# Metrics for Digital Signal Processing Architectures Characterization: Remanence and Scalability

Pascal Benoit, Gilles Sassatelli, Lionel Torres, Didier Demigny, Michel Robert, Gaston Cambon

# Metrics for Digital Signal Processing Architectures Characterization: Remanence and Scalability

Pascal Benoit[1], Gilles Sassatelli[1], Lionel Torres[1], Didier Demigny[2], Michel Robert[1], and Gaston Cambon[1]

[1] LIRMM, UMR UM2-CNRS C5506,
161 rue Ada, 34392 Montpellier Cedex 5, France
(33)(0)4-67-41-85-69
{first_name.last_name}@lirmm.fr
[2] ETIS, UMR-CNRS 8051,
6 av. du Ponceau, 95014 Cergy Pontoise Cedex, France
(33)(0)1-30-73-66-10
{name}@ensea.fr

**Abstract.** SoCs became reality: an increasing number of products powered by this type of circuits hits the market. Reduced power consumption, increased performance are some of the usually stated benefits. Besides approaches aiming at enabling system level exploration for multiple million gates designs, like the SystemC initiative, choosing the right IP core, or the right set of parameters among those available is not straightforward. In this article we first present a generic model for digital signal processing architectures. Several metrics, later referred as Remanence and Operative Density are presented in this paper. The methodology is illustrated through a case study.

## 1 Introduction

Among the last couple of years lots of new approaches appeared [1][2]. Real innovations like coarse grain reconfigurable fabrics [2] or dynamical reconfiguration have brought numerous improvements, solving several weaknesses of traditional FPGA architectures. Besides this point, several recurrent issues remain, and the proliferation of architectures lays to an additional problem for SoC designer: choose the right IP core for a given set of specifications. Despite some works have already proposed some useful tools, like the Dehon metric [3], allowing to compare the computing density for different architectures in different silicon technologies, the need of additional metrics is now obvious. The goal of this paper is to address this characterization problem by the way of defining two metrics: remanence and scalability allow to compare more efficiently different architectures dedicated to digital signal processing.

## 2 Digital Signal Processing Architectures

Each architecture dedicated to digital signal processing exhibits benefits as well as limitations, extensively listed in [1] for Von Neumann architecture and [2][8] for re-

A. Pimentel and S. Vassiliadis (Eds.): SAMOS 2004, LNCS 3133, pp. 128–137, 2004.

configurable architectures (RA). Figure 1 depicts a general model for both processors and RAs. Depending on the architecture, each constituting element differs.

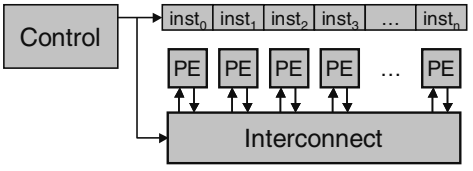


**Fig. 1.** The Generic model of DSP architectures

The constituting elements are:

- interconnect subsystem,
- array of processing elements (PE), PE structure,
- control unit,
- instruction / configuration memory.

The following architectures can be modeled:

- Processors are based on the Von Neumann paradigm [5]. Operation execution is carried out in the data path in a sequential way. Usually a single PE is present. The configuration memory is no more than a single register storing the current instruction.
- VLIW DSP: they carry out parallelism at the instruction level. A VLIW instruction consists of several RISC instructions, each one being executed in one PE.
- Fine grain RAs like FPGAs. The PE array is two-dimensional. Each PE features bit-level reconfigurable logic, often Look-Up-Table based. In most devices, no controller is present, the configuration being uploaded in the configuration memory offline.
- Coarse grain RAs Each PE often features hardwired arithmetic operators (coarse grain) instead of bit-level reconfigurable logic.

## 3 Remanence and Scalability

### 3.1 Remanence

A RA is constituted by a set of operators $N_a$ running at the clock frequency $F_e$. Each architecture is able to reconfigure $N_c$ operators each configuration cycle of frequency $F_c$. $F_c$ may be different from $F_e$, depending on the considered architecture. The *remanence* is defined by the following expression:

$$R=\frac{N_a.F_e}{N_c.F_c}$$

The remanence [6] subsequently characterizes the dynamical character of the RA by reporting the number of cycles needed to reconfigure the PE array. This criteria provides an information on the minimal amount of data to be processed between two configuration cycles.

