

The Case for a Generic Implant Processor

Christos Strydis, Georgi N. Gaydadjiev
Computer Engineering Lab, Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands
Phone: +31-(0)15-27-83591 E-mail: christos@ce.et.tudelft.nl

Abstract—A more structured and streamlined design of implants is nowadays possible. In this paper we focus on implant processors located in the heart of implantable systems. We present a real and representative biomedical-application scenario where such a new processor can be employed. Based on a suitably selected processor simulator, various operational aspects of the application are being monitored. Findings on performance, cache behavior, branch prediction, power consumption, energy expenditure and instruction mixes are presented and analyzed. The suitability of such an implant processor and directions for future work are given.

Index Terms—implant, low power, low energy, compression, encryption, checksum, microarchitecture, personalized healthcare

I. INTRODUCTION

In the face of current socioeconomical and ongoing technological advances, healthcare in the 21st century is changing rapidly. Healthcare in advanced countries is slowly moving from a public to a more personalized nature. In advanced countries the following cascading trends are currently being observed:

- population is aging through a net reduction in birth rates combined with an increase in life expectancy;
- healthcare costs are increasing;
- customized, ad-hoc healthcare solutions are sought; and
- higher demands for betterment of quality of life are placed (health, fitness, convenience etc.).

Present healthcare systems seem to undeviatingly follow the New-Public-Management (NPM) paradigm. This paradigm claims that, under conditions of heavy public demands but a severely constrained public budget, the only feasible alternative to cutting public services or raising taxes, seems to be to reduce costs, increase effectiveness and efficiency, and deliver "more value for the money" [1], [2]. Presently observed cost overruns and inefficiencies are clear indications of systemic failures in the existent healthcare construct.

In the legal domain, governmental parties in many countries are now attempting to preempt the coming change by revising the standing legislation and passing new one in order to cope with this new era [3]. In the technological field, rapid advances in key areas of science like microelectronics and micromachining technologies as well as the gradual maturing of computer-architecture and compiler designs have untied engineers' hands and have enabled unprecedented improvements in various areas of the medical discipline.

Molecular biology, novel medical-imaging techniques, pharmacogenomics and microelectronic implants are only few of the benefitted areas. Such societal needs as the ones previously described can and will, unavoidably, use technology as their vehicle, a trend already witnessed in the cell-phone and portable-computing revolutions. Nowadays, everyone is talking about "ubiquitous computing", that is, computing anytime and anywhere.

The trends towards personalized healthcare are partly driven by this "ubiquity" trend. A number of technological innovations are currently attempting to carry healthcare systems to the next level, such as wearable electronics, portable medical monitors and body-area networks (BANs). Towards the same end, a promising field of biomedical engineering is microelectronic implants, such as the infamous implantable pacemakers and cochlear implants. The implantable pacemaker in particular, apart from saving lives, has acted as a catalyst on the general public closed-mindedness against biomedical implants. Indicative of the penetration and impact pacemakers have achieved is the fact that, in the U.S. alone, a total number of 180,000 implantable pacemakers have been registered for the year 2005 (source: American Heart Association [4]).

Implants have been around for more than 50 years, yet over the last decade they have clearly benefitted from the technology miniaturization trends, such as smaller sizes, lower power consumption and increased performance. In effect, they are now being designed for a large, and constantly increasing, range of applications. These applications are primarily grouped into two main categories: physiological-parameter monitoring (for diagnostic purposes) and stimulation (actuation, in general). Instances of the former are devices measuring body temperature [5], blood pressure [6], blood-glucose concentration [7], gastric pressure [8], tissue bio-impedance [9] and more. In the latter category belong pacemakers [10], [11] and implantable intracardiac defibrillators (ICDs) [12], various functional electrical stimulators for paralyzed extremities [13], for bladder control [14], for blurred-eye cornea [15] and more pathoses. For a more involved discussion on the current state of the art, the interested reader can refer to [16].

