IMPLEMENTING A FLEXIBLE COMPUTATIONAL UNIT TO PERFORM FDPA OPERATIONS TO LIMITING THE CARRY IN OPTIMIZED DATA PATH

Mr. SK.MASTAN[1], Miss.B.MAMATHA[2]
1. Associate Professor, Department of ECE ,Jayamukhi Institute of Technogical Sciences , Warangal, India
2. Department of ECE, Jayamukhi Institute of Technological Sciences, Warangal, India.

Abstract: This paper presents a novelapproach to design Flexible computational unit(FCU) exploiting carry save arithmetic anddadda multiplier. Earlier projects emphasized ondesigning systems with low power and fastoperations resulting in increased area. The goalof this project is to design DSP system to makethe design more efficient by proposing Daddatree multiplier. Utilizing Dadda tree multiplierreduces the area required for the advisement. Asin these modern times, most of the electronicsystems are adapted to perform digital signalprocessing; the DSP integrated systems havewide range of applications. Kernel which is thecore of the DSP system consists of data flowgraph that features the corresponding arithmeticoperation performed by the system. These DFGsbasically consist of arithmetic units such asadders, multipliers etc that are mapped onto theproposed flexible control units which are carrysave formatted data. As we know that the basicintension of designing integrated circuits is tominimize the dimensions of a given circuit. Thisobjective is achieved by incorporating dada multiplier as multiplier unit of the DSP dataflow graph template. This implementationshows considerable difference in terms of areawhich further leads to other advantages such asto fabricate more number of components ontothe resulted area.

Index Terms: Digital signal processing (DSP),Data flow graph (DFG), Register Transfer logic(RTL), Template(T), Spurious powersuppression technique (SPST), Carry save arithmetic (CSA).

I. Introduction

Digital signal processing (DSP) play a very vital role in our daily life.Digital signal processing techniques are increasingly replacing conventional analog signal processing methods in many fields. DSP systems have wide range of applications such as Radar, Multi-rate processing and non-linear DSP, Analogue emulation systems, Networking, Audio/video/multimedia, Noise cancellation systems, Biomedical, Non-destructive testing, Pattern recognition and matching, Control systems engineering, Digital waveform synthesis, remote sensing, Earth-based telecommunications, Robotics, Image processing, Satellite telemetry, Image processing in all its representations, Seismology, Industrial signal conditioning, Speech recognition/ synthesis, Mechatronics, Scientific instrumentation, Signal analysis, Military/surveillance, Multiplexing etc. DSP system is the use of digital processing, such as by computers, to perform a wide variety of signal processing operations. The signals processed in this manner are a sequence of numbers that represent samples of a continuous variable in a domain such as time, space, or frequency. A Digital signal processing (DSP) is a specialized microprocessor (or a SIP block), with its architecture optimized for the operational needs of Digital signal processing (DSP). This project projects the

evolution of current project from preceding projects that are existing and proposed project. The Existing project mainly dealt with implementation of SPST adders in adder unit of the DFG of the given DSP data flow graph resulting in speed of the system. The proposed system incorporated carry save adder in the data flow graph instead of the SPST adder resulting in further increase in the overall speed of the system but compromising for the area i.e. increase in the dimensions of the system. Our project which is its extension incorporates dadda multiplier reduction in area which is remarkably advantageous especially for integrated circuits.

Modern embedded systems target high-end application domainsrequiring efficient implementations of computationally intensivedigital signal processing (DSP) functions. The incorporation ofheterogeneity through specialized hardware accelerators improvesperformance and reduces energy consumption. Althoughapplication-specific integrated circuits (ASICs) form the ideal accelerationsolution in terms of performance and power, their inflexibilityleads to increased silicon complexity, as multiple instantiated ASICsare needed to accelerate various kernels. Many researchers haveproposed the use of domain-specific coarse-grained reconfigurableaccelerators in order to increase ASICs' flexibility withoutsignificantly compromising their performance.

II. Proposed flexible accelerator

Flexible computational unit (FCU) exploiting carry save arithmetic and dadda multiplier. Earlier projects emphasized on designing systems with low power and fast operations resulting in increased area. The goal of this project is to design DSP system to make the design more efficient by proposing Dadda tree multiplier. Utilizing Dadda tree multiplier reduces the area required for the advisement. As in these modern times, most of the electronic systems are adapted to perform digital signal processing; the DSP integrated systems have wide range of applications. Kernel which is the core of the DSP system consists of data flow graph that features the corresponding arithmetic operation performed by the system. These DFGs basically consist of arithmetic units such as adders, multipliers etc that are mapped onto the proposed flexible control units which are carry save formatted data.

