Digital Signal Processing Accelerator for RISC-V

Calicchia, L.; Ciotoli, V.; Cardarilli, G. C.; Di Nunzio, L. ; Fazzolari, R.; Nannarelli, Alberto

Published in:
Proceedings of 26th IEEE International Conference on Electronics Circuits and Systems

Link to article, DOI:
10.1109/ICECS46596.2019.8964670

Publication date:
2019

Document Version
Peer reviewed version

Digital Signal Processing Accelerator for RISC-V

L. Calicchia, V. Ciotoli, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, A. Nannarelli,[(1)] *and M. Re*

Department of Electronics, University of Rome Tor Vergata, Rome, Italy
[(1)]DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark

***Abstract*—In this work, we present a configurable accelerator for the RISC-V processor oriented to digital signal processing applications for energy efficient Internet-of-Things devices. The supported operations in the accelerator are addition, multiplication, and linear combination. The accelerator can support different applications: mono-dimensional and bi-dimensional filtering and pattern matching. The results show that the configurable accelerator offers better performance and lower energy consumption when compared to the software execution of the same application on the RISC-V.**

***Index Terms*—DSP, accelerator, energy efficiency, RISC-V.**

I. Introduction

Modern SoCs may contain several hardware accelerators to implement computationally intensive tasks that are unfeasible in time and energy by using traditional CPUs. Analysis of die photos from different consumer and professional electronics companies shows that more than half of the die area is used by blocks other than the CPUs [1].

Traditionally, accelerators have been introduced to lower the execution time for particularly computationally intensive tasks. More recently, the extensive use of accelerators has been motivated by the need to lower the energy consumption. In fact, the accelerator, even if characterized by high power consumption peaks, guarantees a task execution in a very short time, lowering the energy per task [2].

In this work, we propose to use an accelerator in a RISC-V ecosystem to accelerate Digital Signal Processing (DSP) applications in a System-on-Chip (SoC) intended for Internet-of-Things. The target is to off-load the RISC-V core(s) from heavy computations, sometimes required in DSP, to optimize the timing and to reduce the energy consumption of the SoC. The accelerator is configurable in terms of bit-width (dynamic range), type of operations (multiplication or addition), and sequence of operations.

II. Related Work

In general, the use of accelerators is very important in Intelligent Computation applications and the RISC-V is emerging as the CPU of choice for those systems. In [3] the authors presented LACore, a novel, programmable accelerator architecture for general-purpose linear algebra applications suitable for the RISC-V new generation of processor cores. In [4], the authors integrate Nvidia Deep Learning Accelerator (NVDLA) into a RISC-V SoC. NVDLA, is an open-source deep neural network (DNN) accelerator which has received a lot of attention by the research community since its introduction by Nvidia.

Moreover, different companies and research groups are developing accelerators and CAD Tools. Open Celerity [5] is an accelerator-centric SoC which uses a tiered accelerator fabric to improve energy efficiency in high-performance embedded systems. Celerity currently holds the world record for RISC-V performance. Bluespec [6] sells a complete product suite for integrating, verifying, and debugging embedded systems accelerators in RISC-V ecosystem to increase software speed-power ratios 10 to 100 times (BlueAccel tool chain).

III. Accelerator Architecture

The accelerator is intended for DSP or image processing applications such as filtering in the time domain (mono-dimensional convolution) and image filtering in the spatial domain (bi-dimensional convolution). A local control unit is used to configure the accelerator, based on an instruction received from the RISC-V core.

To save power, the dynamic range of the accelerator's datapath can be reduced depending on the requirements of the application. The dynamic range is adjusted by disabling the logic gates in the least-significant bits of the processing elements. This method is referred as Degrading Precision Arithmetic method I, or DPA-I, in [7].

By DPA-I, the overall switching activity of the datapath is reduced by either clock-gating paths starting from a register or by forcing zeros, or ones, in combinational logic. The results in [7] show that for a FIR filter the power savings are about 30% if the application can tolerate an error $\epsilon = 2^{-\frac{n}{2}}$ for a n-bit (fractional) dynamic range.

We design for the accelerator a processing element (PE) performing addition and multiplication depending on two configuration bits OP (Fig. 1). The PE is a fixed-point multiply-and-add configured by OP as follows:

$00 \rightarrow Z = X \times Y + 0 = X \times Y$ (multiplication)
$10 \rightarrow Z = 1 \times Y + X = X + Y$ (addition)
$11 \rightarrow Z = -1 \times Y + X = X - Y$ (subtraction)

By adding a third input and changing the muxes' set-up, the PE can easily perform fused multiply-and-add : $Z = X \times Y + W$.

