Identifying Optimal Generic Processors for Biomedical Implants

Christos Strydis and Dhara Dave

Computer Engineering Laboratory, Electrical Engineering Dept.
Delft University of Technology
Postbus 5031, 2600 GA, Delft
The Netherlands
C.Strydis@tudelft.nl, D.Dave@student.tudelft.nl

Abstract— **The extremely limited resource budget available to medical implants makes it imperative that they are designed in the most optimal way possible. The limited resources include - but are not limited to - battery life, expected responsiveness of the system and chip area. We have already detailed the design of a design-space exploration (DSE) tool specifically geared towards finding the Pareto-optimal design front. In this paper, we choose processor configurations from the Pareto-optimal processor set found by the DSE using real implants as case studies. We find that even under the extremely biased constraints that we use, our processor(s) perform better than many of the real implants. This provides strong hints towards designing an implant processor that is generic enough to cover most, if not all, implant applications.**

I. INTRODUCTION

The market for biomedical-implants is slowly but surely expanding with a rising number of applications. While, at first restricted to the field of pacemakers [2], implants have now diversified and cover nearly all bodily systems, from musculoskeletal, to circulatory, to neural [3]. Moreover, recent trends in global healthcare [4] are pushing towards "smarter" implants with increased capabilities. An extended study performed on more than 60 different implantable systems backs this claim [1]. For the 12-year study period 1994 – 2005, Fig. 1a reveals an increasing number of implants charged with non-trivial processing duties and featuring in-system memory blocks. Every year, about 12% more implants perform some complex processing task(s) *in vivo* while 17% more implants are designed with sizeable memories on them.

However, such provisioning comes at a cost. As Fig. 1b reveals, even though operational voltages are dropping in agreement with shrinking process-technology trends, implant power consumption exhibits an aggregate increase of 15%[1].

A third, related trend has also been observed: It has been common practice so far to custom-design the hardware for each implant application, often completely from scratch (see Fig. 1c). Although this was easier for the simpler devices, designing a processing-capable core for every implant application is not practical due to large development and deployment costs. Therefore, the use of commercial, off-the-shelf (COTS) components is also gradually increasing (seen in the same figure). However, such designs are ad-hoc,

and are not consistently designed with the restricted resource limitations of implants – as exemplified for power in Fig. 1b. They also involve long design and testing times and therefore have higher costs.

Therefore, it is becoming apparent that "smart", pre-designed and pretested components are needed, which are specifically geared towards medical implants. Such components must cover a *large application range* in order to be economical as well as reliable and safe. This is the express goal of the SiMS project [5]. Our final goal is to design a (so-called SiMS) generic, low-power implant processor or processor family, for covering a large part of the implant domain.

In this paper, we present the results of an automated, design-space-exploration (DSE) effort performed to identify such SiMS processor candidates. We also select a number of representative, real implant applications in the literature and explore the possibility of covering them with a few of the identified processors. Concisely, the contributions of this work are:

- To propose a new, realistic, worst-case workload mix for future implant processors;
- Along with the previously generated DSE toolset and the new workload mix, to provide a complete framework enabling the implant designer to make informed decisions about resource allocation for future implant design;
- To propose Pareto-optimal, alternative microarchitectural configurations for the SiMS processor;
- To make a proof-of-concept, first attempt at fitting a single (or a few) of the identified configurations to real implant applications.

It should be noted that this study focuses on the microarchitectural aspects of the SiMS processor, thus no Instruction-Set-Architecture (ISA) analysis is present.

The paper is organized as follows: section II gives an overview of related works in the field. Section III gives an overview of the experimental tools, and the synthesis of the implant workload, used for this work. In section IV a concise presentation of the implant study cases is given. Section V presents in detail the findings of this work. Finally, overall conclusions and future work are listed in section VI.

---

[1]For the observed dip in the middle years 1998 – 2001 a biasing artifact is responsible in the sampled data; see [1] for a detailed explanation.

(a) More implants require processing capabilities and memories.

(b) Implants exhibit decreasing voltage but increasing power needs.

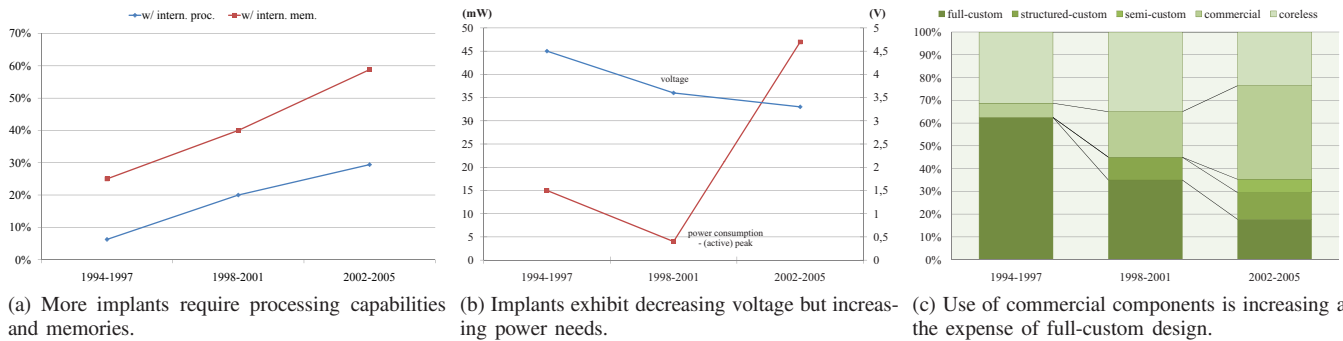(c) Use of commercial components is increasing at the expense of full-custom design.

