

# An optimal architecture for a multi-standard reconfigurable radio: Cost-minimising common operators under latency constraints

Virgilio RODRIGUEZ, Christophe MOY, Jacques PALICOT

SCEE Laboratory

IETR/Supélec

Cesson-Sevigne, France

e-mail: vr <at> ieee.org, {christophe.moy, jacques.palicot}@supelec.fr

**Abstract**—We build a mathematical model to determine an optimal architecture for a multi-standard reconfigurable radio. We examine the trade-off between (a) building complex dedicated functional modules providing high performance at a high cost (as well as size and weight) (“Velcro approach”), versus (b) relying on simpler lower-level components, which reduces cost but increases system latency. On the foundation of the “common operators” approach, we describe a procedure that identifies an architecture that minimises the cost of the radio, while keeping its latency under specified limits.

## I. INTRODUCTION

We build a mathematical model to identify the optimal architecture for a multi-standards reconfigurable radio. The basic trade-off we examine is that of the (monetary) cost of a multi-standard reconfigurable radio versus its performance. We find an architecture which minimises the cost, while observing performance (computational time) constraints.

We model the radio as a hypergraph of progressively simpler functional modules. The functionality of one such module can be provided either through a self-contained component, or through the invocation of simpler (lower level) modules. A self-contained component is an optimised hardware/software combination built to perform a task in the most efficient way. One such component could be an equaliser, for example. Simpler, lower-level components are generally less expensive to build, and can be reused by several upper-level modules inside and across standards. The use of lower-level components reduces the manufacturing cost, and quite possibly the size and the weight of the radio. Unfortunately, such use generally increases the execution time of the concerned task. As a consequence, the total execution time of a given operation may exceed practical limitations.

Thus, we see the design of a multi-standard reconfigurable radio as choosing the optimal point between two extremes. At one extreme is the Velcro approach: to install self-contained complex communication modules each exclusively dedicated to a given standard. At the other extreme, we can attempt to build the entire multi-standard radio through very simple components (adders, multipliers, MAC, etc) that are invoked by

more complex modules to perform the various communication tasks necessitated by the supported communication standards. The Velcro approach will generally provide the best performance, but at the highest cost (and probably greatest size and weight). Conversely, by going to the other extreme, we can minimise the (monetary) cost (and the size and weight) of the radio, but at a performance level that may be unacceptable. Thus, we need to find the right level of complexity for the various modules that gives us the best trade-off between performance and cost.

This study is based on the “common operators” approach to the design of reconfigurable equipment. Its main principle is the identification and (re)use of common operators that can each match several processing contexts by a simple parameter adjustment. To achieve, from this perspective, an optimal architecture capable of supporting several communication standards, one must identify an optimal level of complexity for various functional modules. The selected modules may be simpler than a self-contained module that implements a major communication task (such as equalisation or modulation), but more complex than primitive operators such as AND, OR, adder, etc. This approach can greatly increase the efficiency of a multi-standard software-defined radio, both in terms of manufacturing cost, and of the speed of reconfiguration during operation.

The common operators approach is discussed more extensively in [1], a work *not* focused on architecture optimisation. In [2], a parallel strand of work, we followed an approach similar to the present one. However, [2] combines economic and latency considerations into a single cost function. This combination (a weighted sum) is a reasonable first step, as it simplifies the exposition and the solution algorithm. But unfortunately, the combined cost function has some drawbacks: it adds the monetary cost (paid once by the designer) with the “delay costs” incurred by the user each time it executes a standard throughout the useful life of the radio, it fails to account for the hard time constraints often arising from communication applications, and makes the chosen design highly dependent on the weights (which are themselves arbitrary). In the present paper, we minimise the (monetary) cost only.

Latency plays a role, as a constraint that cannot be exceeded while performing certain operations. Neither [2] nor the present work addresses the identification of new common operators useful to the design of multi-standard reconfigurable radios. Such identification is, in its own right, an active area of research. Researchers are proceeding along several directions. For instance, [3] shows that many important tasks of a communication receiver can be implemented through the fast Fourier transform (FFT). In turn, the FFT can be implemented via the butterfly operator, and, as argued more recently[4], via CORDIC. With this in mind, some researchers are seeking frequency-domain implementations for different families of algorithms. For example, [5] studies the frequency domain implementation of Reed-Solomon channel decoding.

Relevant works describing interesting approaches to the design of reconfigurable radios include: [6], [7], which argue for parametrised design from a “common functionality” perspective, as originally advocated in [8]; [9] whose approach is inspired by object-oriented programming; [10] which attempts to cover hardware and software design under a common methodology; and [11] which proposes designs that integrate the entire system on a chip (SoC).