The architecture of the reconfigurable accelerator is shown in Fig. 2. We designed our accelerator to have a maximum dynamic range of 32 bits, which is compatible with most DSP applications, 64 processing elements PEs, (either multipliers or adders), and a tree of adders to add the output of the 64 PEs. The inputs of the PEs are connected to two 16-bit shift-registers (Reg. A and B in Fig. 2) and their output to the tree of adders.

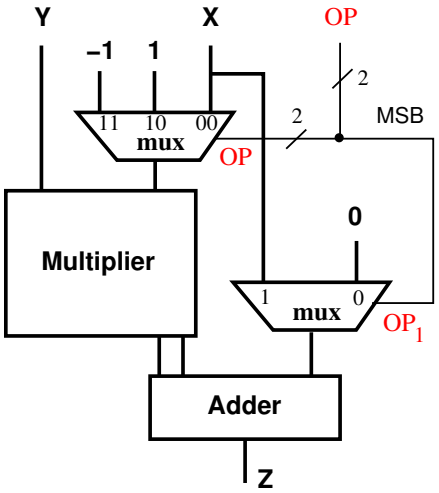

Fig. 1. Reconfigurable basic cell (PE).

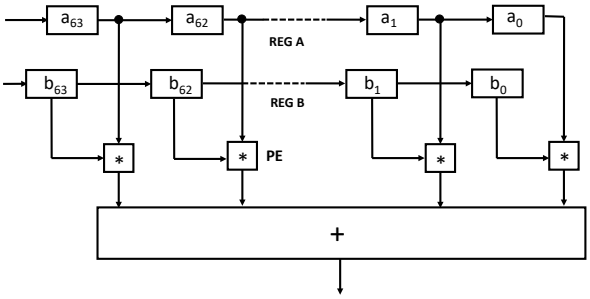

Fig. 2. Architecture of configurable processor.

To reduce the dynamic range by DPA-I, we force zeros in the LSBs of the shift-registers A and B.

The instruction sent by the RISC-V core makes the accelerator controller to execute the following tasks:

- To set the dynamic range by forcing zeros in the LSBs of the registers.
- To set the operations to be performed in the PEs. This is done by setting OP in all the PEs (Fig. 1).
- To enable/disable loading the shift-registers.
- The pointer to the memory location where the stream of data are loaded and stored, and the number of data to process.

IV. RISC-V Acceleration Environment for DSP

A. RISC-V Design Flow

For the RISC-V SoC we opted for the Berkeley's RISC-V Rocket Chip open-source release [8]. The Rocket Chip development tools generate a RISC-V multi-core system, plus interfaces, floating-point units, and so on. The chip configuration is specified in the "Chisel" language, and the tools are able to generate synthesizable Verilog code and test-benches. The design flow used to develop the SoC with the accelerator is summarized in Fig. 3. The flow is based on Synopsys' simulation (VCS) and synthesis (Design Compiler, or DC) tools.

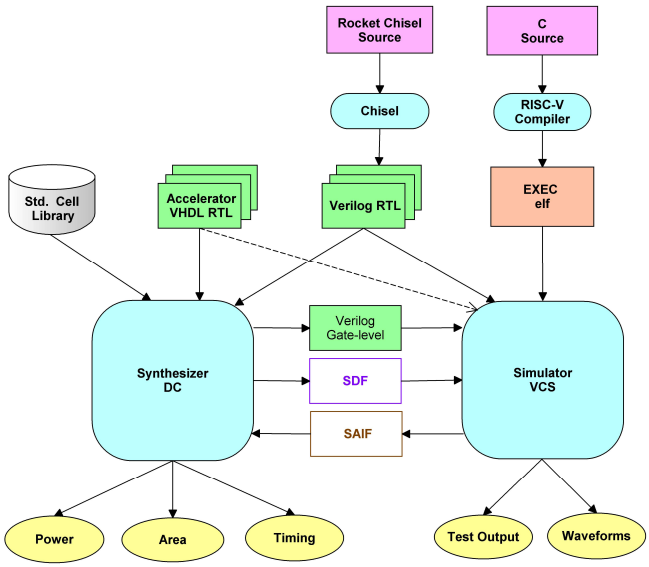

Fig. 3. Design flow for RISC-V plus accelerator.