The proposed flexible accelerator architecture is shown in Fig. 1.Each FCU operates directly on CS operands and produces data inthe same form1 for direct reuse of intermediate results. Each FCUoperates on 16-bit operands. Such a bit-length is adequate for themost DSP datapaths but the architectural concept of the FCUcan 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 CStoBinmodule is a ripple-carry adder and converts the CS form to the two'scomplement one. The register bank consists of scratch registers andis used for storing intermediate results and sharing operands amongthe FCUs. Different DSP kernels (i.e., different register allocationand data communication patterns per kernel) can be mapped ontothe proposed architecture using post-RTL datapath interconnection
Sharing techniques. The control unit drives the overallarchitecture (i.e., communication between the data port and theregister bank, configuration words of the FCUs and selection signalsfor the multiplexers) in each clock cycle.

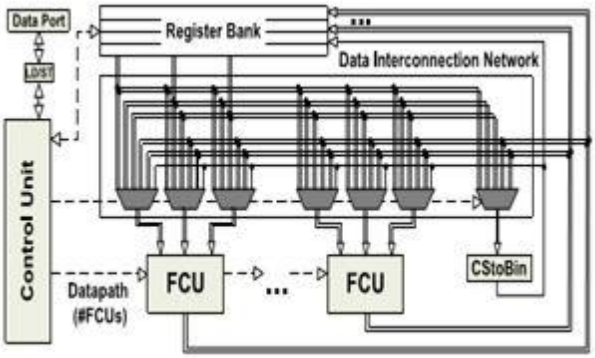

Fig.1 : Flexible accelerator architecture.

Data Path

The structure of the FCU (Fig. 2) hasbeen designed to enable high-performanceflexible operation chaining based on a library ofoperation templates. Each FCU can beconfigured to any of the T1–T5 operationtemplates in template library. The proposedFCU enables intra-template operation chainingby fusing the additions performed before afterthe multiplication &performs any partialoperation template of the following complexoperations:

$W = A \times (X + Y) + K$ (1)

$W = A \times K + (X + Y)$ (2)

The above specified equations can beimplemented simultaneously in a operationaltemplate found in the FCU as shown in figure 2.

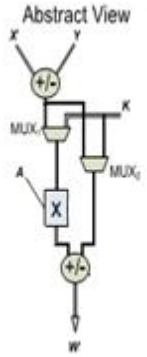

Fig.2: shows a segment of the internal structureof the FCU i.e. operational template of the datapath specified in equation (1) and (2)

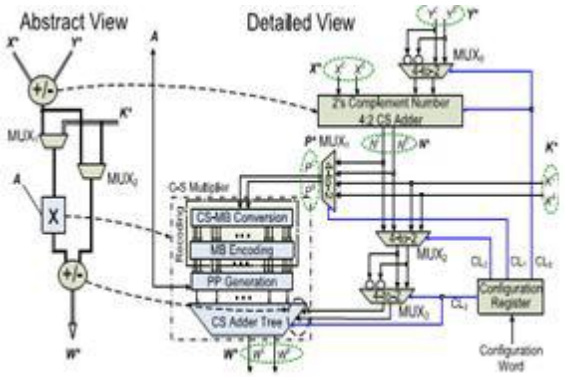

Fig.3. Implementation of carry-save technique for adders and modified booth technique for multiplier in FCU.

The aforementioned reconfigurable architectures excludearithmetic optimizations during the architectural synthesis andconsider them only at the internal circuit structure of primitivecomponents, e.g., adders, during the logic synthesis. However, research activities have shown that the arithmeticoptimizations at higher abstraction levels than the structuralcircuit one significantly impact on the datapath performance. The timing-driven optimizations based on carry-save (CS)arithmetic were performed at the post-Register Transfer Level (RTL) design stage. The common subexpression eliminationin CS computations is used to optimize linear DSP circuits.The developed transformation techniques on theapplication's DFG to maximize the use of CS arithmetic prior theactual datapath synthesis. The aforementioned CS optimizationapproaches target inflexible datapath, i.e., ASIC, implementations.Recentlyproposed a flexible architecturecombining the ILP and pipelining techniques with the CS-awareoperation chaining. However, all the aforementioned solutions featurean inherent limitation, i.e., CS optimization is bounded to mergingonly additions/subtractions. A CS to binary conversion is insertedbefore each operation that differs from addition/subtraction, e.g.,multiplication, thus, allocating multiple CS to binary conversionsthat heavily degrades

performance due to time-consuming carrypropagations.

Structure of the Proposed Flexible Computational Unit:

The structure of the FCU (Fig. 2) has been designed to enable high-performance flexible operation chaining based on a library of operationtemplates. Each FCU can be configured to any of the T1–T5 operation templates shown in Fig. 4. The propose FCU enables intra template operation chaining by fusing the additions.