– If the configuration phase is shadowed, a new configuration is loaded during processing. The configurations are then switched within the next clock cycle. The architecture is efficient if during this cycle most of the operators are processing data.
– If the configuration phase is not shadowed, the number of processing cycles must be greater than R for a limited overhead: usually in the range of 10 to 20 times R.

Moreover, a data parallelism of β (β data processed concurrently) increase according to a factor β the minimal number of data to be processed between two configuration cycles. Therefore, the ratio between the amount of data to be computed and R figures out an important information which helps to choose between data or instruction parallelism. Besides this point, one can notice that 1/R is a metric assessing the dynamical character of an architecture. The less R, the more dynamically reconfigurable the architecture is. The system reconfiguration frequency is lower to $F_e/R$.

This metric has three main advantages:

– It reports the dynamical character of an architecture independently from its granularity: The operators can either be fine grain (CLBs) or coarse grain (multipliers, ALUs). This is enabled thanks to the use of the concept of operators instead of any lower-level consideration.
– Although some architectures provide only inter-operators path routing, this implies to stop processing while configuring. Hence, it is functionally equivalent to reconfigure the operators. It can nevertheless be more efficient to directly reconfigure the operators. For a given processing power, $N_c$ can be greater or/and require less configuration bits. This it implicitly taken into account by the remanence, thanks again to the concept of operators.
– No matter how the reconfiguration takes place. It can be done in a single pass, after the processing related to the current configuration is done, or continuously, a few operators being reconfigured each cycle while processing keeps on.

*Remanence and power consumption.* In a processor, up to 50% of the power is consumed in the control unit. Reconfiguration frequency and volume (i.e. number of bits) might consequently impact on the power consumed. Some architectures providing a 'freeze' mode (configuration frozen during a given time) can achieve interesting power savings.

The processing power $P_{proc}$ of a given architecture can be expressed as the product between the number of operators $N_a$ and the clock frequency $F_e$ ($P_{proc} \sim N_a.F_e$)The power consumed can then be expressed as:

$$P_{cons} \sim N_a.F_e.U^2$$

with U being the voltage supply. According to this formula, equivalent power saving might be achieved by either optimising $N_a$ or $F_e$. However, decreasing the clock frequency allow to decrease proportionally the voltage supply. Let assume that, the power consumed is :

$$P_{cons} \sim N_a.F_e^3$$

Then the ratio $P_{cons}/P_{proc}$ grows according to a factor $F_e^2$. For a given processing power, it is then worthwhile to increase the number of operator and reduce accord-

ingly the clock frequency. Nevertheless, applying such an approach might increase consequently the control unit complexity and then its power consumption. This observation figures out clearly the significance of the remanence. The power consumed is proportional to the bit switching activity (each second). Hence, it is possible to define a cost in power consumption per MIPS by the way of considering both processing-related cost and configuration-switching cost.

### 3.2 Scalability

Due to the continuous technology scaling, scalability is today becoming a key issue; the problem can be stated as follows: given a customisable architecture model (in terms of number of PEs), how does the $N_a/A$ ratio grow, $N_a$ being the number of PEs and A the core area. We define the operating density Do as the ratio $N_a/A$. Hence, for an architecture fully scalable OD ($N_a$) will be constant.

Accordingly to our general model (figure 1), and assuming the core area as the sum of the constituting elements' area, architecture scalability analysis sum up to each component scalability analysis:

$$OD \sim \frac{N_a}{A_{PEs} + A_{control} + A_{config\_mem} + A_{interconnect}}$$

## 4 The Systolic Ring

The Systolic Ring architecture features a DSP-like coarse grain reconfigurable block; following an original concept (figure 2). The configuration (microinstruction code) can either come from the configuration layer (FPGA-like mode, *global mode*) or from a local sequencer (*local mode*) depicted in figure 2.