Fig. 1. Implant trends over the survey period 1994 – 2005 [1].

II. RELATED WORK

In the past, a few attempts have been made to design implants with a certain degree of modularity in order to make them capable of adapting to different application scenarios.

Fernald et al. [6], [7] propose a modular microprocessor architecture which accepts various peripheral modules such as sensors, actuators and transceivers. Application flexibility is underpinned by a dual ring-bus interconnect linking an arbitrary number of modules to the processing core which is a fully featured 16-bit μP (PERC), based on Hector [8]. Command and data packets, traveling across each bus, have predefined, consistent structures and plugged modules are built to interface to them.

Contrary to the additive nature of the above design, Smith et al. [9], [10] have addressed the problem of flexibility from a subtractive angle. An implantable stimulator device with provisions for a large set of peripherals was designed. Given a specific application, unutilized components of the initial, baseline design can be removed, resulting in a reduced system, tailored to the application needs and with lower power/area requirements than those of the base design.

Valdastri et al. [11] present a versatile implantable platform that provides multi-channel telemetry of measured biosignals. Its versatility resides in its ability to support different types of sensors and to allow for easy reprogramming so as to fulfill different application requirements. To demonstrate the correctness of the concept, a specific case study is implemented for gastric-pressure monitoring which is a PCB-mounted assembly, supporting up to 3 sensor channels. This implant can transmit digitally modulated data to an external receiver over a wireless link with robust error control.

Furthermore, Salmons et al. [12] perform a design and comparative study between an ASIC-based and a microcontroller-based microstimulator device for restoring functionality to paralyzed muscles. Analysis has shown that, if carefully designed with low-power modes and checked for software bugs, the latter version is beneficial to the ASIC with respect to development and testing costs.

The work presented here is original in that it attempts to develop a truly generic and low-power processor architecture while at the same time providing the performance needed by current and future applications in the field. A *systematic*, *structured* approach to the problem, supported by the recent, rapid advances in microelectronics technology [13], finally make such a venture realistic.

III. EXPERIMENTAL SETUP

Our work so far has been focused on investigating the design space of implant processors in order to propose one (or a few) processor architectural configurations able to support a diverse range of implant applications. This task is difficult to tackle as we have repeatedly encountered the following problems:

- Implant applications (and their requirements) are very diverse, mirroring the wide range of potential pathoses in the human body. To make matters worse, biomedical implants are a relatively new field, traditionally dominated by a handful of companies which are extremely protective of their product designs. With literature being limited, consensus in the "application domain" cannot be easily established;

- No systematic approach exists for designing processors specifically tailored towards implant applications. Further, there are no established operational parameters either. Thus, a number of educated assumptions are necessary for introducing boundaries to the design problem;

- Verified tools for modeling the desired processors and exploring the design space are not readily available. The ones used are best-effort ones which introduce accuracy errors and deviations between simulated and actual results. These deviations are not linear and, thus, cannot be easily predicted in advance.

Except, perhaps, for the first item in this list, the above problems are well-known and have already been encountered in other application fields. If we are to attempt a first take on a (few) processor(s) capable of serving a number of implant applications, we need to fill the missing information with some further estimations. However, we will have to ensure that these estimations are drawn such that the resulting implant-processor architectures are guaranteed to cover the targeted applications under *worst-case conditions*. In effect, we will intentionally overprovision our processor(s) by a certain margin. For this study to become possible, a number of components are required. In the following subsections, we briefly introduce these components along with their capabilities and limitations.

TABLE I

ARCHITECTURAL DETAILS OF (MODIFIED) XTREM.

Feature	Value	Feature	Value
ISA	32-bit ARMv5TE-compatible	Ret. Address Stack	VAR[*]size
Pipeline depth / width	7/8-stage, super-pipelined / 32-bit	I/D TLB (separ.)	VAR size / VAR size
RF size	16 registers	Write Buf. / Fill Buf.	VAR size / VAR size
Issue policy / Instr. Window	in-order, single-instruction	Mem. bus width	1B (1 mem. port)
I/D Cache L1 (separ.)	VAR size/assoc. (1-cc hit / 170-cc miss lat.)	INT/FP ALUs	1/1
BTB	VAR size, fully-assoc. / direct-mapped	Clock frequency	2 MHz
Branch Predictor	VAR (4-cc mispred. lat.)	Implem. Technology	0.18 μm @ 1.5 Volt

[*] Values denoted with 'VAR' indicate adjustable parameters by the GA. For complete parameter ranges refer to [14]

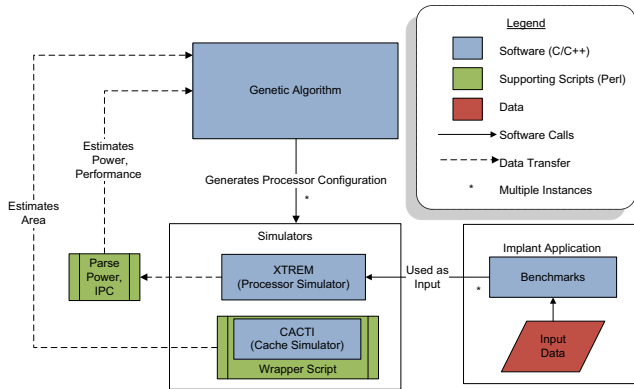

Fig. 2. Overview of ImpEDE exploration framework.