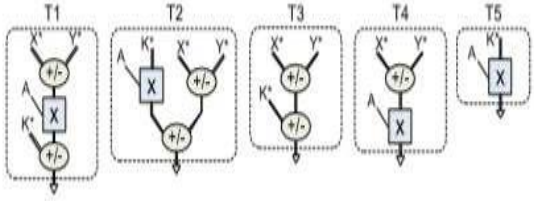

Fig. 4. FCU template library

DFG Mapping Onto the Proposed FCU-Based Architecture:

In order to efficiently map DSP kernels onto the proposedFCU-based accelerator, the semiautomatic synthesis methodology has been adapted. At first, a CS-aware transformationis performed onto the original DFG, merging nodes ofmultiple chained additions/subtractions to 4:2 compressors. A patterngeneration on the transformed DFG clusters the CS nodes with themultiplication operations to form FCU template operations (Fig. 4).The designer selects the FCU operations covering the DFG forminimized latency.

Given that the number of FCUs is fixed, a resource-constrainedscheduling is considered with the available FCUs and CStoBinmodules determining the resource constraint set. The clustered DFGis scheduled, so that each FCU operation is assigned to a specificcontrol step. A list-based scheduler has been adopted consideringthe mobility2 of FCU operations. The FCU operations are scheduledaccording to descending mobility. The scheduled FCU operationsare bound onto FCU instances and proper configuration bits aregenerated. After completing register allocation, a FSM is generatedin order to implement the control unit of the overall architecture.

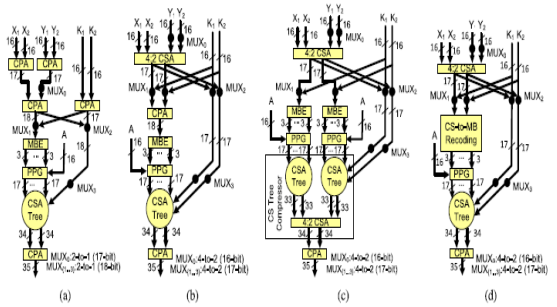

Fig. 5. Typical chaining of addition–multiplication–addition operations reflecting T1 template of Fig. 4.