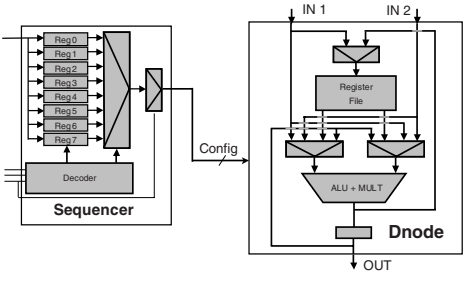


**Fig. 2.** The Dnode architecture

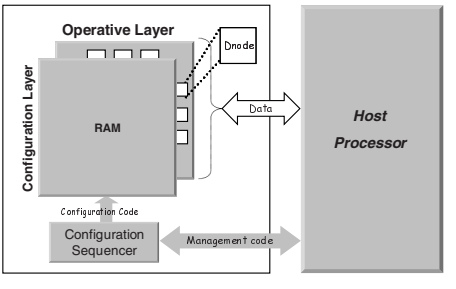


**Fig. 3.** System overview

A custom instruction set RISC processor (*configuration sequencer*, figure 3) is also used in order to upload the microprograms into the local sequencers of the Dnodes set to *local mode*. It is also used to write the configuration into the configuration layer (*global mode*).

The specific structure of the operating layer is depicted on figure 4a. The Ring topology allows an efficient implementation of pipelined datapath. The switch compo-

nents establish a full connectivity between two layers, refer to [4] for complete description. The Systolic Ring also provides a feedback network that proves useful for recursive operations. It allows to feedback data to previous layers by the way of using feedback pipelines implemented from *each* switch in the structure. Each other switch in the architecture has a read access on *each* other switch's pipeline. Figure 4b depicts the east switch's feedback pipeline. In addition a bus connecting all switches in the architecture and the global sequencer is available, mainly for conditional configuration: a data computed in the operating layer can be retrieved in the configuration sequencer for further analysis and thus different configuration evolution.

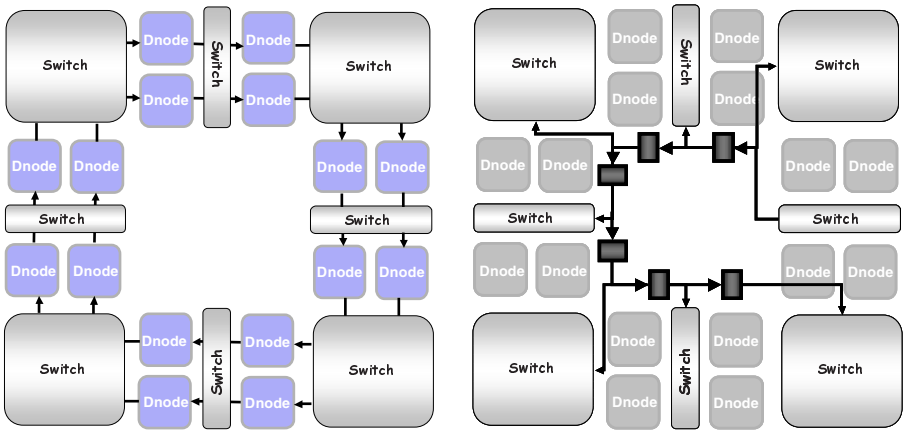


**Fig. 4.** Operating layer (a) and a switch's feedback pipeline (b)

The program running on the global sequencer is able to modify the configuration of an entire Dnode layer (2 Dnodes on the 16 Dnodes Systolic Ring depicted on figure 4) each cycle. Up to 12.5% of the Dnodes can be reconfigured each cycle in the exposed version, but this can be tailored, especially when C/N vary, C being the number of Dnodes per layer and N the number of layers.

## 5 Remanence and Scalability Analysis

### 5.1 Remanence Analysis

Considering the *global mode*, the remanence of the Systolic Ring (Figure 4) is $R_{sring\_static}$=8, 8 cycles are indeed needed to reconfigure the whole structure, $F_e$ being equal to $F_c$. As previously said, the Systolic Ring is customisable, thus the remanence can be tailored. This of course impacts the instruction size, and other parameters like memory bandwidth. This will be pointed out in the scalability section.

The *local mode* allows to change the configuration of each Dnode of the structure each cycle (assuming that all Dnodes are in local mode). However, 8 configuration

cycles are needed to store a maximum length microprogram (one local sequencer register loaded per cycle, Figure 2), this microprogram being considered as a single Dnode configuration. In this case, a maximum of 64 cycles are needed, thus $R_{sring\_dynamic}$=64.