The rest of the paper proceeds as follows. First, we build the mathematical model. This step includes the drawing of a graph that represents design alternatives, the consideration of possible performance metrics, and the specification of a cost function and appropriate constraints. Subsequently, we discuss the optimisation procedure, whose results we give and discuss in an immediately following section. Finally, we conclude with a discussion addressing further interpretations of our results, as well as limitations and future directions.

## II. MATHEMATICAL MODEL

### A. Graph-theoretic representation

As illustrated by figure 1, we represent a multi-standard radio as a hypergraph of progressively simpler functional modules. Each node (module) represents a functionality that can either be implemented via a dedicated hardware/software component (an ASIC for example), or can be achieved by invoking lower-level modules. The hyperarcs leaving a node (parent) specify the simpler modules (descendants) that could provide the required functionality through multiple calls. Descendant nodes may not all be at the same level.

An OR arc (direct arrow) means that only one of the descendant nodes is necessary to implement the functionality of the parent node. An AND arc (inverted Y connection as pointing from S3 to A4 and A5) means that *all* descendant nodes are needed to implement the functionality of the parent node. Note that in some cases, a parent node may have both AND and OR dependencies with its descendants. The roots of this graph, at the highest level, represent the standards to

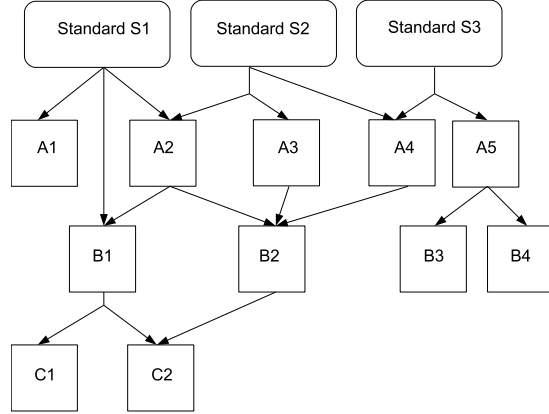


Fig. 1. Graph corresponding to a conceivable tri-standard reconfigurable radio

be supported by the radio. Below, we do *not* consider the possibility that the graph may be cyclic, which would complicate the exposition and analysis.

### B. Optimisation parameters

The decision to provide a functionality via a self-contained, dedicated component or by invoking multiple times simpler, reusable components is determined by two key considerations: (monetary) cost and (execution) time.

The monetary cost of a component which is paid only once during the useful life of the radio) represents the total cost of including the component in the design. In some software-defined radio (SDR) architectures, the monetary cost can be represented by (is proportional to) the number of logic units necessitated by an FPGA implementation. It is best to take the view point of a system integrator that “outsources” the components (software or hardware). Then, the monetary cost represents the fair-market value, at design time, of acquiring the finished component in the open market. Execution time (incurred every time a component is employed throughout the life of the system) is also a critical consideration, because communication standards may impose hard time constraints (deadlines) for the completion of certain operations.

Determining the “deadlines” to be observed while designing the radio to support a given standard is non-trivial and should be done with great care. If deadlines are set too high, they may lead to a design that fails to perform at acceptable levels in practical situations. But if deadlines are set lower than necessary, the resulting design will be more expensive than necessary. The key to determining the right delay tolerances is to examine the “transmission chain” (the block diagram depicting the various steps necessary to establish end-to-end communication under a considered standard, as shown by figure 2), and the numerical specifications given by the standard bodies. A chosen architecture must yield a performance able to support end-to-end communication under the considered standards. We do

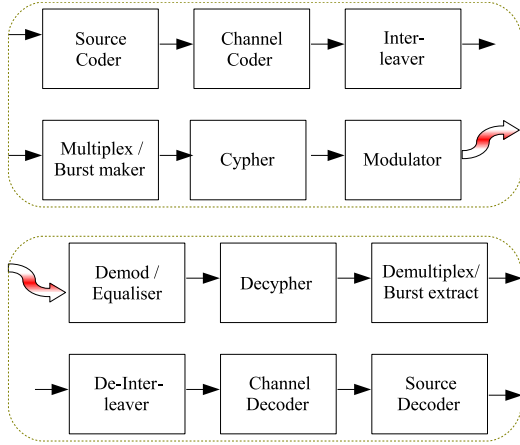


Fig. 2. The key to determining appropriate “deadlines” is an examination of the “transmission chain”. This corresponds to GSM, but our development does not specifically target that standard.

not further address here the determination of these deadlines, and assume that the pertinent information has been obtained elsewhere and given to us.