This plethora of existing and future implant applications gives explicit directions towards a more structured approach in the design of microelectronic implants. In our ongoing research we are primarily interested in the implant processor residing in the core of microelectronic implants and directing their functionality. We advocate the design of a generic

processor with explicit provisions for fault-tolerant operation and suitable for covering a large subset of applications as the ones previously mentioned. In order to do so, we have studied a large number of implantable systems. In the current work, we present a typical implant application whose functionality is largely implemented as executed software in the envisioned processor. Through the use of a suitable processor simulator, we profile various application aspects using diverse metrics. Concisely, the contributions of this work are:

- to quantify performance, power and energy metrics as well as instruction mixes for our implant processor;
- to identify microarchitectural traits such as popular instructions and potential optimizations; and
- to offer a proof-of-concept application of the implant processor and, thus, exhibit its viability, usefulness and potential in future implant design.

The rest of the paper is organized as follows: in section II the motivation behind the design of a novel, generic processor for implantable devices is explained. Section III discusses typical implant-application characteristics and introduces the profiled implant case study. Section IV discusses the chosen processor simulator used for running our experiments. In section V we present our experimental results and provide detailed discussion. Overall conclusions and future work are discussed in section VI.

II. A FRAMEWORK FOR MICROELECTRONIC-IMPLANT PROCESSORS

With a market finally mature enough to embrace implants and the technological innovations of late to support them, implant designers are slowly changing their approach. Already established product cases such as the family of pacemakers introduced by Medtronic [17], where previous design expertise is (re)used to enhance the next device version, are currently the exception. It has come to our attention that implant design has been largely custom-based; that is, implants have been developed as ASIC circuits tightly fitting the application requirements at hand.

However, this is nowadays changing with implants moving from custom-designed, application-specific - e.g. Finite-State-Machine (FSM)-based systems [18], [19], [20] to more generic and software-based ($\mu P/\mu C$ -based) ones [21], [22], [23]. This trend has been well-studied [16] and is depicted in Fig.1. What the figure tells us is that implant-processor design is becoming more streamlined and structured than it used to be and that, in the near future, implant functionality will be based on executed software (written in some high-level, established language like C) rather than on hardwired circuits.

With the list of potential implant applications constantly expanding and the number of software-based implant solutions increasing, the need for a formal, standardized way of designing future implant architectures becomes apparent. Our long-term work focuses on designing a novel, minimalistic, low-power and fault-tolerant processor suitable for a large subset of biomedical applications as the ones mentioned above. We are currently defining the architecture of such a digital processor.

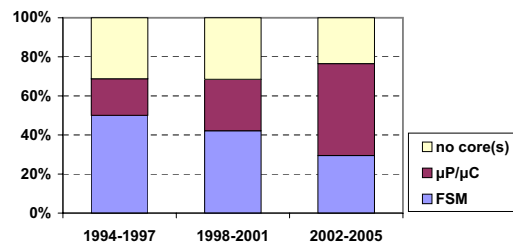


Fig. 1. Relative distribution of implant-core architecture types over the last 12 years (Source: [16]).

So far, extensive work has been performed for identifying and profiling common applications to be executed on such an architecture. Algorithms for lossless data compression [24], symmetric-key encryption [25] and error detection/correction as well as representative real-world applications have been evaluated and suitable candidates have been isolated. Moreover, a carefully selected benchmark suite for microelectronic implants has been proposed [26] to guide and assist future implant design. In this work we build upon our previous findings and present the detailed case study of a typical implant application that can be serviced by our envisioned, novel processor. The case study is implemented on a properly modeled processor simulator.

III. THE GENERIC-IMPLANT CASE STUDY

In order to study and simulate a representative implant application, commonly met characteristics of implantable systems need to be identified. Our prior work [16] has revealed the following facts. First, biomedical implants perform periodic, in-vivo measurements of physiological data through appropriate sensors. The collected data need to be stored inside the implant for later telemetry to an external monitoring/logging device. Second, data must be transmitted securely as well as reliably; information eavesdropping or loss thereof can not be tolerated. Third, open- or closed-loop control of (in-vivo) physiological parameters may be effectuated through appropriate actuators, e.g. the "artificial pancreas" application whereby insulin is released to the blood based on periodic, in-vivo, glucose-level measurements. Fourth, biological or other data manipulation in implants can in most cases be coped with through integer (INT) arithmetic. Expensive, floating-point (FP) operations can be avoided by smart manipulation of the data or postponed until the time when data is telemetered to an external logging station with infinite (in our context) computational resources. Last, typical data-memory sizes inside the implants range from 1 *KB* to 10 *KB*. Program memories are equally restricted, with sizes in the order of magnitude of 10 *KB*.