It must be pointed out that:

- A microprogram being considered as a single instruction, 8 instructions are needed to carry out a single data. Therefore, $R_{sring\_static}$ only characterizes the amount of data.
- Despite in local mode all Dnodes can modify their configuration each cycle, from a system point of view, only $R_{sring\_dynamic}$ should to be taken into account. This mode is worthwhile only when the number of cycles of the considered process is at least 10 times greater than $R_{sring\_dynamic}$. The global mode is of great interest for data parallelism while the local mode features intermediate granularity data parallelism and potential instruction parallelism.

Table 1 gives remanence values for three different architecture described below:

- Texas Instruments TMS320C62: this DSP is a powerful VLIW processor featuring 8 processing units. It reaches 1600 MIPS (max power) when running at 300MHz. The remanence $R_{C62}$ is equal to 1: it is able to reconfigure all its processing units each cycle.
- Xilinx Virtex XC2V2000 FPGA [7]: this one is partially reconfigurable, and requires 14.29 ms to be totally reconfigured at $F_c$=66 MHz. While $F_e$ is application-dependant, the ratio $F_e/F_c$ is non constant. Results depicted in table 1 are given for $F_e$=100 MHz.
- Systolic Ring: a 16 Dnodes realisation, described above in section 4.

**Table 1.** Remanence comparisons

| | **TMS 320C62** | **Xilinx XC2V2000** | **Ring-8** | |
|---|---|---|---|---|
| | | | **Dynamic** | **Static** |
| **Number of op.(Na)** | 8 PEs | 2688 CLBs | 8 Dnodes | |
| **Reconfigured op. / cycle** | 8 | $2.8.10^{-3}$ | 0.25 | 2 |
| **Fe/Fc** | 1 | 1 ($F_e$=66MHz) | 1 | 1 |
| **Remanence (R)** | 1 | 936540 | 64 | 8 |

As shown in table 1, the remanence of the Systolic Ring in full global mode (i.e. static) is 8, as to say, 8 cycles are required to fully reconfigure the structure. The Systolic Ring also provides a *hybrid mode*, allowing to set independently each Dnode in the structure in global or local mode. In this last case, the effective remanence is ranging from $R_{sring\_static}$ to $R_{sring\_dynamic}$. The most 'dynamically reconfigurable' architecture is however the VLIW processor. Hence, its use should be recommended for rela-

tively irregular applications implying instruction-level parallelism. The remanence however does not give the number of PEs that one can expect to have for a given silicon area : the scalability analysis addresses this problem.

### 5.2 Scalability Analysis

As assumed in 3.2, the total area is approximated by the sum of the 4 constituting elements of our model. Two different scaling techniques are to be considered:

*Scaling technique 1: N/C tradeoff*
$N_a$ can be tailored between N (number of Dnodes per layer) and C (number of layers) according to the formula: $N_a = N.C$

Increasing N will encourage parallelism level (either instruction or data) while increasing C will improve pipeline depth (i.e. computation parallelism).

*Scaling technique 2: MIMD approach*
It is also possible to increase $N_a$ by the way of using multiple Systolic Rings witch will lead to a MIMD (Multiple Instructions Multiple Data) like solution. This technique provides a maximal scalability, as the resulting silicon area will be proportional to the number $N_a$ of PEs.

$$\left(\frac{N_a}{A}\right)_{MIMD} = \alpha = cte$$

In the following, only scalability issues related to technique 1 will be considered.

***Processing elements (i.e. Dnodes)***
Given a PE of core area A, the instantiation of $N_a$ PEs occupies $\alpha A$ area units on the die, as to say this part is fully scalable, independently from the N/C ratio:

$$\left(\frac{N_a}{A}\right)_{PEs} = \alpha = cte$$

***Control unit***
The global sequencer is a simple RISC processor featuring a specific instruction set. The 16 lower bits of the instruction format are dedicated to internal RISC management, whereas the upper ones are directly addressing a given Systolic Ring layer (configuration of N Dnodes and the corresponding switch). Figure 5 depicts the format of the instruction register used in the configuration sequencer.