A. Exploration framework

In order to perform automated exploration, we have employed ImpEDE – a previously proposed multiobjective, DSE framework for investigating Pareto-optimal, implant-processor alternatives [14]. An overview of the framework is shown in Fig. 2. Optimization (minimization) objectives within the framework are: *maximum execution time* (in sec), *total area utilization* (in mm^2) and *total average power consumption* (in mW). At the core of the framework is a genetic algorithm (GA) that traverses the design space by encoding each processor configuration as a chromosome. These chromosomes are *evolved* using a process mimicking natural evolution, in order to yield a Pareto-optimal set of processor configurations in terms of the three optimization objectives. For detailed working of the genetic algorithm, please refer to [14]. Trading off execution times and quality of results, all full runs of the GA were allowed to evolve for 200 *generations* with a population size of 20 *chromosomes* per generation.

A genetic algorithm needs comparison metrics in terms of the objectives in order to evolve the objectives in question. Within the framework, performance (i.e. execution time) and power metrics are provided by utilizing XTREM, a cycle-accurate, performance and power XScale-processor simulator [15]. XTREM allows monitoring of 14 different subsystems, the most pertinent to our study being: Branch-Target Buffer (BTB), Instruction Cache (I$), Data Cache (D$), Internal Memory Bus (MEM) and Memory Manager (MM). While we have kept some XTREM parameters fixed in order to model implant processors more accurately, we have purposefully left some others variable for the GA to

explore their optimal settings, as summarized in Table I. More advanced microarchitectural structures such as caches and branch predictors have not been disabled in XTREM as they have been shown [16], [17] to be relevant within the biomedical-implant context.

While XTREM has been very useful in our studies so far, it is not an ideal simulator. One of the major drawbacks of using XTREM is that it models a low-power, high-performance embedded processor – an overkill in the implant application domain. Another shortcoming is that it does not simulate any (off-chip) memory, thus making system-level simulations difficult. Also, our long usage of XTREM has revealed a number of bugs and modeling inaccuracies (see [18] for an extensive list), most of which have been solved by a newer simulator XEEMU [19]. Therefore, an XEEMU porting for our framework is in the process of being developed. In the meantime, XTREM has been maintained in our exploration chiefly for reasons of compatibility with previous work, availability and ease of use. We have combined readouts from XEEMU regarding memory power consumption and have updated the power metric in our exploration in order to overcome some of the above stated limitations of XTREM.

For quantifying each chromosome's area cost, we have used CACTI v3.2, a well-known, cache-area estimation tool. The total area cost has been calculated as the sum of the (fixed) net processor and (off-chip) memory area, based on related literature; and the per-case cache (BTB, I$, D$ etc.) estimates derived from CACTI simulations.

B. Biomedical workload

The above framework is supplemented by biomedical workloads that are input to the simulators in order to drive the exploration process. We focus on benchmarks representative of the actual workloads that will be fed in real implants, in terms of *functional* as well as of *timing* behavior. To represent these workloads, we make use of the benchmarks found in ImpBench v1.1 [20], comprising compression, encryption, data-integrity, synthetic and stress benchmarks (see Table II).

Most implant applications are iterative process wherein sensors are read, actuators are enabled, and processing tasks are triggered; all within a fixed, periodical, time frame. This is also indicated by the benchmarks. A generic, typical workload mix for future implants has already been presented in [4]. Concisely, a synthetic application (DMU-variant) executes (manipulating sensors and actuators) and periodically

TABLE III

ImpEDE-EVOLVED PROCESSOR CONFIGURATIONS.

conf	BPRED	BTB sets	assoc	RAS	L1-I$ sets	bl.size	assoc	repl	L1-D$ sets	bl.size	assoc	repl	Mem lat.	Ex. Time	Power	Area
	(-)	(#)	(#)	(#)	(#)	(bits)	(#)	(-)	(#)	(bits)	(#)	(-)	(#cc)	(sec)	(mW)	(mm²)
1	bimod	64	8	8	4096	16	32	FIFO	4096	16	1	FIFO	2	27.465	17.539	2521.36
2	bimod	128	8	2	256	16	16	LRU	4096	8	2	LRU	16	37.166	15.368	394.53
3	bimod	64	32	0	256	16	16	RAND	1024	32	2	FIFO	1	1.790	123.143	400.92
5	taken			1	1024	32	32	RAND	16	16	16	FIFO	8	26.143	13.842	1325.39
6	nottaken			8	1024	16	4	FIFO	512	32	2	FIFO	1	1.433	63.217	327.10
7	bimod	128	8	4	2048	16	8	FIFO	4096	32	1	LRU	1	1.751	93.200	659.94
8	taken			0	16	32	8	RAND	512	32	4	RAND	8	2.777	74.860	299.37
9	bimod	128	2	4	64	32	8	LRU	128	32	16	FIFO	8	2.181	63.288	327.61
10	bimod	32	16	8	128	8	2	FIFO	16	32	8	RAND	1	4.516	87.887	243.68
11	bimod	64	8	1	256	16	4	FIFO	64	32	16	FIFO	2	1.951	93.366	298.79
12	nottaken			4	16	8	2	FIFO	64	8	2	RAND	8	35.571	88.153	215.30
13	bimod	64	4	2	8	16	1	FIFO	128	32	2	RAND	16	6.834	69.729	227.71
14	nottaken			2	64	16	2	LRU	16	32	16	FIFO	16	4.605	67.197	250.99
15	nottaken			2	16	8	4	FIFO	32	32	2	FIFO	4	6.823	80.947	218.21
16	nottaken			2	8	32	2	LRU	64	16	2	FIFO	1	24.463	71.681	218.84
17	nottaken			1	32	16	16	FIFO	16	32	4	LRU	8	2.868	69.781	238.62
18	bimod	128	2	8	64	8	16	FIFO	16	32	16	FIFO	2	2.222	90.816	268.36
19	bimod	32	1	8	64	16	16	LRU	128	32	32	FIFO	4	1.922	74.419	421.30
20	nottaken			1	128	16	4	LRU	64	32	4	RAND	16	3.395	62.336	236.57