Rather than creating an artificial and, thus, potentially biased application based on synthetic application descriptions, we chose to use a real-world scenario. Cross et al. [22] have developed intravaginal drug-delivery & monitoring units (DMUs) for regulating the oestrus cycle of dairy cows. The functionality of each DMU is implemented as embedded-C code running in a M16C, a 16-bit microcontroller (μC) from Mitsubishi. This μC is the central component in a system consisting of a transceiver module, temperature, pressure, motion and other sensors as well as a current-driven gas

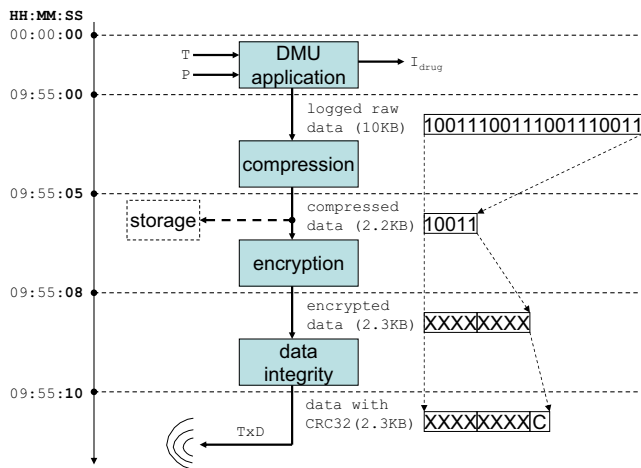


Fig. 2. Conceptual block diagram of simulated implant application and data-payload sizes.

cell (i.e. an actuator) which is used for controlled drug release based on electrolytic-gas production. According to the authors, the DMUs have been designed: i) to deliver an arbitrary and complex variable-rate profile of a viscous vehicle, ii) to be controlled externally from the animal, and iii) to be monitored externally and provide immediate or logged data over a wireless link.

We have extracted the embedded-C code and have adapted it from the implantable system. The current program version does (and can) not simulate all real-time aspects of the actual (interrupt-driven) system, such as low-level functionality (e.g. sensor/actuator calibrations), transceiver operation and so on. Nonetheless, the emphasis here is on the computations performed by the implant core in response to external and internal events (i.e. interrupts). Having contacted the DMU designers directly, we have acquired real data collected from the field (e.g. temperature, pressure and current output). They have been used in our source code to drive the (simulated) run-time behavior of the actual DMU system as closely as possible.

This particular application has been selected since it incorporates all aspects we consider common and crucial in current and future implants. That is, real-time, closed-loop control of actuating elements based on sensory read-outs, device self-calibration and self-check operations (e.g. battery-level check, adherence to the desired drug-delivery profile etc.), to name a few. At the same time, the application imposes low- to moderate-speed requirements on the device which, for our targeted field of ultra-low-power implants, is a desired feature. All in all, the selected application is considered highly representative for our envisioned biomedical processor.