In addition to the Verilog generator, the Rocket's suite of tools provides also a RISC-V gcc compiler generating executables in "elf" format. These "elf" files can be read in the test-benches and simulated in VCS.

As for accelerator, the VHDL RTL-level code is integrated with the RISC-V core for both functional simulation (VCS) and synthesis (DC). The synthesized gate-level netlist (core plus accelerator) is written in Verilog to be simulated in VCS to determine the switching activity (Switching Activity Interchange Format, or SAIF, file). The SAIF file is used to estimate the power dissipation of the synthesized circuit in DC.

B. RISC-V Architecture

For the first implementation of the RISC-V plus accelerator SoC, we selected to have a single 64-bit RISC-V core with the minimum functionality and no floating-point unit. Also caches were not synthesized and used only for simulation purposes: functional simulation and SAIF file generation.

Fig. 4 shows the architecture of the RISC-V and the accelerator. The key element is the RoCC (Rocket Custom Co-processor) interface that provides the communication between the accelerator and the core and D-Cache. The designer can define custom instructions with a special opcode to be passed at the accelerator. Moreover, the RoCC interface handles the accelerator's load/store requests.

C. Supported Applications in Accelerator

In this first version, the DSP accelerator provides configuration for three typical applications, derived from [9]: mono and bi-dimensional (image) filtering and pattern matching.

1) FIR (mono-dimensional) filter: The mono-dimensional convolution for a maximum order of 64

$$y(n) = \sum_{k=0}^{63} a_k x(n-k) \tag{1}$$

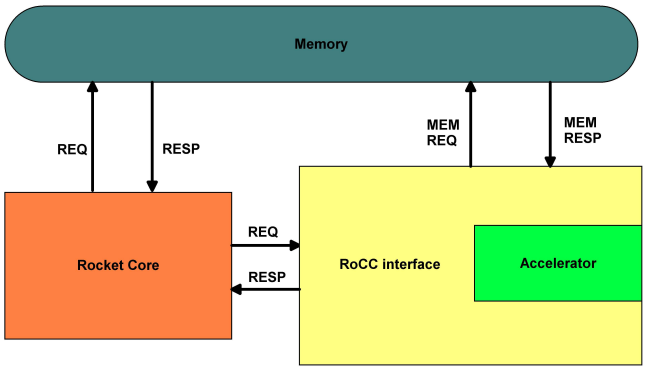

Fig. 4. Architecture of RISC-V and RoCC interface to the accelerator.

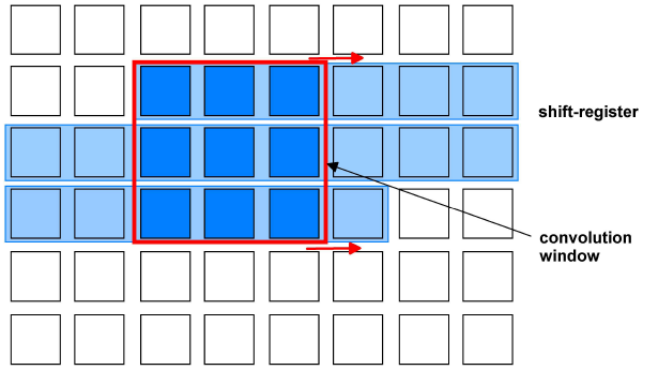

Fig. 5. Convolution window (3×3) for image filtering.

is realized in FIR direct form by the datapath of Fig. 2. The dynamic range is adjusted by DPA-I according to the characteristics of the filter.

For this application, the coefficients a_k are loaded in register A and the samples $x(n-k)$ are fed into register B one per clock cycle. The PEs are all set in multiply mode.

2) Bi-dimensional Convolution: The bi-dimensional convolution is used to process an image in the space domain. Averaging, smoothing and sharpening of a digital image are done by the convolution of the matrix of pixels representing the image $B(n \times n)$ with a specific mask $W(m \times m)$. For example, if $m = 3$ the value of the new pixel is computed as:

$$B'[x,y] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} B[x+i, y+j] \cdot W[i,j] \qquad (2)$$

In the accelerator the shift register A is used to store the coefficients of the mask W. Using an approach similar to that in [10], we can extract windows of pixels from a single data stream, as in the case of the mono-dimensional filter. Fig. 5 shows how the method works. Pixels are fed line by line, from top to bottom, until two complete lines and three pixels of a third line are stored in the shift register B (maximum size 64 pixels). At this point, the first 3×3 convolution can be performed, and, from this moment on, each new pixel inserted in the shift register moves the convolution window one position. Because B can store up to 64 pixels, we have to cut the image in vertical bands 30-pixel wide and process a vertical band at the time. The repositioning at the beginning of a new band is done by the RISC-V core which computes the address of the first pixel in the new band and starts the new convolution in the accelerator.

