SEYBOLD Report
ISSN: 1533 - 9211
DOI: 10.5281/zenodo.6877538

RECONFIGURABLE CONSTANT MULTIPLICATION FOR FPGA'S

Kashapaga Soni
Assistant Professor, Department of electronics and Communication Engineering
Sridevi Women's Engineering College, Hyderabad, India, sonikashapaga@gmail.com

Murala Roshitha
U.G Student, Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Hyderabad, India

Modukuri Ramya
U.G Student, Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Hyderabad, India

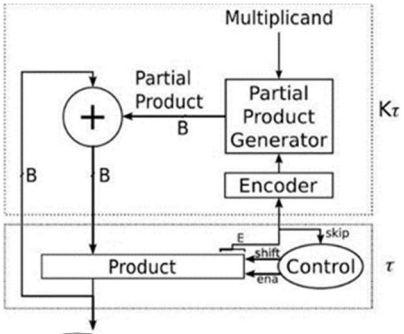
Errola Nirosha
U.G Student, Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Hyderabad, India

Abstract: The article provides a new proposed system for reconfigurable constant multiplication for fpga's as it improves the 70% accuracy compared to that of existing system. For every constant multiplication here, we are using the control units to store the shift registers to avoid the conflict for every constant multiplication of (zeros). This is mostly used in filtering techniques and digital signal processing (DSP). Further included VDHL code generation based on Xilinx software.

Keywords: Fpga's, VHDL, shift registers, control units

1. Introduction

As the multiplication operation in the system will be done by using the binary digits the product of the multiplication is the partial product(PP).When we take only the ones for multiplication we get the same partial product but when we take the ones and zeros for multiplying the binary digit we get the zeros partial product for zeros so here we have to use the shift register to shift the digit to right to avoid the zeros in partial product, but all the time i.e., for n zeros we cannot use the shift registers n times in that case we use the control unit to store the shift register to shift the digit to right.



SEYBOLD Report
ISSN: 1533 - 9211
DOI: 10.5281/zenodo.6877538

Fig: Reconfigurable constant multiplier for FPGA's

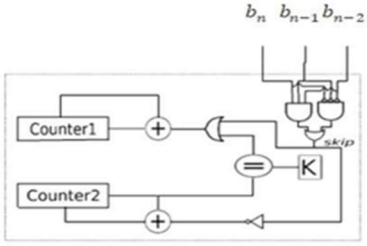


Fig: Control type shift register

So, for constant multiplication of n field gate arrays we can use the control unit to get optimum result. As Reconfigurable constant multiplication for FPGAs uses the control unit shift register as shown in the figure, so using control unit that it can perform the multiplication for n field gate arrays which gives the optimum results.

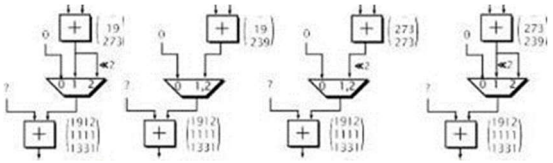
2. Method

The input to the algorithm are pipelined adder graphs (PAGs) generated with the reduced pipelined adder graph (RPAG) heuristic [1]. In general, the presented fusion is not limited to RPAG generated circuits as pipelined MCM input. However, RPAG was chosen as it proved to outperform state-of-the-art MCM methods like Hcub.when these are optimally pipelined. The results of RPAG are adder graphs representing multiplier-less pipelined constant multipliers using additions, subtractions and bit-shifts only. The main idea of multiplier-less multiplication as applied in RPAG is to compose a constant multiplication of an addition of shifted inputs. This is beneficial because a constant shift is only a wire in hardware. All constants can be formally represented as A-operation , which is defined as

Aq(u, v) = |2l1 u + (−1)sg2l2 v|2−r (1) with q

= (l1, l2, r, sg), where u and v are the input constants, l1, l2 and r are shift factorsand the sign bit sg 0, 1 denotes whether an addition or subtraction is performed. A multiplication by 17 could for example be realized as an addition of the input with theinput left-shifted by 4

a) Possible mappings for first preceding adder.



b) Possible mappings for second preceding adder.

