Signal Path Scheduling for Reconfigurable SDR RF Hardware

Sami Kiminki
Aalto University
Computer Science and Engineering
P.O.Box 15400, FI-00076 Aalto
sami.kiminki@aalto.fi

Vesa Hirvisalo
Aalto University
Computer Science and Engineering
P.O.Box 15400, FI-00076 Aalto
vesa.hirvisalo@aalto.fi

Abstract

Software-defined radios (SDR) have received great attention recently. We believe that the SDR architecture will become an important design approach for multi-standard radio systems. We study the scheduling of analog resources in reconfigurable parallel SDR systems. We formulate the problem as the Signal Path Scheduling Problem*: given a graph of heterogeneous resources and a set of concurrent RF operations that are produced on-line, a set of time-dependent paths are to be found that can be used to execute the operations. We also describe a scheduler for the problem and demonstrate its feasibility.*

1 Introduction

Modern wireless devices are required to support multiple standards, such as LTE [5] and WLAN [2]. The trend is for more complex radio systems, as advanced techniques, such as MIMO and multiband aggregation, are being included in the standards, and especially, in the advent of cognitive radio standards [1]. One approach to prevent unmanageable growth of RF hardware is to use software-defined radios (SDR). We study an important subproblem of the SDR control, namely, scheduling of hardware resources for radio protocols.

Conventionally, dedicated chips are used for each protocol family. In this approach, the increased complexity requirements and the number of protocols to support translates directly to a greater number of hardware blocks. Because analog RF processing elements do not scale down like the digital elements, this creates design issues for the small portable devices such as smart phones. Yet, in such devices, all radio systems almost never operate simultaneously at peak capacity (bottlenecked by internal buses). This means suboptimal utilization of the hardware.

SDR architecture (*e.g.*, [3]) is an approach to reuse the hardware between protocols. The RF processing requirements for the major protocols are more or less the same, except for the exact frequency bands. Instead of wide range SDRs, we have studied coarse-grain reconfigurable RF platforms (RF-CGRA) [6], which use conventional RF elements in a switch matrix.

Dynamic reuse of hardware for multiple concurrently active radio protocols presents a control problem. This is arguably the main technical obstacle for commercial use. Radio operations, such as frame transmission, require µs-level precision in timing and ms-level response times for decisions. Further, radio protocols are not designed for resource sharing, and therefore, they do not cope well with resource contention. However, time-domain in-device interference avoidance mechanisms that are currently discussed in the standardization committees should provide the required flexibility. When dynamic resource sharing can be used, the advantages are significant [7].

In this paper, we study the problem of on-line resource assignment for analog RF processing. We formulate the related scheduling problem as the *Signal Path Scheduling Problem* (Sec. 2) and propose a solution, the *Fixed-Job Path Scheduler* (Sec. 3). The scheduler is applicable to resource assignment of wide range SDRs and RF-CGRAs. We run an experiment on the latter on three concurrent protocols (GSM, WLAN, DVB-H) using resource-constrained and non-constrained platforms (Sec. 4).

Scheduling reconfigurable analog RF processors differs in some important aspects from scheduling reconfigurable digital processors. First, the analog RF tasks are often rigid, *i.e.*, their timing cannot be changed. Second, analog signals cannot be buffered, which implies that all the processing elements in the RF signal path must be allocated simultaneously. Third, some analog elements, *e.g.*, amplifiers and synthesizers, can be simultaneously shared by multiple signal paths when the conditions are favorable. These differences have a profound effect to the scheduling problem and its solutions. To our knowledge, this scheduling problem has not been studied by the academia.

2 Signal Path Scheduling Problem

The Signal Path Scheduling Problem consists of a model of the RF processor and the requests for signal

We thank Aarno Pärssinen and Antti Immonen for the analog platform architecture proposal used in the experiment (Fig. 4).

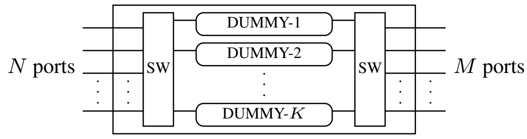

Figure 1. An N:M switch with at most K concurrent connections may be modeled by two full crossbar switches and K dummy resources.

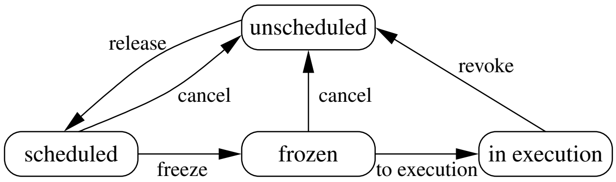

Figure 2. The state machine representing the life cycle of a job in the Fixed-Job Path Scheduler.

paths by the protocols. The task is to find the time-dependent processor configurations that satisfy the requests with minimum cost.