3) Pattern Matching: We consider the problem of detecting from a stream of data a sequence of symbols, where each symbol is represented by a word of n bits. If, as an example, we consider a stream of ASCII characters, we want to detect the 8-character sequence *"AB*A***D"*, in which the wildcard (*) indicates that any character can be in that position. A simple approach to perform the task in the DSP accelerator is the following: 1) Load the sought pattern in register A. Load a zero when there is a wildcard. 2) Set the PEs to perform the following operations: multiplication in correspondence of wildcard/zero, subtraction otherwise. 3) Feed the character stream into shift-register B. If the output is zero the pattern is matched.

V. Experimental Results

In this section, we describe the experiment run on the SoC. The experiment is limited to the trade-offs of a software implementation of a bi-dimensional image smoothing filter run in the RISC-V core, and the same filtering executed on the hardware accelerator.

The synthesized designs are implemented in a commercial 28 nm FDSOI (fully-depleted silicon-on-insulator) library of standard cells. We set as a target clock period of 2 ns (500 MHz) and verified that both the RISC-V core and DSP accelerator could meet the timing constraints. The caches and the memory are not synthesized, but used only in simulation to generate the switching activity files.

Since the DSP accelerator is developed in VHDL, to integrate it in the Chisel Rocket environment we need a *"wrapper"* to connect the signals between the RoCC interface and the accelerator. The wrapper includes two First-In-First-Out (FIFO) buffers to synchronize the data read/written in memory to the accelerator operations and a controller to supervise the operations (Fig. 6). For example, when the input FIFO is empty due to a cache miss, the accelerator is stopped. The cache parallelism is 64 bits and the data serialization to 8 bits for image processing is done in the accelerator in Fig. 6.

The architecture of the accelerator is that of Fig. 2. This choice is compatible with most digital filtering operations. The input dynamic range is 16 bits for both A and B and the internal dynamic range for the PEs' output (when multiplication) and the adder tree is 32 bits. The range is configurable by DPA-I forcing zero in the input registers and at tree's inputs.

The implementation results for the DSP accelerator (without wrapper) are reported in Table I. We estimated the average power dissipation (P_{ave}) for the three supported applications. The dynamic range in the table refers to the active bits in the accelerator, the rest of the datapath is deactivated by DPA-I.

Next, we integrate the accelerator, the wrapper and the RISC-V core. The core area is 31,150 μm^2 and it is signifi-

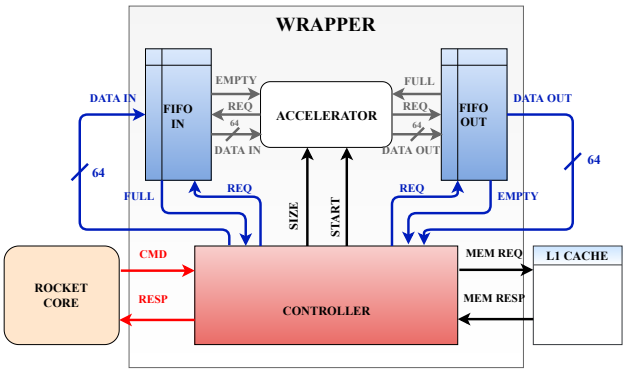

Fig. 6. Accelerator interface and controller.

TABLE I
RESULT OF THE IMPLEMENTATION OF THE DSP ACCELERATOR.

Area [μm^2]	f_{max} [MHz]	Application	dynamic range	P_{ave} [mW]
96,500	500	FIR low-pass	24	12.4
		smoothing 2D	16	6.7
		char. matching	8	4.2

P_{ave} at 500 MHz.

cantly smaller than the accelerator. The accelerator comprises 64 16×16=32-bit multiply-add-units (about 1,100 μm^2 each). The wrapper is about 11,500 μm^2 for a total area of about 139,000 μm^2, excluding caches.

In the first test (called SW), we run the 2D convolution of images in the core only. The arithmetic operations of (2) are executed in the integer ALU and Mult/Div units in the core. In the second test (called ACC), the convolution is run on the SoC in Fig. 6.