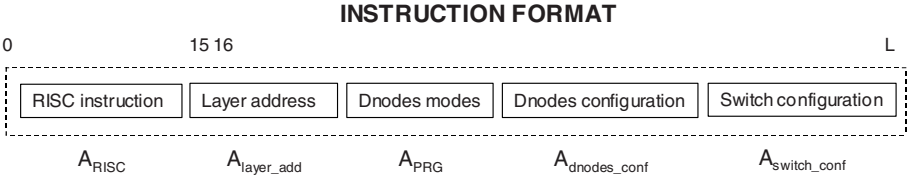


**Fig. 5.** RISC instruction format

The area $A_{part}$ corresponding to a given *part* of the instruction register will be considered proportional to the number of bits required for its coding, $M_{part}$.

– $A_{RISC}$. The size of the sequencer-related instruction is constant, thus, fully scalable.

$$A_{RISC} \sim M_{RISC} = 16$$

– $A_{layer\_address}$. $M_{layer\_address}$ bits being required for a C-layer addressing ($2^M$=C), and taking into account that C may not be a power of two:

$$A_{layer\_address} \sim M_{layer\_address} = \left|1 + \log 2\,(C\text{-}1)\right|$$

– $A_{PRG}$. 2 bits are required to code the 4 run-modes. Hence, for N Dnodes, the required number of bits given above exhibit a maximal scalability:

$$A_{PRG} \sim M_{PRG} = 4.N$$

– $A_{dnodes\_conf}$. Again, considering that 17 configuration bits are required for each Dnode, the resulting area is:

$$A_{Dnodes\_conf} \sim M_{Dnodes\_conf} = 17.N$$

– $A_{switch\_conf}$. In order to provide a full inter-layers connectivity, let n be the number of inputs of the MUX and p the number of outputs: C(n,p) addresses combinations must be supported. The availability of a bus implies to be able to write the result of any Dnode output, plus an additional bit putting the bus driver in high impedance. The resulting number of bits required is:

$$A_{switch\_conf} \sim M_{switch\_conf} = \left|1+Log_2(C_n^p\text{-}1)\right|+\left|Log_2(N)+1\right|$$

The number of inputs is determined by the expression:

$$n = 2.N + (C\text{-}1).N + 1$$

The first term is related to number of Dnodes of the upper layer, while the second is related to the feedback network: C-1 feedback network are implemented, each one constituted by the aggregation of N Dnodes outputs. The number of outputs p is equal to N (number of Dnodes per layer).

***Configuration Memory***

The use of a coarse grain technology drastically decreases the size of the configuration memory. In addition, the size of the PE-only configuration memory grow linearly with the number of PEs. Only the routing-relative configuration size grows non-linearly with respects to the number of processing elements, due to the fact that the Systolic Ring provides full interlayer connectivity. The size required for the storage of a (N,C) version of the Systolic Ring is:

$$A_{config} \sim M_{config} = C.(M_{PRG}+M_{Dnodes\_conf}+M_{switch\_conf})$$

## 6 Case Study: Tailoring the Parameters

Let us suppose that the processing power needed (proportional to the number of Dnodes) corresponds to a number of Dnodes between 40 and 60. First, a C/N ratio must be selected. This ratio will be set according to the targeted applications. Let us also suppose that promoting pipeline degree seems to be more attractive for the targeted applications. Therefore, we choose a C/N ratio equal to 4. Following the area evolution as a function of C and N for one instance of the Systolic Ring, corresponding curves are plotted for several instances of the Systolic Ring. The figure 6 represents the corresponding curves. The constraint tube is then plotted for the C.N values chosen (40 and 60). In order to show the architecture dynamism, we also plot the *Remanence* curve and add it to the graph. The figure 6 shows how a Systolic Ring user can tune the architectural parameters. Indeed, in the constraint tube, many solutions are possible. Choosing for example eight Systolic Ring instances (40 Dnodes total) provide the smallest silicon area in the constraint tube. Moreover, it also will be the most dynamical solution, *i.e.* the one requiring the minimum number of cycles to configure the 40 Dnodes. However, it will be also the solution offering the worst interconnection resources (implying inter-Systolic Ring communications) and the worst processing power. At the opposite, choosing only one instance of the Systolic Ring with a total number of 60 will significantly increase the processing power (1.5 times) but also the silicon area. However, more interconnection resources will be available due to the full layer interconnectivity allowed in a single Systolic Ring instance. The increase of operators will also involve a higher *Remanence*.