The processor is modeled as a set of resources, ports, and switches (see Fig. 5 for an example). The resources represent logical elements of the processor and the ports their inputs and outputs. The switches connect dynamically the ports of different resources. A subset of resources is called border resources, which are the starting or ending points of signal paths. Thus, every signal path contains at least one of these. In concrete designs, border resources usually provide external connectivity.

Switches have two sides, left and right. A port in the left side can be connected only to a port in the right side. Switches in the model are full crossbars, and thus, there are no restrictions on the number of concurrent connections. A single port in one side may also be connected to multiple ports in another. Restricted switch topologies can be modeled by adding dummy resources and switches. For example, an N:M switch with at most K connections can be modeled by an N:K switch, K dummy resources, and a K:M switch as illustrated in Fig. 1.

A signal path consists of a set of resources. A signal path is complete, when every port of a border resource is connected to a resource, and recursively, the ports of these resources are connected. Usually, there are conditions for signal paths, which are represented by resource-specific cost and shareability functions.

A resource-specific cost for a signal path request is numeric in range $[0, \infty]$. The cost is used in resource selection. Generally, more specialized resources (*e.g.*, less dynamic range, narrower bandwidth) have lower cost for better schedulability. More specialized resources are also often less power-hungry. Resources that cannot be used in the signal path have infinite cost. We consider only context-insensitive costs in this paper.

Resources may be concurrently shared by multiple signal paths. Shareability is determined by resource-specific Boolean-valued functions. For example, a synthesizer can usually be shared by signal paths using the same carrier frequency, whereas an amplifier cannot be shared when the signal frequencies overlap.

The requests for signal paths are often *rigid*, which means that they have fixed start and end times. The requests arrive on-line. When a signal path is granted, the requesting protocol may use it for its operations. The requests may also be denied, which is signalled back. Protocols may change behavior in response to denied requests.

3 Scheduler Design

The Fixed-Job Path Scheduler is an optimizing on-line scheduler for rigid jobs. The scheduler uses a multi-stage approach and either eager or lazy allocation strategy. The application interface of the scheduler contains functionality for requesting signal path allocations and callbacks for tracking updates in the allocations. The allocation requests are called *jobs*.

The scheduler contains two main components: job lifecycle management and signal path resolution. The lifecycle of a job follows the state machine of Fig. 2. The job is initially UNSCHEDULED. When the signal path is resolved for the job and the associated resources are allocated, the state becomes SCHEDULED. A period before the job is due to start, the job is FROZEN. This prevents priority overrides by other applications. This period is referred to as the *freeze period*. Frozen jobs may be modified by global optimizers, which can rearrange and combine signal paths for lower global cost. Finally, when the job execution begins, the signal path is activated and the job state is set to IN EXECUTION. After completion, the job is revoked from the system.

The path resolution algorithm is greedy. Border resources are the starting point of the search. The lowest-cost *suitable* resource is always tried first. A suitable resource has finite cost for the the request and is unallocated for the duration of the job or shareable with its existing allocations. Then, suitable resources are sought for ports, recursively with possible backtracking, until a signal path is found or all alternatives have been exhausted. If alternatives are not found, lower-priority jobs in SCHEDULED state may be rescheduled or canceled.

To speed up the path resolution, a precomputed decision tree and dynamic programming are used. The decision tree is also partitioned based on job classes, which speeds up resource selection. For example, resources that can be used only for 470–900-MHz RF operations need not be considered for other jobs. Dynamic programming stores the costs of resolved sub-paths, as it is likely that the same sub-paths are considered in case of backtracking.

For each resource, an allocation table is assigned. The

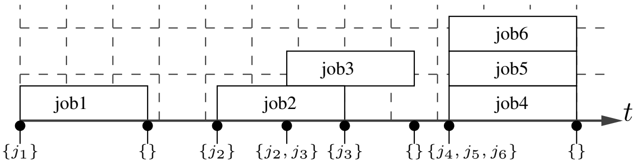

Figure 3. A resource allocation table is implemented using a map of points in time to a set of jobs, ordered by time.

allocation table is implemented as a map of points in time to a sets of jobs in SCHEDULED, FROZEN and IN EXECUTION states. A map entry represents an allocation to a set of jobs from the key (inclusive) to the following key (exclusive). See Fig. 3 for illustration.

The eager allocation strategy resolves signal paths and allocates the associated resources immediately at job arrival. The lazy strategy is the opposite: allocation is postponed to the latest possible point, *i.e.*, the freezing.