The basic DMU functionality has been enhanced with data compression, encryption and data-integrity runs which we consider crucial tasks for future implant applications. The functionality of our overall case study is illustrated as a block diagram in Fig.2. Over a period of approx. 10 (simulated) hours, the implant periodically (i.e. every 6 min) collects intravaginal temperature- and pressure-sensor readings and logs them. Based on those readings, it switches the gas cell

feature	value
ISA	32-bit ARMv5TE-compatible
Pipel. depth / Datap. width	7/8-stage, super-pipelined / 32-bit
RF size	16 registers
Issue policy / Instr.window	in-order / single-instruction
I-Cache, L1	64KB, 64-way assoc. (1cc hit/170cc miss)
D-Cache, L1	32KB, 2-way assoc. (1cc hit/170cc miss)
TLB / BTB	1-entry fully-assoc.
Branch Predictor	2-bit Bimodal (32-entry ret. addr. stack)
Write Buffer / Fill Buffer	2-entry / 2-entry
Mem. port no / bus width	1 port / 1 Byte
INT/FP ALUs	1/1
Clock freq. / Implem. tech.	2 MHz / 0.18 μm @ 1.5 Volt

TABLE I

XTREM (MODIFIED) ARCHITECTURE DETAILS.

on and off. This gas cell is responsible for the rate of drug delivery into the animal intravaginal space, following a user-defined drug-delivery profile. Besides, every 40 minutes, the implant performs some housekeeping tasks like safety checks and recalibrations of the sensors and actuators.

At the end of the 10 simulated hours (pure DMU operation is finished), logged data is compressed and remain stored in native memory or transmitted to an external host. Transmitted data are first compressed, then encrypted and, finally, augmented with data-integrity check bits. In order to comply with the previously described specifications, data logs of maximally 10 KB each have been generated. All above tasks are performed in software by the implant processor. Based on our previous work, suitable algorithms in terms of performance, power, energy and size have been used for the compression (miniLZO [27]), encryption (MISTY1 [28]) and data-integrity (CRC32 [29]) operations.

IV. EXPERIMENTAL SETUP

Simulation of our implant application has been based on XTREM [30], a modified version of SimpleScalar [31]. The XTREM simulator is a cycle-accurate, microarchitectural, power- and performance- functional simulator for the Intel XScale core. It models the effective switching node capacitance of various functional units inside the core, following a similar modeling methodology to the one found in Wattch [32]. XTREM has been selected for its straight-forward functionality but mostly for its high precision in modeling the performance and power of the Intel XScale core [33]. More precisely, it exhibits an average performance error of only 6.5% and an average power error of only 4%.

Main XTREM characteristics are summarized in Table I. XTREM allows monitoring of 14 different functional units of the Intel XScale core: Instruction Decoder (DEC), Branch-Target Buffer (BTB), Fill Buffer (FB), Write Buffer (WB), Pend Buffer (PB), Register File (REG), Instruction Cache (I\$), Data Cache (D\$), Arithmetic-Logic Unit (ALU), Shift Unit (SHF), Multiplier Accumulator (MAC), Internal Memory Bus (MEM), Memory Manager (MM) and Clock (CLK). Various XTREM architectural parameters have been further reduced or cut down to better reflect the highly constrained implant processor (also included in the table above). Concisely, the BTB has been reduced to a 2-entry, direct-mapped structure, the WB and the FB have been reduced also to 2-entry structures, the MEM width has been

reduced to 1 Byte, both L2 caches have been disabled, both L1 caches have been configured based on a prior optimization study [34] while the number of INT/FP ALUs has been reduced to 1. Performance and power figures have been checked and scale properly with the changes.

V. EXPERIMENTAL RESULTS

In order to gain insight on the behavior and requirements of the tasks executed inside the implant, various metrics have been monitored and concisely presented hereafter. Unless otherwise stated, reported average values are actually median values since normal distribution of the data is not generally guaranteed. As we can see from Fig. 2, tasks are executed in a sequential fashion. **Execution times** are sufficiently small for this real-time application as is the case with most implantable systems. MiniLZO compression for a 10 – *KB* data payload is high (78%) and achieved in about 5.1 *sec*. Encryption adds a small overhead in size to the compressed data due to quantization since MISTY1 operates on 8 – *Byte* quantities. It achieves symmetric-key encryption of the data in about 3.2 *sec*. Last, CRC32 data integrity adds a negligible size overhead of 4 *Bytes* to the payload by appending an unsigned-long-integer checksum value and costs an extra 1 *sec* in time.