III. Simulation results

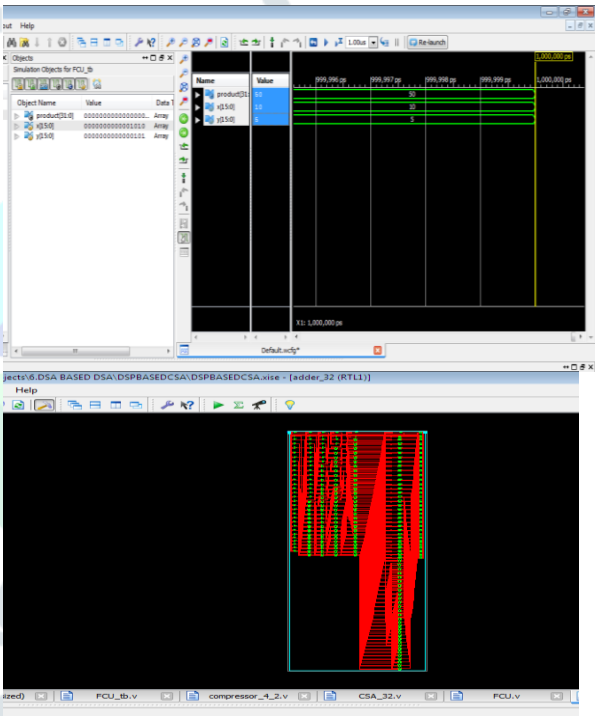

Fig.6.DSP based CSA multiplier

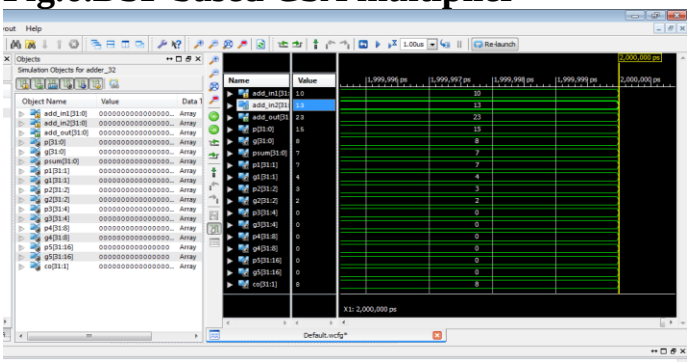

Fig.7. DSP based CSA adder

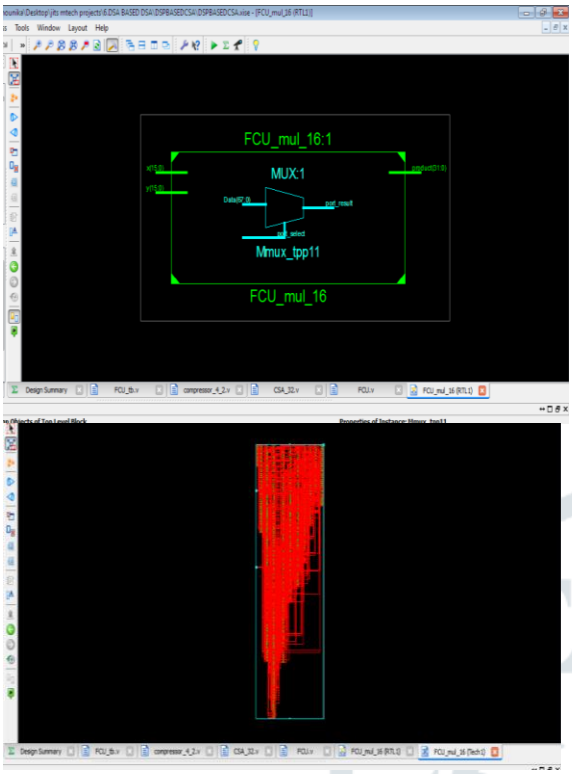

Fig 8. Schematics of multiplier unit

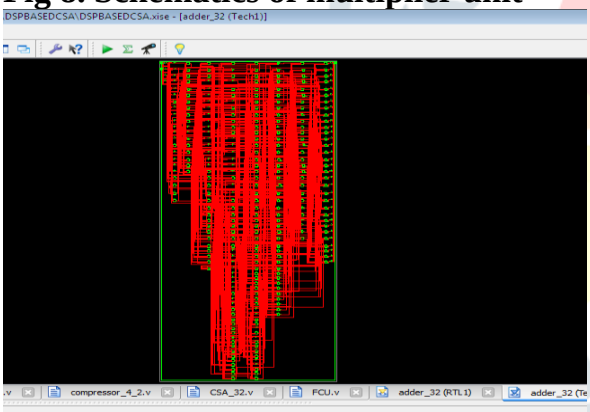

Fig.9. Schematics of adder unit

IV. CONCLUSION

Since area parameter is a major concern in the integrated circuits, so our extended project exploits dadda multiplier for multiplier in FCU. The internal operation in the dada multiplier includes adders exploiting carry save arithmetic. In our project as stated earlier, the dadda multiplier is implemented in the multiplier unit of the operational template found

Concisely, we introduced an FCU architecture that exploits the incorporation of CS arithmetic and dadda algorithmic optimizations to enable fast chaining of additive and multiplicative operations. This flexibleaccelerator architecture allow to operate on both conventional two's complement and CSformatted data operands, thus enabling high degrees of computational density to be achieved.Theoretical and experimental analyses have shown that the extended solution forms an efficient design delivering considerable amountin terms of area.

REFERENCES:

[1] P. Ienne and R. Leupers, *Customizable Embedded Processors:Design Technologies and Applications*. San Francisco, CA, USA:Morgan Kaufmann, 2007.

[2] P. M. Heysters, G. J. M. Smit, and E. Molenkamp, "A flexible andenergy-efficient coarse-grained reconfigurable architecture for mobilesystems," *J. Supercomput.*, vol. 26, no. 3, pp. 283–308, 2003.

[3] B. Mei, S. Vernalde, D. Verkest, H. D. Man, and R. Lauwereins,"ADRES: An architecture with tightly coupled VLIW processor andcoarse-grained reconfigurable matrix," in *Proc. 13th Int. Conf. FieldProgram. Logic Appl.*, vol. 2778. 2003, pp. 61–70.

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

[5] K. Compton and S. Hauck, "Automatic design of reconfigurable domainspecificflexible cores," *IEEE Trans. Very Large Scale Integr. (VLSI)Syst.*, vol. 16, no. 5, pp. 493–503, May 2008.

[6] S. Xydis, G. Economakos, and K. Pekmestzi, "Designing coarse-grainreconfigurable architectures by inlining flexibility into custom arithmeticdata-paths," *Integr., VLSI J.*, vol. 42, no. 4, pp. 486–503, Sep. 2009.

[7] S. Xydis, G. Economakos, D. Soudris, and K. Pekmestzi, "High performanceand area efficient flexible DSP datapath

synthesis," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 3, pp. 429–442,Mar. 2011

[8] G. Ansaloni, P. Bonzini, and L. Pozzi, "EGRA: A coarse grainedreconfigurable architectural template," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 19, no. 6, pp. 1062–1074, Jun. 2011.

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

[10] R. Kastner, A. Kaplan, S. O. Memik, and E. Bozorgzadeh, "Instructiongeneration for hybrid reconfigurable systems," *ACM Trans. DesignAutom. Electron. Syst.*, vol. 7, no. 4, pp. 605–627, Oct. 2002.