Frozen jobs present a global optimization window, which has the size of the freeze period. The window can be exploited by optimizer passes. We describe two: localization enhancer and job grouping pass. The localization enhancer looks into consecutive jobs of a protocol. If two consecutive jobs use different resources, the enhancer tries to rearrange the signal paths to use the same resources. The job grouping pass attempts to promote simultaneous resource sharing of different signal paths. The path resolution algorithm is greedy, and therefore, it may not always be able to find signal paths with shared resources.

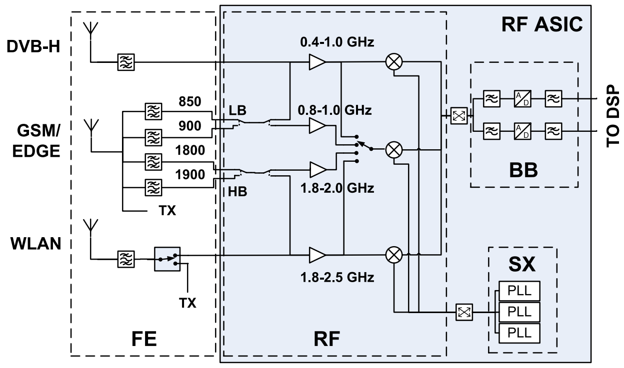

Figure 4. The receiver part of the RF-CGRA processor design for experimentation. An RF pipe is formed dynamically by activating switches.

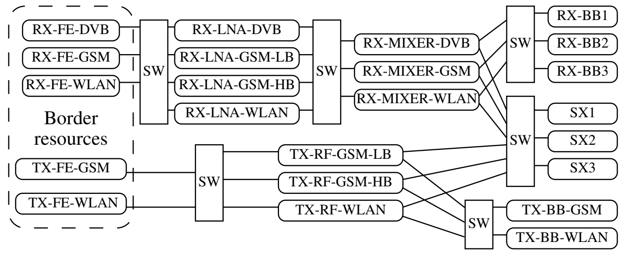

Figure 5. The resource graph for the experiment. The number of RX-BB and SX elements is varied.

4 Experiment

The experiment explores scheduling using a realistic RF-CGRA multiprocessor design and plausible protocol models. The experiment models a scenario of multiple active radio protocols on a mobile internet device.

The receiver (RX) design for the test platform is illustrated in Fig. 4. The front-end area (FE) consists of antennas and bandpass filters. The RF area consists of low-noise amplifiers (LNA) and mixers. Additionally, there are analog baseband units (BB) and synthesizers (SX). The transmitter (TX) incorporates a similar architecture but with the opposite signal direction.

The design comprises three parallel receiver pipes for DVB-H, four-band GSM and WLAN 802.11g. The RF-pipes are merged such that the synthesizer and the receiver baseband elements may be connected freely to any pipe. Additionally, the GSM receiver may share an LNA simultaneously with either the DVB-H or the WLAN receiver when the signal conditions are favorable. The operating ranges of the shared resources are well within reported capabilities [3]. Concurrent LNA sharing is discussed in, *e.g.*, [4].

Fig. 5 presents the model for the concrete RF processor design. The front-end blocks RX-FE-DVB, RX-FE-GSM, TX-FE-GSM, RX-FE-WLAN, and TX-FE-WLAN have been chosen as the border resources, as one of them must belong to any signal path.

For the receiver, cost functions are set up such that a specific protocol is required by the FE and MIXER elements. The LNA-GSM elements may be allocated only to GSM. LNA-DVB may be allocated by DVB or GSM and LNA-WLAN to WLAN or GSM, respectively. LNA elements also require transmission frequencies within limits (see Fig. 4). For the transmitter, cost functions are set up similarly. Amplifier and mixer elements of the transmitters are combined as logical RF elements and they may not be shared between protocols. Additionally, for each element except SXs, a signal direction constraint (*i.e.*, RX or TX) is applied.

The sharing rules are set up for LNA and SX resources as follows. LNA elements are always shareable (we assume favorable signal conditions). An SX resource is shareable between jobs if the jobs have a common carrier frequency.

We consider 5 cases in the experiment. Cases 1–4 differ in using either unconstrained or constrained platform and eager or lazy scheduling strategy. In case 5 we disable decision tree partitioning. Table 1 summarizes the cases. In the constrained platform, only two RX-BB and

Table 1. The measurement cases

Case	Platform	Scheduling strategy	Decision tree partitioning
1	unconstrained	eager	yes
2	constrained	eager	yes
3	unconstrained	lazy	yes
4	constrained	lazy	yes
5	unconstrained	eager	no

two SX elements are used, which inflicts resource contention when all three protocols attempt simultaneous operation. In the unconstrained platform, three RX-BB and three SX elements are used.