An overall (simulated) real execution time of 9.3 *sec* is required to perform all data-manipulation tasks after the 10 – *KB* log file has been generated; that is, an extra processing time of 9.3 *sec* every 10 hours. Even though we are using a highly-resource constrained processor, the system response time is very low indicating a processor performance which is more than adequate for the subclass of moderate-throughput applications we are targeting. To illustrate, in Fig. 3 **Instructions Per Cycle (IPC)**, **cache- and branch-behavior** are depicted. The exceptionally low D-cache hits reveal strong data-locality characteristics of the biological data and hint on clear performance gains should larger D-cache sizes be allowed. Conversely, the high I-cache hits indicate that relatively small I-cache sizes (see [34] for the scaling factor assumed) are sufficient due to the highly predictable program behavior of the considered tasks. Given that we have used a relatively simple branch-prediction scheme (2-bit Bimodal), BPRED rates are rather high with miniLZO scoring exceptionally high. Its IPC though remains the smallest due to its low D-cache hit rates. Besides, IPC is low for all programs but, as discussed previously in the execution times, it is more than sufficient for covering the real-time-application demands of the implant.

In fact, the low IPCs - as long as they cover the demands of the application - are a desired feature since they imply limited power demands on the part of the processor. This is a much sought attribute in power-starved systems as implants are. To illustrate this, overall and per-component **average power-consumption** figures for all three tasks are depicted in Fig. 4. We can see that miniLZO consumes remarkably low power (about 20 *mW*) but, in general, all tasks consume less than 100 *mW*. The low power profile of miniLZO agrees with the lower IPC it exhibits, as previously predicted. We

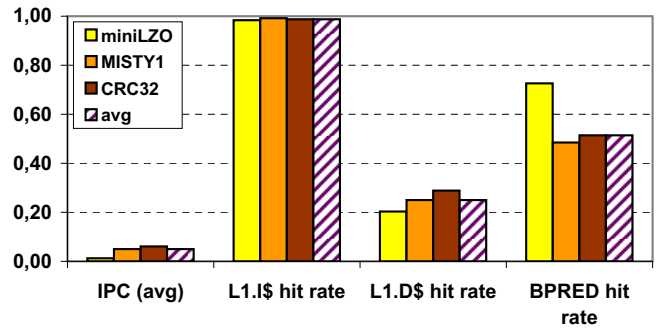


Fig. 3. Average IPCs, I/D-cache hit rates and branch-prediction rates.

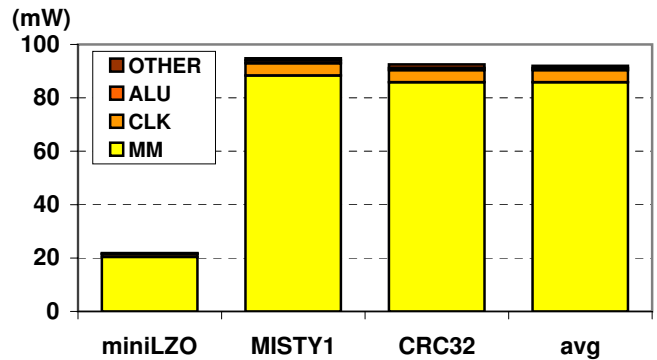


Fig. 4. Per-component and overall average power consumption.

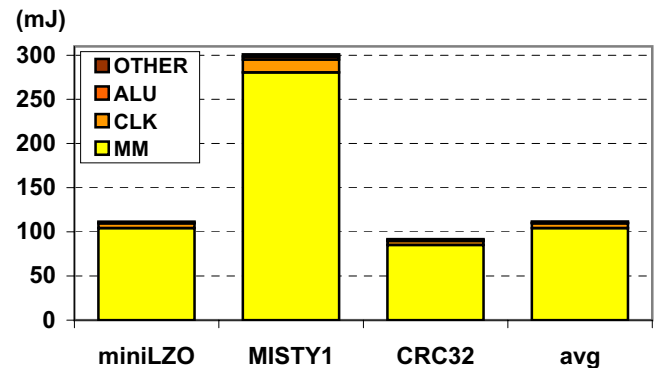


Fig. 5. Per-component and overall total energy expenditure.

can further deduce from the figure that the main culprit of power consumption in the processor is the memory-manager unit (MM), followed by the clock network (CLK). This indicates that the selected 2 – *MHz* operating frequency is high enough for the tasks to execute in time and, at the same time, low enough to impact power consumption minimally. It also indicates that in implantable systems as the one modeled here, the MM is under heavy use and should be carefully designed for low power consumption.