### C. Optimisation problem

Let:

$S_i$  with  $i = 1, \dots, I$  be the standards to be supported,

$\delta_i$  the “design deadline” of standard  $S_i$

$F_k$  with  $k = 1, \dots, K$  be 0-1 variables such that  $F_k = 1$  if component  $k$  is chosen to be installed in self-contained (dedicated) form.

$C_k$  and  $T_k$  be respectively the (monetary) cost and the execution time of module  $k$  (if self-contained)

$\tau_i(F_1, F_2, \dots, F_K)$  be the time of executing the tasks used to calculate the “deadline” of standard  $i$  for a particular choice of components

Notice that not all the possible choices are acceptable. For example, if there are three root nodes (standards), and the self-contained components that can each execute an entire standard (for the “Velcro” design) are labelled 1, 2, and 3, then, the point  $(1, 1, 1, 0, 0, \dots, 0, 0)$ , which means setting  $F_1 = F_2 = F_3 = 1$  and all others  $F_j = 0$ , is in principle acceptable (but not necessarily optimal!... this is the “Velcro” architecture). But some other choices would *not* support the desired standards (for example  $(0, 0, \dots, 0)$ , which means building nothing at all, is clearly unacceptable.

Let  $\Omega$  denote the set of all  $K$ -tuples of binary numbers that correspond to acceptable choices of the set of components (this is information taken directly from the hypergraph).

Then we seek to solve:

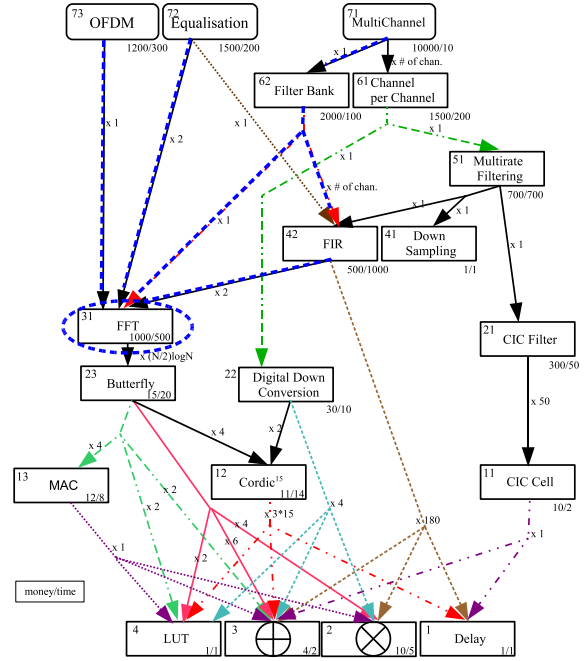


Fig. 3. Partial hypergraph with some design parameters, and a plausible solution

$$\begin{aligned}
 & \min_{(F_1, F_2, \dots, F_K) \in \Omega} \sum F_k C_k \\
 & \text{subject to} \\
 & \tau_1(F_1, F_2, \dots, F_K) \leq \delta_1 \quad (1) \\
 & \quad \vdots \\
 & \tau_I(F_1, F_2, \dots, F_K) \leq \delta_I
 \end{aligned}$$

### III. SOLVING THE OPTIMISATION

In principle the preceding problem can be solved simply by computing the cost of every feasible choice of components (that supports the desired standards and obeys the deadlines), and choosing the one of minimal cost. For larger problems, a “smarter” algorithm is needed. Below we provide an illustration which can be solved with minimal computing effort.

#### A. A realistic illustration

Figure 3 (representing an evolution of a figure from [2]) shows a sub-graph corresponding to the decomposition of several processing elements (equalisation, multi-channel, OFDM) that could be part of a multi-standard radio system. The sub-graph is *not* intended to show *all* the possible alternative implementations for each of the considered processing elements. Root nodes (standards) are *not* shown.

The equalisation block compensates for the multi-path impairment typical of wireless channels. It can be either implemented through a finite-impulse response filter (FIR) or through the fast Fourier transform (FFT) operator. The implementation of equalisation in the frequency domain is particularly attractive for channels

that exhibit long impulse responses, which leads to FIR filters with a very high number of taps. Notice that the inverse FFT is computationally equivalent to the FFT; thus, we attach a multiplicative factor of 2 to the arc pointing from the equaliser to the FFT.