TABLE II

IMPBENCH V1.1 BENCHMARKS. (*) INDICATES TYPICAL VALUES FOR
$10-KB$ WORKLOADS, EXCEPT FOR DMU-VARIANTS WHICH USE
THEIR OWN SPECIAL WORKLOADS.

| benchmark	name	size (KB)	dyn. instr.* (average) (#)	dyn. μops* (average) (#)
Compression	miniLZO	16.30	233186	323633
	Finnish	10.40	908380	2208197
Encryption	MISTY1	18.80	1267162	2086681
	RC6	11.40	863348	1272845
Data integrity	checksum	9.40	62560	86211
	CRC32	9.30	418598	918872
Real applications	motion	9.44	3038032	4753084
	DMU4	19.50	36808080	43186673
	DMU3	19.59	75344906	107301464
Stressmarks	stressmotion	9.40	288745	455855
	stressDMU3	19.52	124212	224791
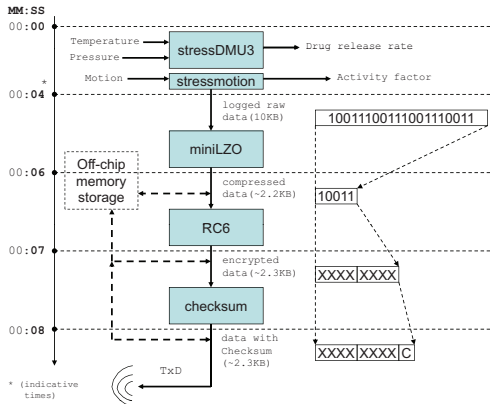
Fig. 3. Conceptual block diagram of simulated implant application (based on [4]).

(when 10 KB of logged data are collected) compression, encryption and data-integrity tasks are invoked on the data.

In this case, and in order to provide a realistic, *worst-case*, SiMS-processor design, we update the above workload mix as follows: Per benchmark category, we select the fastest executing algorithm - i.e. *miniLZO* for compression, *RC6* for

encryption and *checksum* for data integrity. As for the synthetic benchmark, we replace it by *both* stressmarks *stressmotion* and *stressDMU3* which simulate a single-iteration, worst-case instance of the regular benchmarks *motion* and *DMU3*, respectively. This combination is depicted in Fig. 3.

Every processor configuration (or chromosome) evolved through ImpEDE is made to execute this whole sequence of benchmarks, representing *the busiest (i.e. worst-case) iteration* in the implant's operational lifetime. The *execution-time* metric is calculated as the accumulation of execution times of all involved benchmarks while the *power-consumption* metric is calculated as the weighted average of the power consumptions of all involved benchmarks with each one's execution time used as the weighting coefficient.

To push the worst-case, processor-design envelope further, and without loss of generality, we use $10-KB$ EMGII as the input dataset to the above benchmarks. It features a realistic size and has been shown to evoke the longest execution times among the available physiological datasets [14].

It should be noted, last, that all ImpBench benchmarks (and, thus, the ones currently used) are kernels simulating the processing load of an implant processor. Therefore, they suffer from certain modeling limitations: they have no way (a) of modeling the behavior of any implant peripherals (biosensors/bioactuators), and subsequently (b) of accurate modeling any externally triggered (timing or other) events, i.e. they have no sense of real time. This is a well-known problem in benchmarking (event-driven) embedded systems. This has been addressed by introducing extra code in the benchmarks to imitate the passage of time and the occurrence of external events (e.g. timer/sensor interrupt). This, of course, has to be done in a careful fashion as it can potentially pollute simulation results in terms of timing behavior, executed instruction mix and so on.

With the above considerations, ImpEDE has been allowed to run over significant periods of time in search of optimal SiMS-processor configurations. Table III lists the results of

TABLE IV

STUDY CASES OF REAL IMPLANTABLE APPLICATIONS (TAKEN FROM [1]).

case	Author	Pub. Year	Application	Power source (-)	Sensor count (#)	Sampl. rate (Hz)	ADC resol. (bits)	Core arch. (-)	Core freq. (MHz)	Ex.Time Worst-case (sec)	Power Peak (mW)	c/s Area Total (mm^2)
A	Smith et al. [9], [10], [21]	1998	restoration of paralyzed muscle, MES	RF-ind.	2	100	12	FSM	1	34.1333	96.00	937.50
B	Eggers et al. [22], [23], [24]	2000	ICP-based diagnosis for brain diseases	RF-ind.	1	100	10	no	0.125	81.9200	0.24	58.50
C	Rollins et al. [25]	2000	continuous ECG for spontaneous cardiac arrhythmias	battery (ext.)	8	1000	12	FSM	2	0.8533	34.00	4209.67
D	Valdastri et al. [11]	2004	gastric-pressure monitoring	battery	1	25000	10	8-bit μC	4	0.3277	50.40	162.00
E	Au-Yeung et al. [26]	2004	continuous AEG, delivery of atrial ATP	battery	4	333	10	8-bit μC	8	6.1502	115.30	5106.00
F	Liang et al. [27]	2005	ENG	RF-ind.	1	11000	10	8-bit μC	n/a	0.7447	90.00	1350.00

this search. Each one of the 19 entries is a Pareto-optimal, non-dominated solution to the problem. Performance, power and area metrics are also reported for each entry.