A. Reconfigurable Multiple Constant Multiplication

When reconfigurable multiple constant multiplication is considered, it can again be compared to the CoreGen soft- core multiplier with RAM for the coefficients. To have more than one output, multiple CoreGen multipliers and coefficient RAMs are used. A benchmark for 5 different MCM cases (2, 4, 6, 8 and 10 outputs), each with 2, 4, 6, 8 and 10 configurations, consisting of 50 constant sets per case using randomly generated constants uniformly distributed between 1 and 216 1 was created. The search width was again set to 64. The results of the 2-input and 3-input adder implementations compared to CoreGen multipliers can be

SEYBOLD Report
ISSN: 1533 - 9211
DOI: 10.5281/zenodo.6877538

found in Figure.It can be seen that the CoreGen soft-core multiplier implementation is better for 6 or more configurations in the 2-input (multiplication by 16). This can be seen in the leftmost example in Figure. In the following subtractor, 17 times the input is subtracted from 256 times the input, which corresponds to a constant multiplication by 239. Finally, this intermediate result is left- shifted by three to get the final result of 1912 times the input. If the constant to multiply with is known in advance, this kind of realization is much cheaper in termsof resources than implementing a generic multiplier. In order to automatically generate such constant multipliers, RPAG is backward-exploring reachable intermediate constants, called predecessors by evaluating the operation. This leads to a step-wise constant composition, starting with the required output constants. The goal of the heuristic is to select predecessors which result in the lowest number of intermediate constants in the preceding stage and which reduce the adder depth. Two more examples for such a circuit of a pipelined SCM realization can be found in Figure, which are used as running example. The stage s denotes the pipeline depth of each realized constant.

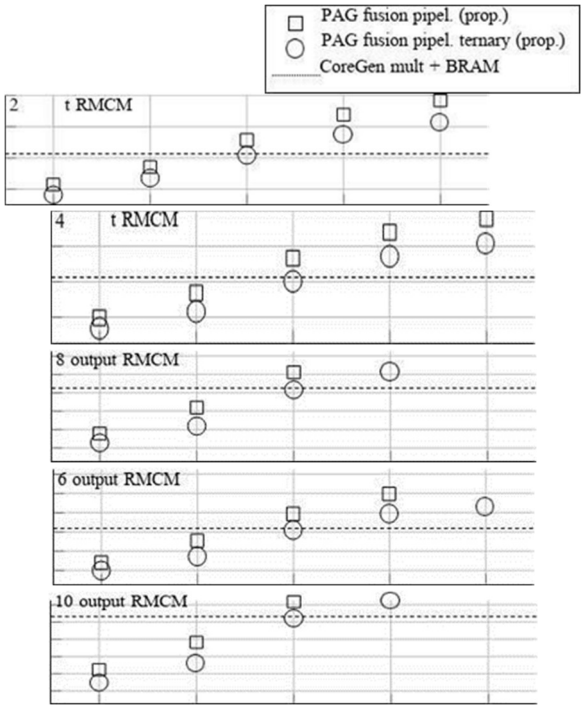


Fig: Comparison of RMCM and tRMCM to a CoreGen soft-core multiplier + RAM. adder case and 8 or more configurations in the ternary adder case. Below these numbers of configurations, up to 75% of the resources can be saved, when the proposed reconfigurable shift and add based implementation is preferred, which is up to 750 slices in the 10 output RMCM case. Note that without the heuristic only the results for 2 outputs and 2 configurations could have been generated optimally within a run-time limit of 3 hours. MCM solutions normally have more adders in each stage, which leads to a much larger search space and thus a much larger run-time. Using the heuristic with its controllable

SEYBOLD Report
ISSN: 1533 - 9211
DOI: 10.5281/zenodo.6877538

B.DSP Block Usage Conisderation

On modern FPGAs, DSP blocks in combination with RAM for the coefficients can be used instead of the proposed run- time reconfigurable constant multiplication. If limited quantity of DSP blocks is not a problem, each of the 16 x16 bit multipliers of reported cases could be replaced by one DSP block and two slices for the coefficient RAM. For multiplication word sizes larger than 18 Bit, more than one DSP block would be required for Xilinx FPGAs. A comparison of the usage of DSP blocks to the proposed slice-based method can be done by relating the two types of resources (DSP blocks and slices) according to their relative availability, referencing their utilization ratio [15]. Alternatively, the chip area consumed by the resources can be related [36]. However, neither of the two methods addresses the frequent requirement to select the smallest, hence cheapest, possible FPGA the design fits into. Usually, in a complete design other parts are competing for DSP resources in digital signal processing applications [37], [38]. For such cases a trade-off must be available. This is provided by the proposed slice- based run-time reconfigurable constant multiplier implementation.

3. Research Results

There are two broad groups of the key resources needed for this project.

- Need for hardware
- Needed applications

On hardware component, you must provide a device with a minimum machine setup Pentium III, 1 GB of RAM, and 20 GB of hard disk, where Xilinx ISE 10.1i program will easily be run. This method can be used for the creation, execution.