Except for average power consumption, it is interesting also to see what the **overall energy budgets** of the various tasks are; that is, by how much we must deplete the implant battery to perform each task. In Fig. 5 we can see that the encryption program, MISTY1, consumes a disproportionately large amount of energy compared to the other tasks. This indicates that we should carefully select whether to encrypt the biological data or not prior to transmission, depending on the application scenario and the sensitivity of the data itself. If privacy is not required or is guaranteed through other

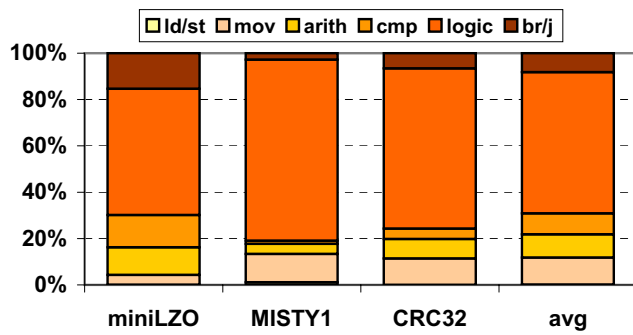


Fig. 6. Relative frequencies for load/store, move, arithmetic, compare, logical and branch/jump μ op.

means, e.g. transmission in a trusted environment, considerable battery reserves can be saved by disabling encryption. Alternatively, a compromise between level of provided security and consumed energy could be investigated. While this is not (currently) supported in MISTY1, future versions of it or other low-power encryption algorithms might be considered that are able to achieve such a trade-off. Overall, Fig. 5 reveals that the energy costs of the various tasks are not necessarily identical to their power profiles and is essential in deciding which tasks can be performed at a given point in time, based on available battery-capacity levels.

The final topic of our discussion relates to the **instruction mix** of the various tasks. XTREM, which is based on SimpleScalar, implements ARM instructions through (elementary) μ ops. We included μ op (rather than instruction) statistics at this point and in the following discussion so as to better capture the workings of the underlying architecture. Overall instruction mixes are shown in Fig. 6. All programs heavily utilize logical μ ops; MISTY1 expectedly scores the highest which is typical of encryption algorithms. In terms of arithmetic operations, it should be stressed that all tasks (except DMU) are integer programs and miniLZO displays the highest concentration of arithmetic and compare operations. It also includes the largest ratio of branch or jump μ ops. MISTY1 and CRC32, on the contrary, exhibit larger ratios of data move μ ops.

In Fig. 7, we further collect (dynamic) **data-dependent μ op pairs and triplets**. μ op pairs or triplets are consecutive μ ops whereby data generated by the first μ op is consumed by the second and/or third μ op; i.e. whereby data dependencies occur. We have limited the plot to only those combinations appearing with a frequency of 4% or more during dynamic-code execution. With this constraint we see that, overall, dependent "and-eor" (and: logical and) and "eor-eor" (eor: logical exclusive-or) pairs are by far the most frequent ones, followed by "eor-cmp" (cmp: compare) pairs. This observation reveals a high popularity of dependent logical- μ op pairs. We, thus, get a clear indication that data-forwarding in the logical-operation part of the ALU, interlock-collapsing-ALU techniques [35] or other (micro)architectural optimizations will significantly benefit the implant processor. Further, the "eor-cmp" pair, combined with the previously seen μ op mixes, gives directions on optimizing the compare-and-branch subsystem of the processor. Last but not least,