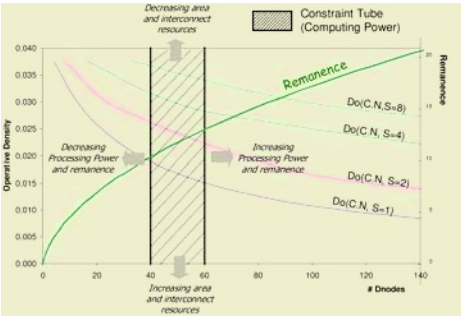


**Fig. 6.** Processing power constrained

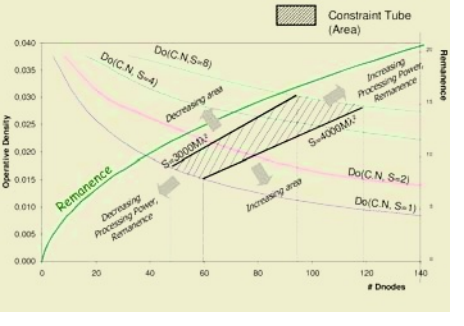


**Fig. 7.** Area constrained

Let us now suppose that the available silicon area is between 3000Mλ² and 4000Mλ² (λ being the half width of the transistor channel). This design space defines the constraint tube. As mentioned previously, the C/N ratio must be selected. This ratio will be set according to the target applications. We choose here for example a C/N ratio equal to 4. Following the area evolution as a function of C and N for one instance of the Systolic Ring, corresponding curves are plotted for several instances of the Systolic Ring. The figure 7 depicts the related curves. The constraint tube is then plotted according to the area constraints (the two lines were extrapolated from the Systolic Ring area formalisation). The *Remanence* curve is also plotted.

The figure 7 shows how a Systolic Ring user (a platform-based designer for example) can tune his core with the architectural parameters. Indeed, in the constraint tube, many solutions are possible**.** For the lower bound (3000Mλ²), tradeoffs ranging from 45 to 90 Dnodes are available. This characterizes an increased operative density. This is allowed by the way of using eight instances of the Systolic Ring instead of only one. However, this multi-instantiation implies a reduced connectivity between the Dnodes of the architecture and an increased *Remanence*. For the upper bound (4000Mλ²), the processing power can also be doubled by the same means, implying the same consequences.

## 7 Conclusion

After having compared different architectures and shown the limitations of classical comparison approaches, we have presented a general methodology for the characterization of architectures dedicated to digital signal processing. This methodology is based on evaluation metrics, *Remanence* and *Operative Density*, as functions of the architecture parameters. This methodology helps the designer to choose between several architectural trade-offs, as shown for the Systolic Ring example. The architecture presented in the first section, was used as a case study for both *Remanence* and *scalability* analysis. These considerations helped to determine architecture trade-offs and also contributed to establish the limitations of the architecture considering a set of application-relative constraints (parallelism type, area, processing power). Future works take place in analysing other crucial factors in a SoC design context such as the power consumption.

## References

1. W. H. Mangione-Smith et al, "Seeking Solutions in Configurable Computing," IEEE Computer, pp. 38-43, December 1997
2. R. Hartenstein, H. Grünbacher: The Roadmap to Reconfigurable computing Proc. FPL2000, Aug.27-30, 2000; LNCS, Springer-Verlag
3. André DeHon, “Comparing Computing Machines”, Configurable Computing: Technology and Applications, Proc. SPIE 3526, 2-3 November 1998.
4. G. Sassatelli, et al.: “Highly Scalable Dynamically Reconfigurable Systolic Ring-Architecture for DSP applications”, IEEE Design Automation and Test in Europe (DATE’02) , pp. 553-557, mars 2002, Paris, France.
5. G. Sassatelli, “Architectures reconfigurables dynamiquement pour les systèmes sur puce”, Ph.D. thesis, Université Montpellier II, France, April 2002.
6. D. Demigny, et al.: «La rémanence des architectures reconfigurables, un critère significatif des architectures», proc. of JFAAA, pp. 49-52, décembre 2002, Monastir, Tunisie.
7. Xilinx, the Programmable Logic Data Book, 2002.
8. D. Demigny, et al.. «Architecture à reconfiguration dynamique pour le traitement temps réel des images» Techniques et Science de l’Information Numéro Spécial Architectures Reconfigurables, 18(10) : 1087-1112, décembre 1999.