IV. IMPLANT STUDY CASES

For selecting representative study cases of the implant application domain, we draw upon the extensive survey performed by Strydis et al. [1] who has investigated more than 60 cases of experimental as well as commercialized implantable devices. The selected applications will help provide diverse operational requirements for our targeted SiMS processor(s).

In order for a direct and fair comparison with the candidate SiMS processor(s), we have to place the study cases in the same design space as the one traversed by ImpEDE. That is, we need to know the worst-case execution time, the power consumption and the area cost of each of the studied implantable systems. This requirement limits the number of eligible systems to only 6, as shown in Table IV. In spite of this, the scope of applications addressed is diverse – spanning the muscular, neural, cardiac, gastric, atrial, and nervous systems. An extensive description of the various devices can be found in [1], yet short descriptions are given below for convenience.

Device #A, by Smith et al. [9], [10], [21], is used for functional neuromuscular stimulation (FNS). The authors are describing a flexible implantable-stimulator and telemetry (IST) system which makes provisions for multiple channels of stimulation, multiple channels of sensor or biopotential-electrode sensing and power and bidirectional data communication between the implant and an external control unit (ECU) over a transcutaneous, inductive RF link.

Device #B, by Eggers et al. [22], [23], [24], is a miniature, implantable, intra-cranial pressure (ICP) measurement system for monitoring patients in the ER (e.g. post-surgery patients). This essentially is a telemetry-powered, implantable system consisting of an absolute-pressure sensor and two low-power ASICs for pressure read-out and telemetric data/power transmission.

Rollins et al. [25] have developed an implantable radio-telemetry system (**device #C**) for continuous monitoring of

ECG signals over a period of weeks to months for capturing all events preceding sudden-death incidents. The design of the system centers around two separate but inter-dependent units: the implantable unit and a backpack which holds batteries for powering the implant, a processor and a WLAN-card for forwarding the data wirelessly to a base station for further archiving and analysis.

Valdastri et al. [11] present a new, versatile implantable system (**device #D**) that provides multichannel telemetry of measured biosignals. The presented system consists of the microcontroller-based implant which can monitor and wirelessly transmit up to 3 channels to an external receiver and, in this case, monitors gastric pressure in the stomach.

Au-Yeung et al. [26] have built an implantable **device (#E)** which is capable of continuously monitoring the electrophysiological state of the heart atria and, also, of delivering chronic and programmable atrial pacing. In effect, the proposed system can induce standard AF, can measure the atrial effective refractory period (AERP), can deliver anti-tachycardia pacing (ATP) therapy and can sense and telemeter atrial electrograms (AEGs).

The developed system by Liang et al. [27] (**device #F**) allows recording and telemetry of electroneurogram (ENG) signals to an external host computer. The implant is built to receive power and ASK-modulated commands over a wireless RF-link and to transmit physiological data back through passive telemetry. The device consists of a μC with on-chip, $10 - bit$ ADC which digitizes and forwards data acquired from an analog-sensing front-end through cuff electrodes.

As illustrated in Table IV, actual implant chipset sizes have been employed for the area metric. The term 'chipset' represents the dimensions of any design and assembly type; ranging from fully integrated and multi-chip module (MCM), to PCB-mounted. Figures were also available for the implant chip-only size (in mm^2) - e.g. processor die but no supporting PCB - and for the implant package size (in mm^3). However, the chipset area was finally preferred so as to allow more direct and fair comparisons with the XTREM processor plus off-chip memory. Memory is specifically included in this

work as the initial analysis (see section I) revealed rising trends in memory usage for future implants.

As far as power consumption is concerned, the most frequently reported figure in the actual implantable systems is active (peak) power which was the power measured during full load. This is the power simulated by XTREM as well, since XTREM does not support any low-power or sleep modes of operation. Therefore, peak power values as reported by XTREM, without any conversion, have been used as the power metric.

Finally, for the performance metric, some estimations were required in order to make the real and simulated systems commensurable. All case studies are devices with periodic monitoring windows, thus exhibiting a specific sampling rate, as shown in Table IV. The inverse of this rate (or frequency) signifies the maximal amount of time the device has to read a sensor value and process it before the next value arrives. In effect, this is the worst-case execution time of the implant. Note that we might have used the 'Core frequency' as a measure of the processing rate but this would be accurate only for designs with very simplistic cores as in cases #A, #B and #C. For the rest of the cases whereby a full μC is used, the core frequency is much higher (typically three orders of magnitude) than the actual sampling frequency and, thus, does not reflect the real-time deadlines of the implant.

However, the performance metric for the study cases (as the inverse of the sampling rate) is not yet completely normalized with respect to that of our processor configurations. As discussed in the previous section, our processor configurations consume EMG input data of $10\ KB$. The study cases, on the other hand, are assigned (by design) the task of consuming a single sample of size equal to the ADC resolution used (e.g. $8\ bits$), from each sensor they have on-board. Therefore, for each study case to collect $10\ KB$ of sample data, a longer execution time is needed which is inversely proportional to the number of available sensors. The normalized, worst-case execution time is then given by:

$$ET_{norm} = \frac{10\ KByte}{F \times N \times S}, \tag{1}$$

where F is the sampling frequency (in Hz) of the sensor(s), N is the ADC resolution (in bits) and S is the number of working sensors on the implant. This is the execution time given in Table IV. It should be noted that formula (1) does not account for any further processing of the data once acquired. On the contrary, our evolved processor configurations perform significant processing tasks, as discussed in section III-B. Therefore, by considering an ideal situation involving zero processing time for the real life comparison cases, while keeping a non-zero processing time for our designs, we design our implant processors for the worst-case.