Multi-channel refers to the channelisation function of a cellular base station. This can be accomplished via the “classical” channel per channel procedure, or by proceeding in parallel, through a filter-bank channeliser (which can be implemented via FFT). Other lower-level modules correspond to well-known signal processing constructs.

The graph shows that at least the FFT operator is needed to implement OFDM. The graph also shows that both equalisation and OFDM could employ the same FFT operator, although it is optional for FIR-based equalisation. A reconfigurable FFT component could provide the functionality of FFT operators of different orders.

The numerical values near the bottom right of a block represent cost / time associated with the component that could provide that functionality. The units of measurements are immaterial for our purposes. We only need relative figures, to be able to compare one design alternative to another. But the time figures must be consistent with those in which the deadlines are expressed. At the top left there is a numerical identifier for the corresponding module.

The arcs are tagged with a number of calls (NoC) figure. When a node is needed several times by a higher level module, it is called several times, and not physically replicated. Accordingly, the multiple calls affect the latency of the system, but not its monetary cost.

### B. Results and interpretations

The sub-graph of figure 3 is small enough to be solved by exhaustive search, although the number of design alternatives quickly explodes. The thick dotted lines in figure 3 show an FFT-only design, which is a conceivable outcome of the optimisation, for a specific set of deadlines.

Figure 4 provides a summary of our solution procedure, and shows some interesting design alternatives. For this illustration, we have ignored the designs that use the simplest (lowest level) components (adder, multiplier, etc). Thus, the simplest considered component is the CORDIC. A one in the column corresponding to a module means that, in that particular design, the module is implemented via a dedicated component.  $T_1$ ,  $T_2$  and  $T_3$  are the execution times corresponding to OFDM, equalisation and the channeliser for a given design. The deadlines should be applied to the entire transmission chain (not to individual tasks). Nevertheless, for the sub-design illustration under consideration, we pretend that the tasks at the top of the graph have associated deadlines (determined by the supported communication standards). We have

considered that an order-64 FFT operator satisfies our requirements, and that the multi-channeliser handles 25 channels.

The “Cost” column shows that the least expensive design consists of a CORDIC component only, but it performs slowly. This would be the chosen design if the deadlines are, for example, 6 200, 12 300 and 320 000 respectively (recall that time units have not been specified here). If the deadlines are tighter, the CORDIC-only design falls outside the feasibility region. Then other candidates may be chosen. For example, the butterfly-only design performs a good deal better in all three tasks, and is only marginally more expensive. The FFT-only design costs about 70 times more than the butterfly-only alternative, but performs 7-8 times better, and could very well be the optimal choice under tighter time constraints. The “Velcro” solution performs best and costs most, as it implements the top modules with dedicated components. The FFT-only design is far behind the Velcro approach only with respect to  $T_3$  (multi-channeliser). The performance gap between the FFT-only design and Velcro narrows considerably if one adds a filter bank. This combination provides performance near Velcro’s, for about one quarter of the cost.

## IV. DISCUSSION AND OUTLOOK

We have built a mathematical model that enables us to identify an architecture for a multi-standard reconfigurable radio, which minimises the cost of the radio while observing pertinent execution time constraints. The chosen architecture represents an optimal point between two extremes: (i) highly complex communication components each exclusively dedicated to a given standard (“Velcro” approach), and (ii) very simple components to be invoked by “higher layers” in support of the various standards. We represented the radio as a hypergraph of progressively simpler functional modules.

We have illustrated our approach through a simplified “sub-design”, which is sufficiently close to reality to provide useful insights. Our results agree with intuition. When the execution time constraints imposed by the supported standards on the considered high-level modules are very tight, the more complex dedicated components need to be chosen. On the other hand, if the time constraints are lenient, money is saved by supporting the standards through simpler (lower-level) components, such as the FFT, butterfly or even the CORDIC operator. Our analysis also sheds some light on some of the economic issues that may arise when a single architecture targets several communication standards. An optimal architecture that exclusively supports standards oriented to “slow” applications, such as voice, and text messaging, will favour lower-cost simpler, reusable components. Thus, a consumer primarily interested in such “traditional” applications will be better off by purchasing equipment that sup-

