suggested exploiting ILP and operation chaining [2]. Lately, Ansaloni et al. adopted aggressive operation chaining to allow the computation of entire sub expressions using multiple ALUs with heterogeneous arithmetic features. These reconfigurable architectures exclude arithmetic optimizations throughout the architectural synthesis and think about them limited to the interior circuit structure of primitive components, e.g., adders, throughout the logic synthesis. However, research activities have proven the arithmetic optimizations at greater abstraction levels compared to structural circuit one considerably effect on the datapath performance. Timing-driven optimizations according to carry-save (CS) arithmetic were carried out in the publish-Register Transfer Level (RTL) design stage. Common sub expression elimination in CS computations can be used to optimize straight line DSP circuits [3]. Verma et al. developed transformation techniques around the application's DFG to make best use of CS arithmetic prior the particular datapath

synthesis. These CS optimization approaches target inflexible datapath, i.e., ASIC, implementations. Lately, Xydis et al. suggested an adaptable architecture mixing the ILP and pipelining techniques using the CS-aware operation chaining. However, all of the aforementioned solutions feature a natural limitation, i.e., CS optimization is bounded to merging only additions/subtractions. A CS to binary conversion is placed before each operation that is different from addition/subtraction, e.g., multiplication, thus, allocating multiple CS to binary conversions that heavily degrades performance because of time-consuming carry propagations[3]. Within this brief, we advise a higher-performance architectural plan for that synthesis of flexible hardware DSP accelerators by mixing optimization techniques from both architecture and arithmetic amounts of abstraction. We introduce an adaptable datapath architecture that exploits CS enhanced templates of chained procedures. The suggested architecture comprises flexible computational models (FCUs), which let the execution of a big group of operation templates present in DSP popcorn kernels. The suggested accelerator architecture provides average gains as high as 61.91% in area-delay product and 54.43% in energy consumption in comparison to condition-of-art flexible datapaths, sustaining efficiency toward scaly technologies.

.

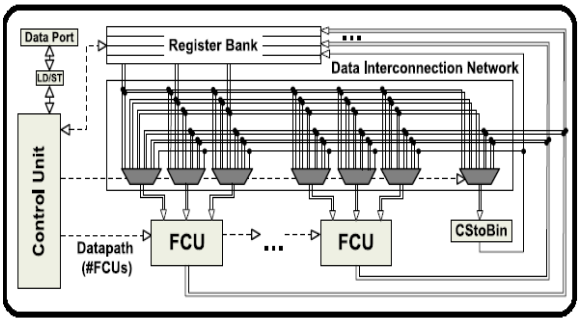

Fig.1. Framework of the flexible datapath

II. PROPOSED SYSTEM

The goal is to maximize the range that a CS computation is performed within the DFG. However, whenever a multiplication node is interleaved in the DFG, either a CS to binary conversion is invoked or the DFG is transformed using the distributive property CS representation has been widely used to design fast arithmetic circuits due to its inherent advantage of eliminating the large carry-propagation chains. CS arithmetic optimizations, rearrange the application's DFG and reveal multiple input additive operations, which can be mapped onto CS compressors.. Thus, the aforementioned CS optimization approaches have limited impact

on DFGs dominated by multiplications, e.g., filtering DSP applications. In this brief, we tackle the aforementioned limitation by exploiting the CS to modified Booth (MB) recoding each time a multiplication needs to be performed within a CS-optimized datapath. The proposed flexible accelerator architecture is presented. Each FCU operates directly on CS operands and produces data in the same form1 for direct reuse of intermediate results. Each FCU operates on 16-bit operands [4]. Such a bit-length is adequate for the most DSP datapaths, but the architectural concept of the FCU can be straightforwardly adapted for smaller or larger bit-lengths. The number of FCUs is determined at design time based on the ILP and area constraints imposed by the designer. The CStoBin module is a ripple-carry adder and converts the CS form to the two's complement one. The register bank consists of scratch registers and is used for storing intermediate results and sharing operands among the FCUs. Different DSP kernels (i.e., different register allocation and data communication patterns per kernel) can be mapped onto the proposed architecture using post-RTL datapath interconnection sharing techniques. The control unit drives the overall architecture (i.e., communication between the data port and the register bank, configuration words of the FCUs and selection signals for the multiplexers) in each clock cycle. The structure of the FCU has been designed to enable high-performance flexible operation chaining based on a library of operation templates The proposed FCU enables intratemplate operation chaining by fusing the additions performed before/after the multiplication and performs any partial operation template of the complex operations: The multiplier's product consists of 17 bits. The multiplier includes a compensation method for reducing the error imposed at the product's accuracy by the truncation technique [5]. However, since all the FCU inputs consist of 16 bits and provided that there are no overflows, In order to efficiently map DSP kernels onto the proposed FCU-based accelerator, the semiautomatic synthesis methodology presented, has been adapted. At first, a CS-aware transformation is performed onto the original DFG, merging nodes of multiple chained additions/subtractions to 4:2 compressors. A pattern generation on the transformed DFG clusters the CS nodes with the multiplication operations to form FCU template operations. The designer selects the FCU operations