V. EXPLORATION RESULTS

A. Analysis

In this section, we see how a single processor or family of processors can be identified as "generic processor(s) for implants". We denote this set of processors as $\mathbb{P}$. Once developed, this family should be able to replace a large number of implant applications - i.e., there must be a processor p in the set $\mathbb{P}$ that has equivalent or better design characteristics than the existing application in question. For reasons of economy, $\mathbb{P}$ must be a minimal set.

As mentioned before, we consider 3 design characteristics – power, performance and area. Therefore, we have a 3D design space, as shown in Fig. 4a. In this Figure, our processor design points (denoted with numbers) and the case-study points (denoted with letters) have been plotted. For clarity purposes, 2D Figs. 4b, 4c and 4d of the same 3D space have also been plotted. The bounding boxes around the study cases represent 10% confidence intervals to compensate for the uncertainty introduced when trying to fit the study cases in the design space.

From the figures, we notice that implant devices #A and #E are dominated by most of the candidate processor points in all three dimensions. Therefore we can include any configuration from {*6, 7, 8, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20*}[2] in order to cover applications #A (restoration of paralyzed muscle) and #E (atrial electrogram and anti-tachycardia pacing).

The other devices are not so easily fitted without applying standard engineering practices. For example, device #B, is largely dominated in terms of execution time and area but not in terms of power. In fact, #B has the lowest power profile ($0.24\ mW$) by a wide margin across processor configurations and study cases alike. Looking at the application details [22], we see that, #B is a minimal functionality device measuring intra-cranial pressure and is powered by an *external* power source (through RF induction). Therefore, power is not a big issue for this application as it allows for a non-implanted power source – which in practice can be replaced by a bigger power source if required without compromising the implanted chip area. Therefore, any of the processors that dominate #B across the other two dimensions may still be used to replace it – with the provision of a bigger external power source, and keeping in mind heat dissipation constraints. Therefore, #B (diagnosis of brain disease) can be replaced by {*10, 12, 13, 14, 15, 16, 17, 18, 20*}.

Furthermore, we see that devices #C, #D and #F are highly performance oriented. Out of these, device #D performs gastric-pressure monitoring at the extremely high sampling rate of $25\ kHz$, the highest rate among all applications in the study. In practice, however, gastric pressure varies at a much lower rate (making 1-5 second samples more than sufficient), making this a good example of implant over-design. We see that configuration {*2*} dominates case #D if this fact is taken into consideration, and can be a suitable replacement in practice. On the other hand, devices #C and #F perform continuous-ECG and ENG monitoring, which are indeed demanding applications in terms of throughput and, therefore, cannot be accommodated by a lower sampling rate. We observe that these devices are dominated by {*1, 2, 5*}
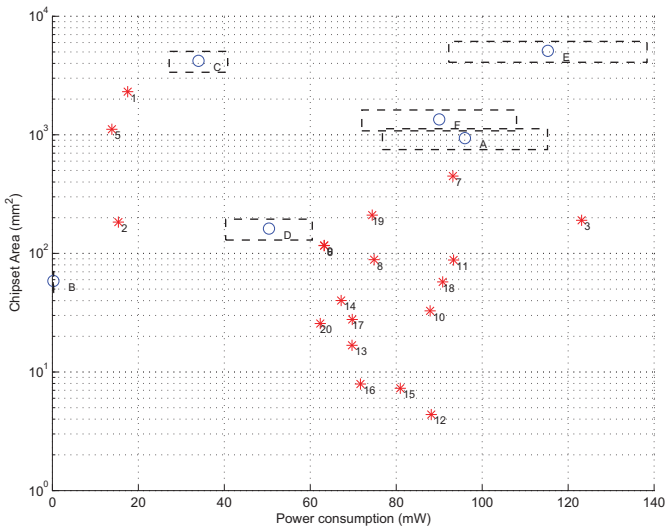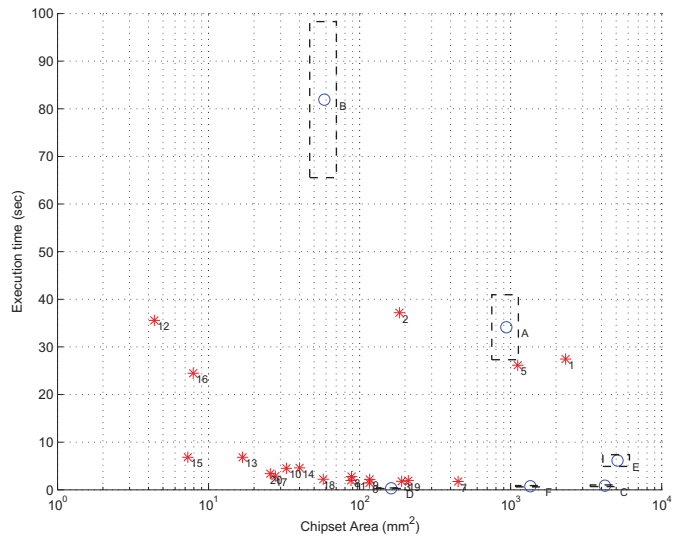
[2]As labeled in Figure 4.
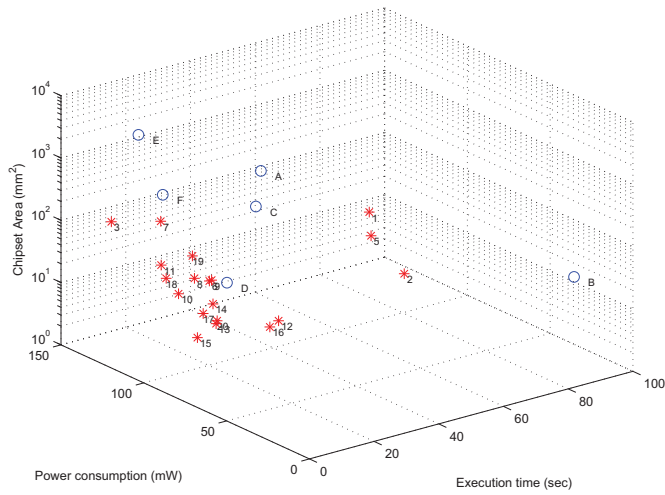
(a) 3D view
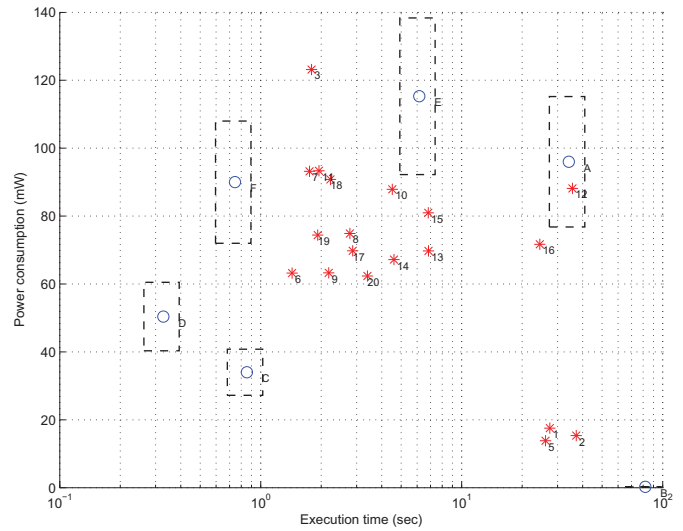
(b) Worst-case, single-loop execution time - Average power consumption