The workload for all cases is as follows, in priority order, highest first:

- GSM: 1800-MHz band, 1 TX, 1 RX, and 1 monitor slot per frame
- DVB-H: one 120-ms frame per second
- WLAN 802.11g: power-save mode, beacon per 102.4 ms, 26.0 ms of data traffic + 1.5 ms allocation overhead per beacon; 0.5 ms allocation granularity.

The processor allocation is per-RF-operation for GSM and DVB-H, as the transmission and reception timings for these are predictable. For WLAN, PS-Poll processing [2] is assumed. When a beacon frame is received, RX and TX resources are allocated until there is no more data buffered locally or at the access point. For each combination the run time was 1 second of simulated time.

Table 2 summarizes the results of the simulation runs. For the unconstrained platforms, eager (case 1) and lazy (case 3) allocation strategies perform equally well and are able to schedule all requests successfully. The main difference is with the maximum size of the allocation table, because the entries in the eager scheduler may live significantly longer.

In the constrained platforms, the eager allocation strategy (case 2) performs better than the lazy (case 4). This is because the eager scheduler is able to give immediate feedback on failed signal path allocations on most occasions for 802.11g. When the 802.11g protocol fails to allocate RX resources for data traffic, it does not attempt to allocate TX resources, and thus, a number of job requests and cancels are avoided. With the lazy scheduler, this feedback comes later at job freeze, and allocation jobs for both RX and TX resources are generated.

Decision tree partitioning reduces more than half of the resource evaluations in this experiment, as seen by comparing cost function evaluation counts in cases 1 and 5.

5 Conclusions

In this paper, we considered the scheduling of SDR RF resources for multiple concurrent protocols. We formulated the underlying scheduling problem and proposed a concrete solution. We also demonstrated the feasibility of the solution by an experiment with realistic reconfigurable hardware design and protocol models.

We believe that software-configurable RF hardware in one form or another is the future for multi-standard radio platforms. This is underlined by the emerging cognitive radio standards, which are often the secondary users of the spectrum, *e.g.*, 802.22 in the TV-bands. For the secondary users, it is only logical that, in addition to the spectrum, the hardware resources are also shared to avoid needless redundancy.

Table 2. Simulation results

	Case 1	Case 2	Case 3	Case 4	Case 5
Job requests	1831	1836	1831	1949	1831
Executed jobs	1759	1650	1759	1650	1759
Rescheduled jobs	0	2	0	1	0
Canceled jobs	0	114	0	227	0
Path alloc attempts	1831	1839	1761	1983	1831
Failed alloc attempts	0	114	0	227	0
Cost evals/path alloc	8.02	6.36	7.98	6.18	16.57
Alloc table lookups	14681	11697	14053	12263	14681
Max alloc table size	62	62	7	8	62
GSM RX jobs total	420	420	420	420	420
GSM RX failed	0	0	0	0	0
GSM TX jobs total	209	209	209	209	209
GSM TX failed	0	0	0	0	0
DVB-H RX jobs total	60	60	60	60	60
DVB-H RX failed	0	0	0	3	0
802.11g RX jobs total	540	599	540	599	540
802.11g RX failed	0	114	0	114	0
802.11g TX jobs total	530	476	530	589	530
802.11g TX failed	0	0	0	113	0

References

[1] L. Berlemann and S. Mangold. *Cognitive Radio and Dynamic Spectrum Access*. John Wiley & Sons, 2009.

[2] M. S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2nd edition, 2005.

[3] V. Giannini *et al.* A 2mm^2 0.1-to-5GHz SDR receiver in 45nm digital CMOS. *ISSCC Digest of Technical Papers*, pages 408–409, Feb. 2009.

[4] H. Hashemi and A. Hajimiri. Concurrent multiband low-noise amplifiers – theory, design, and applications. *IEEE Trans. Microw. Theory Tech.*, 50(1):288–301, Jan. 2002.

[5] H. Holma and A. Toskala, editors. *LTE for UMTS — OFDMA and SC-FDMA Based Radio Access*. John Wiley & Sons, 2009.

[6] A. Immonen, A. Pärssinen, T. Zetterman, M. Talonen, J. Ryynänen, S. Kiminki, and V. Hirvisalo. A reconfigurable multi-standard radio platform. In *1st International Workshop on Energy Efficient and Reconfigurable Transceivers*, 2010.

[7] S. Kiminki, V. Saari, A. Pärssinen, V. Hirvisalo, A. Immonen, J. Ryynänen, and T. Zetterman. Design and performance trade-offs in parallelized RF SDR architecture. In *Proc. 6th Intl. ICST Conf. on Cognitive Radio Oriented Wireless Networks*, 2011.