	O F D M	E Q U A L I Z E	M C H A N N E L I S E R	F I L T E R	C H - p - C H E R	M R A T E	F I L T E R	D N S A M P L E R	F F T	B U T T E R F L Y	D N C O N V E R S I O N	C I C F I L T E R	M A C	C O R D I C	C I C E L L U L A R	A D D E R E N C E	M U L T I P L E X E R	D E L A Y	N C H	N F T	F F T b f y			
M o d u l e	1 2 0 0 0	1 5 0 0 0	1 0 0 0 0	2 0 0 0 0	1 5 0 0 0	7 0 0 0 0	5 0 0 0 0	1 0 0 0 0	0 0 0 0 0	1 3 0 0 0	3 0 0 0 0	0 0 0 0 0	1 0 0 0 0	1 1 0 0 0	1 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0			
T i m e	3 0 0 0 0	2 0 0 0 0	1 0 0 0 0	2 0 0 0 0	7 0 0 0 0	0 0 0 0 0	0 0 0 0 0	5 0 0 0 0	2 0 0 0 0	1 0 0 0 0	5 0 0 0 0	8 0 0 0 0	1 4 0 0 0	2 1 0 0 0	2 2 0 0 0	5 5 0 0 0	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0			
	73	72	71	62	61	51	42	41	31	23	22	21	13	12	11	4	3	2	1	Cost	T1	T2	T3	
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	11	6144	12288	313344
	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	12	6144	12288	307925
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	15	3840	7680	195840
	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	511	6144	1000	31144
	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	515	3840	1000	28840
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1 000	500	1000	25500
	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3 000	500	1000	100
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12 700	300	200	10

Fig. 4. Some interesting design alternatives. A one in the column corresponding to a module means that it is implemented via a dedicated component. T1, T2 and T3 are the execution times corresponding to OFDM, equalisation and the multi-channel channeliser for a given design. The “Cost” column shows that the least expensive design consists of a CORDIC component, exclusively, but it performs slowly. The “Velcro” solution (bottom) performs fastest but costs most, as it implements the top modules with dedicated components.

ports only a few simple communication standards. In other words, certain care must be taken in choosing the set of standards to be supported by reconfigurable radios.

Many important issues remain to be addressed. For example, building the hypergraph of design choices is itself an object of research, as it is the adaptation of the graph to the evolution of communication standards. Likewise, finding an efficient algorithm to explore the solution space is a short-term objective of ours. Other important issues include consideration of (i) the time needed to re-configure the radio while switching from a standard to another, (ii) the “travel time” of signals from a component to another, and (iii) the possible contention among high level modules for the service of the same lower-level module (which may be particularly important if the radio needs to support operation over several standards at the same time). We hope to address these important issues in the near future.

## REFERENCES

- [1] J. Palicot, C. Roland, Y. Louet, and A. Al Ghouwayel, “A new parameterization technique for reconfigurable radio: the common operator approach.” Submitted to *Annales des Telecommunications*, 2006.
- [2] C. Moy, J. Palicot, V. Rodriguez, and D. Giri, “Optimal determination of common operators for multi-standards software-defined radio,” in *Proc. of 4th Karlsruhe Workshop on Software Radios*, (Karlsruhe, Germany), March 2006. Forthcoming.
- [3] J. Palicot and C. Roland, “FFT: a basic function for a reconfigurable receiver,” *Proc. of IEEE ITC*, vol. 1, pp. 898–902, Mar 2003.
- [4] B. Heyne and J. Goetze, “A pure CORDIC based FFT for reconfigurable digital signal processing,” in *12th European Signal Processing Conference (Eusipco2004)*, September 2004.
- [5] A. Al Ghouwayel, Y. Louët, and J. Palicot, “A reconfigurable butterfly architecture for Fourier and Fermat transforms,” in *Proc. of 4th Karlsruhe Workshop on Software Radios*, (Karlsruhe, Germany), March 2006. Forthcoming.
- [6] H. Wang, T. Cai, and J. Song, “Analysis of common structure and software download for software defined radio,” *Proc. of IEEE ICCT*, vol. 2, pp. 1112–1116, 2000.
- [7] F. Jondral, “Parameterization - a technique for SDR Implementation,” in *Software Defined Radio - Enabling Technologies* (W. Tuttlebee, ed.), pp. 233 – 256, London: John Wiley & Sons, 2002.
- [8] A.-R. Rhiemeier, “Benefits and limits of parameterized channel coding for software radio,” in *Proc. of 2nd Karlsruhe Workshop on Software Radios*, (Karlsruhe, Germany), Mar 2002.
- [9] U. S. Jha, “Object oriented HW functions accelerate communication, computing, and multimedia convergence,” *Proc. of IEEE PIMRC*, September 2005.
- [10] W. Tuttlebee, “Software-defined radio: facets of a developing technology,” *IEEE Personal Communications*, vol. 6, no. 2, pp. 38–44, 1999.
- [11] J. Belzile, S. Bernier, and M. Uhm, “Software radio modems enter the SoC era,” *COTS Journal*, September 2004.