(c) Average power consumption - Total chipset area

(d) Total chipset area - Worst-case, single-loop execution time

Fig. 4. Comparison of study cases and DSE results for 10 KB workloads running on the selected benchmarks.

and $\{2, 5\}$ respectively w.r.t. power and area. We see that configuration $\{2\}$ is present in all three replacement sets. Therefore, if $\{2\}$ were available as a cost-effective, generic, pre-tested and pre-approved component as envisioned, application #D can be replaced without loss of functionality; #C, #F could be accommodated by adding a hardware accelerator in order to deliver the required performance. Such a hardware accelerator is feasible as long as it falls in the power and area margin provided by $\{2\}$ as compared to the application in question.

B. Discussion

From the above analysis, we can make the following observations: First off, through this study we provide experimental evidence that existing implants are very diverse but also seriously overdesigned embedded systems. They address medical applications through ad-hoc device implementations which are lacking a systematic design approach. A more

structured and top-down approach needs to be asserted if we want to exploit the benefits microelectronics technology has to offer these days.

One step towards this direction is the careful design of a generic processor family $\mathbb{P}$ which can service a wide number of applications. This generic processor family must have at least one processor from $\{10, 13, 14, 15, 17, 18, 20\}$ in order to satisfy applications #A, #E and #B; and configuration $\{2\}$ in order to satisfy #C, #D and #F. Out of the former set, we observe that $\{15\}$ has the least area and $\{20\}$ the least power. Since area and power are both of primary concern in a constrained implant, the generic-processor family may contain both these processors. The implant designer may, then, choose either of these processors depending on which of these two constraints is more pressing for the (unknown) application in question. Therefore, the family of processors chosen is $\{2, 15, 20\}$.

VI. Conclusions

In this paper, we have presented a complete approach towards systematic, educated and automated microarchitectural specification of processors for biomedical, microelectronic implants. We have provided 19 Pareto-optimal processor alternatives investigating a large set of hardware parameters such as I-cache and D-cache geometries, branch-prediction policy and memory latency. To the best of our knowledge, we have also provided the first comparison between the suggested processor configurations and existing, documented implantable devices across a wide range of applications. To manage this, we have established means of direct comparison based on careful assumptions that take into account the unavoidable inaccuracies of our tools. In doing so, we have proposed processors that can operate under worst-case conditions, i.e. they are suitably provisioned for the mission-critical implant applications.

In the future, we intend to expand our DSE framework to also optimize for system reliability in order to ensure error-free operation of critical implant applications. For this, we need to introduce a fourth metric based on reliability, and expand our tools accordingly. Work has already begun on porting XEEMU to our system as a more bug-free and accurate replacement for XTREM. Finally, we would also like to include more real-life applications in our studies – however, this is influenced by the extremely limited information released for this field.

VII. Acknowledgements

References

[1] C. Strydis *et al.*, "Implantable microelectronic devices: A comprehensive review," Computer Engineering, TU Delft," CE-TR-2006-01, Dec. 2006.

[2] R. Sanders and M. Lee, "Implantable pacemakers," in *Proceedings of the IEEE*, vol. 84, Mar. 1996, pp. 480–486.

[3] F. Nebeker, "Golden accomplishments in biomedical engineering," in *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, Piscataway, NJ, USA, May - June 2002, pp. 17–47.