Fig. 7 shows the schedule of the execution of the two tests. We consider only the portion of energy necessary for the processing. We consider a smoothing filter mask 3×3 and pictures of size 8×8, 30[1]×30, and 30×128.

Table II reports the results of the tests on 2D convolution. Although, the power dissipation of the "Chipset ACC" is almost double that of the RISC-V core only, the impressive speed-up of the acceleration makes the ACC much more power efficient. The energy consumption for the processing is reduced by 40–50 times with respect to the software execution.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented an accelerator for the RISC-V ecosystem, suitable for digital FIR filtering, bi-dimensional convolution and pattern matching. The accelerator is based on a datapath that is reconfigurable in terms of dynamic range and type of operations. The reconfigurable accelerator consists of 64 processing elements, and the maximum dynamic range is 32 bits. The RISC-V core and accelerator have been implemented in a 28 nm FDSOI library of standard cells.

[1]For 64 PEs, the maximum window width for 3×3 mask is 30 pixels.

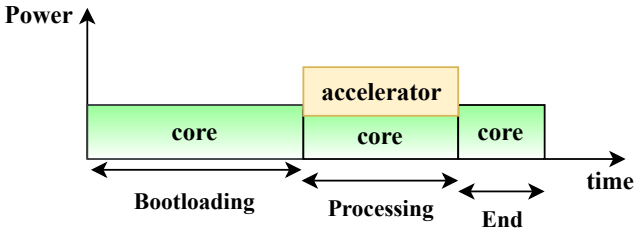

Fig. 7. Energy to run the application.

TABLE II
RESULT OF IMPLEMENTATION OF CHIPSET SW AND ACC FOR IMAGE FILTERING.

	ACC	SW	ratio			
P_{ave}	21.47	11.34	1.89	at 500 MHz		
Image size	n. cycles ACC	n. cycles SW	speed-up	Energy-per-Image [nJ] ACC	SW	ratio
8x8	2,119	170,047	80	91	3,857	42.4
30x30	7,440	699,747	94	319	15,870	49.7
30x128	31,803	3,186,074	100	1,366	72,260	52.9

We ran some experiments to evaluate the performance and the energy consumption of the accelerator chipset and we compared them to the results obtained by the software execution in the RISC-V core.

The speed-up of the accelerated chipset is such that, even a large power dissipation in the extra gates in the accelerator, do not compromise energy savings, confirming that: "low latency is low energy".

In future work, we plan to extend the capabilities of the accelerator by improving its flexibility to be able to run more applications on it.

REFERENCES

[1] AnandTech. "Chipworks Disassembles Apple's A8 SoC". [Online]. Available: https://www.anandtech.com/show/8562/chipworks-a8
[2] T. Chen, A. Rucker, and G. E. Suh, "Execution time prediction for energy-efficient hardware accelerators," in *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2015, pp. 457–469.
[3] S. Steffl and S. Reda, "LACore: A Supercomputing-Like Linear Algebra Accelerator for SoC-Based Designs," in *2017 IEEE International Conference on Computer Design (ICCD)*, Nov 2017, pp. 137–144.
[4] F. Farshchi, Q. Huang, and H. Yun, "Integrating NVIDIA deep learning accelerator (NVDLA) with RISC-V soc on firesim," *CoRR*, vol. abs/1903.06495, 2019. [Online]. Available: http://arxiv.org/abs/1903.06495
[5] Open Celerity Cores. [Online]. Available: http://opencelerity.org/
[6] Blue Spec RISC-V Cores. [Online]. Available: https://bluespec.com/riscv-acceleration-factory/
[7] M. Petricca, G. C. Cardarilli, A. Nannarelli, M. Re, and P. Albicocco, "Degrading Precision Arithmetic for Low Power Signal Processing," in *Proc. of 44th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2010, pp. 1163–1167.
[8] UC Berkeley Architecture Research. "Rocket Chip Generator". [Online]. Available: https://bar.eecs.berkeley.edu/projects/rocket_chip.html
[9] G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re, "Residue Number System Reconfigurable Datapath," *Proc. of IEEE Int.l Symposium on Circuits and Systems (ISCAS 2002)*, vol. 2, pp. 756–759, 2002.
[10] B. Bosi, G. Bois, and Y. Savaria, "Reconfigurable Pipelined 2-D Convolvers for Fast Digital Signal Processing," *IEEE Transactions on VLSI Systems*, pp. 299–308, Sept. 1999.