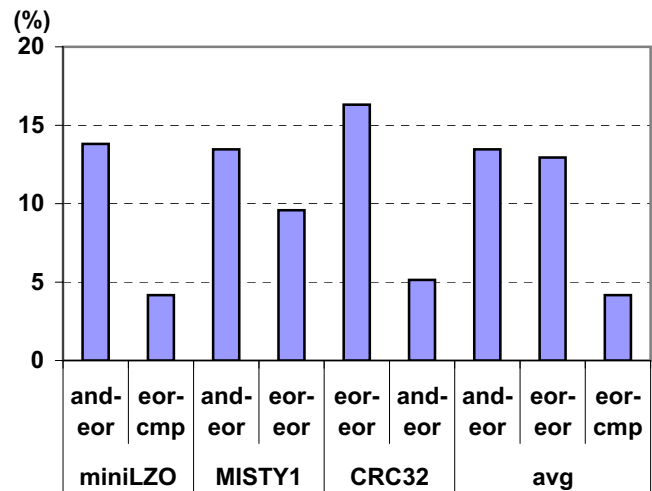


Fig. 7. Relative frequencies of data-dependent, dynamic- μ op combinations.

all above observations on μ op frequencies can give clear directions as to which instructions should be explicitly implemented in hardware and which ones can be afforded to be implemented in software (compiler-side conversion).

To sum up our findings, based on the selected biomedical application, we can support the viability of a highly resource-constrained, novel processor for implants. Featuring a low as 2 – MHz clock frequency and small I/D-caches, it will be able to meet its real-time goals for a broad range of application scenarios similar to or simpler than the one described here. Furthermore, it will feature a low average power profile of less than 100 mW, and - excluding data encryption - a similarly low energy profile of less than 300 J per executed task. It should be noted, however, that at the area and power penalty of a slightly increased D-cache size, program execution times will drop significantly as the simulations indicate. This will, in turn, lead to an even lower energy profile. Besides, reported power/energy figures are likely to be higher than actual ones since the XTREM simulator was not aimed at the ultra-low-power application spectrum. What is more, compression and encryption algorithms designed with implantable systems in mind, should assist further in this direction. Last, explicit microarchitectural optimizations of the envisioned processor will drive power and energy figures further down. Hardware provisions for favorable execution of *logical* and, secondarily, *arithmetic/compare* operations as well as of specific *logical/compare* μ op pairs must be incorporated in the design.

VI. CONCLUSIONS

In this paper we have qualitatively discussed the changes modern healthcare systems are undergoing and identified biomedical implants as a potential technological vehicle to cope with them. We have also traced the current trends in implant design (based on our previous work) and the need for more structured design approaches in the years to come. We directed our focus on the processors for such devices and have presented an implant-processor case study based on a realistic as well as representative biomedical application. The selection of executed tasks such as compression and

encryption has also been based on our previous study and work in the field. On this basis, we have used a suitably modified, highly accurate power and performance simulator to generate various run-time results such as IPC, power consumption and instruction mixes. Through those results, we have shown that such a structured processor design approach for implants is possible. Moreover, we have offered design hints and potential areas of further research such as power-aware data encryption and microarchitectural optimizations for performance improvement.

VII. ACKNOWLEDGEMENTS

This work has been partially supported by the ICT Delft Research Centre (DRC-ICT) of the Delft University of Technology. Thanks are due to Gilberto Contreras for an excellent power simulation tool and Peter Cross for the interesting DMU application source code. Also, many thanks are due to Niki Frantzeskaki for her valuable comments and help.