[4] C. Strydis and G. Gaydadjiev, "The Case for a Generic Implant Processor," in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'08)*, August 2008, pp. 3186–3191.

[5] "Smart implantable Medical Systems," http://sims.et.tudelft.nl.

[6] K. Fernald, T. Cook, T. M. III, and J. Paulos, "A Microprocessor-Based Implantable Telemetry System," in *IEEE Computer*, vol. 24, Mar. 1991, pp. 23–30.

[7] K. Fernald, B. Stackhouse, J. Paulos, and T. Miller, "A System Architecture for Intelligent Implantable Biotelemetry Instruments," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, vol. 11, Nov. 1989, pp. 1411–1412.

[8] T. Miller, B. Bhuva, R. Barnes, J. Duh, H. Lin, and D. V. den Bout, "The Hector Microprocessor," in *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, 1986, pp. 406–411.

[9] B. Smith, Z. Tang, M. Johnson, S. Pourmehdi, M. Gazdik, J. Buckett, and P. Peckham, "An externally powered, multichannel, implantable stimulator-telemeter for control of paralyzed muscle," in *IEEE Transactions on Biomedical Engineering*, vol. 45, 1998, pp. 463–475.

[10] S. Pourmehdi, P. Strojnik, P. Peckham, J. Buckett, and B. Smith, "A custom-designed chip to control an implantable stimulator and telemetry system for control of paralyzed muscles," in *Artificial Organs*, vol. 23, May 1999, pp. 396–398.

[11] P. Valdastri, A. Menciassi, A. Arena, C. Caccamo, and P. Dario, "An implantable telemetry platform system for in vivo monitoring of physiological parameters," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, Sept. 2004, pp. 271–278.

[12] S. Salmons, G. Gunning, I. Taylor, S. Grainger, D. Hitchings, J. Blackhurst, and J. Jarvis, "ASIC or PIC ? Implantable stimulators based on semi-custom CMOS technology or low-power microcontroller architecture," in *Medical Engineering & Physics*, vol. 23, 2001, pp. 37–43.

[13] I. T. R. for Semiconductors (ITRS), "[online] available: http://www.itrs.net/common/2004update/2004update.htm," 2004.

[14] D. Dave, C. Strydis, and G. N. Gaydadjiev, "ImpEDE: A Multidimensional Design-Space Exploration Framework for Biomedical-Implant Processors," in *To appear in: Proceedings of the 21th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP'10)*, July 7-9 2010.

[15] G. Contreras *et al.*, "XTREM: A Power Simulator for the Intel XScale Core," in *LCTES'04*, 2004, pp. 115–125.

[16] C. Strydis and G. Gaydadjiev, "Suitable cache organizations for a novel biomedical-implant architecture," in *International Conference of Computer Design (ICCD'08)*, Lake Tahoe, California, USA, 12-15 October 2008, pp. 591–598.

[17] C. Strydis and G. N. Gaydadjiev, "Evaluating Various Branch-Prediction Schemes for Biomedical-Implant Processors," in *Proceedings of the 20th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP'09)*, July 2009, pp. 169–176.

[18] D. Dave, "Automated implant-processor design: An evolutionary multiobjective exploration framework," Master's thesis, TU Delft, 2010.

[19] Z. Herczeg, A. Kiss, D. Schmidt, N. Wehn, and T. Gyimóthy, "XEEMU: An Improved XScale Power Simulator," *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, pp. 300–309, 2007.

[20] C. Strydis, D.Dave, and G. Gaydadjiev, "ImpBench revisited: An extended characterization of implant-processor benchmarks," in *Submitted to: International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS'10)*, Samos, Greece, 2010.

[21] S. Pourmehdi, P. Strojnik, P. Peckham, J. Buckett, and B. Smith, "A custom-designed chip to control an implantable stimulator and telemetry system for control of paralyzed muscles," in *Proceedings of the 6th Vienna International Workshop on Functional Electrical Stimulation*, Vienna, Austria, 2224 September 1998.

[22] T. Eggers, C. Marschner, U. Marschner, B. Clasbrummel, R. Laur, and J. Binder, "Advanced hybrid integrated low-power telemetric pressure monitoringsystem for biomedical applications," in *MEMS'00*, 2000, pp. 329–334.

[23] ——, "Advanced hybrid integrated low-power telemetric pressure monitoring system for biomedical applications," in *IEEE Proceedings of Microelectromechanical Systems (MEMS)*, Miyuzaki, Japan, 2000, pp. 329–334.

[24] K. Hille, J. Draeger, T. Eggers, and P. Stegmaier, "[Technical construction, calibration and results with a new intraocular pressure sensor with telemetric transmission] [Article in German]," in *Klinische Monatsblatter fur Augenheilkunde*, vol. 218, May 2001, pp. 376–380.

[25] D. Rollins, C. Killingsworth, G. Walcott, R. Justice, R. Ideker, and W. Smith, "A telemetry system for the study of spontaneous cardiac arrhythmias," in *IEEE Transactions on Biomedical Engineering*, vol. 47, July 2000, pp. 887–892.

[26] K. Au-Yeung, C. Johnson, and P. Wolf, "A novel implantable cardiac telemetry system for studying atrial fibrillation," in *Physiological Measurement*, 11 August 2004, pp. 1223–1238.

[27] C. Liang, J. Chen, C. Chung, C. Cheng, and C. Wang, "An implantable bi-directional wireless transmission system for transcutaneous biological signal recording," in *Physiological Measurement*, vol. 26, Feb. 2005, pp. 83–97.