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

covering the DFG for minimized latency. Given that the number of FCUs is fixed, a resource-constrained scheduling is considered with the available FCUs and CStoBin modules determining the resource constraint set. The clustered DFG is scheduled, so that each FCU operation is assigned to a specific control step. A list-based scheduler has been adopted considering the mobility2 of FCU operations. The FCU operations are scheduled according to descending mobility. The scheduled FCU operations are bound onto FCU instances and proper configuration bits are generated. After completing register allocation, a FSM is generated in order to implement the control unit of the overall architecture.

RESULTS:

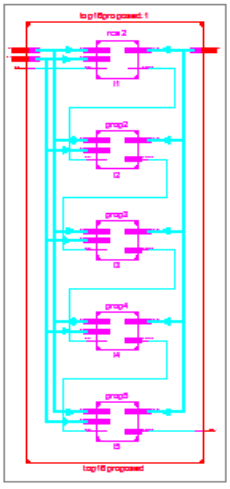

Fig: RTL SCHEMATIC

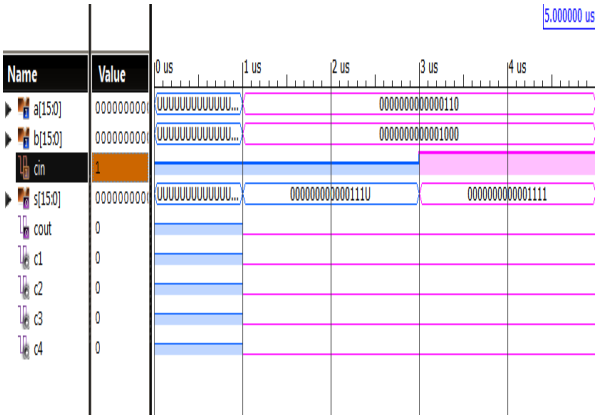

Fig: Simulation Forms

III. CONCLUSION

Within this brief, we introduced an adaptable accelerator architecture that exploits the incorporation of CS arithmetic optimizations to allow fast chaining of additive and multiplicative procedures. The suggested flexible accelerator architecture has the capacity to work on both conventional two's complement and CS-formatted data operands, thus enabling high levels of computational density to become accomplished. Theoretical and experimental analyses have proven the suggested solution forms a competent design compromise point delivering enhanced latency/area and implementations. Case study is dependent on the system gate model. Regarding both execution latency and also the area complexity and thinking about all of the DSP popcorn kernels, the suggested FCU-based architecture outperforms those built around the FCC and also the RAU. Not

surprisingly, the timing constraints and also the results of cell sizing implied through the Design Compiler synthesis tool, in some instances lead to incongruences between your experimental and also the theoretical studies

REFERENCES

[1] N. Moreano, E. Borin, C. de Souza, and G. Araujo, "Efficient datapath merging for partially reconfigurable architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 7, pp. 969–980,Jul. 2005.

[2] M. D. Galanis, G. Theodoridis, S. Tragoudas, and C. E. Goutis, "A high-performance data path for synthesizing DSP kernels," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1154–1162, Jun. 2006.

[3] A. K. Verma, P. Brisk, and P. Ienne, "Data-flow transformations to maximize the use of carry-save representation in arithmetic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1761–1774, Oct. 2008.

[4] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Reading, MA, USA: Addison-Wesley, 2010.

[5] M. Stojilovic, D. Novo, L. Saranovac, P. Brisk, and P. Ienne, "Selective flexibility: Creating domain-specific reconfigurable arrays," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 5, pp. 681–694, May 2013.