- Translate Temporary Development (NGDBuild)
- Transforms all input design netlists to a single merged format, defining the rationale and limitations.
- Towards Routing (MAP) Towards

search width, raises the solvable problem size and and thereby enables the application domain of RMCM for the proposed fusion algorithm. For the application domains given in the introduction [1], [2], [3], [4], [5], 2 to 6 MCM configurations are common configurations restoration, and incorporation of Verilog outlines on FPGA chips.

ISE: Environmental integrated applications

- atmosphere for developing and checking the configuration of FPGA or CPLD computerized systems.
- Integrated selection of devices through an Interface
- A clever hybrid engine (XST: Xilinx Synthesis Technology)

XST embraces different dialects

> THE MANAGE

- Against VHDL

SEYBOLD Report
ISSN: 1533 - 9211
DOI: 10.5281/zenodo.6877538

- XST build a net reworking of demands
- Supports all the means appropriate to complete the plan:Translating, directing, putting and understanding.
- It was a bit stream.

In that case, the usage of Verilog to build a testseat may be conceived to ensure that the contours of documents on the host PC are useful for characterizing jolts, interfacing and comparing them with the client. A Verilog display is turned into "doors and wires" that are mapped in a programmable logic gadget, such as a CPLD or FPGA; so it is the actual machinery that is being built, rather than "executed" the verilog code as on a certain sortof chip.

Delivery

- Description (XST)
- Produce an HDL definition netlist file

- Maps the part logic of the unit.
- Takes logical elements to the CLBs and IOB (FPGA components) and group them into a netlist.
- Towards Location And Path (PAR)

- Position and bind cells with FPGA.
- Towards the processing of bit streams
- Phase of XILINX Architecture

4. Discussion and Results

By using the Proposed system of Reconfigurable constant multiplication for FPGAs i. e., using of control unit for shifting the registers can reducing the cost of the project (hardware kit) and time efficiency and improves the fusion process for the field gate arrays as we are using the and operation here, we can efficiently maintain the accurate partial product for the constant multiplication of fieldgate arrays.

5. Conclusions

This work presented a new heuristic to generate pipelined run-time reconfigurable constant multipliers based on an optimal algorithm. The heuristic was motivated by a complexity consideration of the search space. With this heuristic, problems with a larger size become solvable. An extensive benchmark evaluation showed superiority over previous work, as we could show a 926% slice reduction on average. Additional extensions to the algorithm were presented which further reduce the slice consumption of the resulting solutions.These were the support of ternary adders and optimized multiplexer and switchable adder/subtractor mapping. Finally, it could be shown by RMCM and FIR filter experiments that the heuristic is raising the

DOI: 10.5281/zenodo.6877538

solvable problem size and the application domain of the proposed fusion method. Compared to other reconfiguration approaches our method provides the fastest reconfiguration time with a low resource consumption for a limited number of configurations. The source code of the proposed method is available online as open source within the PAG Suite project to increase reproducibility and provide it for future research.

References:

[1] M. Kumm, P. Zipf, M. Faust, and C.H. Chang, "Pipelined Adder Graph in Circuits and Systems ISCAS, IEEE International Symposium on, May 2012, pp. 49–52.

[2] M. Kumm, "Multiple Constant Multiplication Optimizations for Field Programmable Gate Arrays," Ph.D. thesis, University of Kassel, Springer, 2016.

[3] K. Mo¨ller, M. Kumm, B. Barschtipan, and P. Zipf, "Dynamically Reconfigurable Constant Multiplication on FPGAs," in WorkshopMethodenund Beschreibungssprachen zur Modellierung und Verification von Schaltungen und Systemen (MBMV), 2014, pp. 159–169.

[4] K. Mo¨ller, M. Kumm, M. Kleinlein, and P. Zipf, "Pipelined Reconfigurable Multiplication with Constants on FPGAs," in Field Programmable Logic and Applications (FPL), 2014 24th International Conference on, 2014, pp. 1–6.

[5] M. Kumm, K. Mo¨ller, and P. Zipf, "Dynamically Reconfigurable FIR Filter Architectures with Fast Reconfiguration," Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 8th International Workshop on, pp. 1–8, 2013.

[6] K. Chapman, "Constant Coefficient Multipliers for the XC4000E," Xilinx Application Note, 1996.

[7] M. Kumm, P. Zipf, M. Faust, and C.H. Chang, "Pipelined Adder Graph Optimization for High-Speed Multiple Constant Multiplication," in IEEE Int. Symposium on Circuits and system.