REFERENCES

- [1] W. Kickert, "Public Governance in the Netherlands: An Alternative to Anglo-American Managerialism," *Public administration*, vol. 75, no. 4, pp. 731–752, 1997.
- [2] J. Christensen and T. Pallesen, "Institutions, distributional concerns, and public sector reform," *European Journal of Political Research*, vol. 39, no. 2, pp. 179–202, 2001.
- [3] H. Nys, L. Stultiens, P. Borry, T. Goffin, and K. Dierickx, "Patient rights in EU Member States after the ratification of the Convention on Human Rights and Biomedicine," *Health Policy*, vol. 83, no. 2-3, pp. 223–235, October 2007.
- [4] H. Ector and P. Vardas, "Heart disease and stroke statistics - 2008 Update (At-a-Glance Version)," *American Heart Association*, 2008.
- [5] P. Wouters, M. D. Cooman, D. Lapadatu, and R. Puers, "A low power multi-sensor interface for injectable microprocessor-based animal monitoring system," in *Sensors and Actuators A: Physical*, vol. 41-42, 1994, pp. 198–206.
- [6] B. Flick and R. Orglmeister, "A portable microsystem-based telemetric pressure and temperature measurement unit," in *IEEE Transactions on Biomedical Engineering*, vol. 47, Jan. 2000, pp. 12–16.
- [7] M. Shults, R. Rhodes, S. Updike, B. Gilligan, and W. Reining, "A telemetry-instrumentation system for monitoring multiple subcutaneously implanted glucose sensors," in *IEEE Transactions on Biomedical Engineering*, vol. 41, Oct. 1994, pp. 937–942.
- [8] P. Valdastri, A. Mencicassi, 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.
- [9] M. Min, T. Parve, V. Kukk, and A. Kuhlberg, "An implantable analyzer of bio-impedance dynamics - mixed signal approach," in *IEEE Instrumentation and Measurement*, Budapest, Hungary, 21-23 May 2001, pp. 38–43.
- [10] J. Berkman and J. Prak, "Biomedical microprocessor with analog I/O," in *IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, 19 February 1981, pp. 168–169.
- [11] C. Harrigal and R. Walters, "The development of a microprocessor controlled implantable device," in *IEEE Proceedings of the 1990 Sixteenth Annual Northeast Bioengineering Conference*, Mar. 1990, pp. 137–138.
- [12] J. Warren, R. Dreher, R. Jaworski, J. Putzke, and R. Russie, "Implantable cardioverter defibrillators," in *Proceedings of the IEEE*, vol. 84, 1996, pp. 468–479.
- [13] 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.
- [14] M. Sawan, S. Robin, B. Provost, Y. Eid, and K. Arabi, "A wireless implantable electrical stimulator based on two FPGAs," in *Proceedings of the IEEE International Conference on Electronic Circuits and Systems (ICECS)*, vol. 2, Piscataway, New Jersey, USA, 1996, pp. 1092–1095.
- [15] M. Schwarz, L. Ewe, R. Hauschild, B. Hosticka, J. Huppertz, S. Kolnsberg, W. Mokwa, and H. Trieu, "Single chip CMOS imagers and flexible microelectronic stimulators for a retina implant system," in *Sensors and Actuators A: Physical*, vol. 83, 22 May 2000, pp. 40–46.
- [16] C. Strydis, G. Gaydadjiev, and S. Vassiliadis, "Implantable microelectronic devices: A comprehensive review," *Computer Engineering*, Delft University of Technology, CE-TR-2006-01, December 2006.
- [17] "Medtronic - Cardiology product list," <http://www.medtronic.com/physician/cardiology.html>.
- [18] M. Ghovanloo and K. Najafi, "A modular 32-site wireless neural stimulation microsystem," in *IEEE Journal of Solid-State Circuits*, vol. 39, December 2004, pp. 2457–2466.
- [19] P. Mohseni and K. Najafi, "Wireless multichannel biopotential recording using an integrated FM telemetry circuit," in *26th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society (EMBS)*, San Francisco, CA, USA, 1-5 September 2004, pp. 4083–4086.
- [20] H. Park, H. Nam, B. Song, and J. Cho, "Design of miniaturized telemetry module for bi-directional wireless endoscopy," *IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences*, vol. 6, pp. 1487–1491, June 2003.
- [21] C. Liang, J. Chen, C. Chung, C. Cheng, and C. Wang, "An implantable bi-directional wireless transmission system for transcutaneous biological signal recording," *Physiological Measurement*, vol. 26, pp. 83–97, February 2005.
- [22] P. Cross, R. Kunnemeyer, C. Bunt, D. Carnegie, and M. Rathbone, "Control, communication and monitoring of intravaginal drug delivery in dairy cows," in *International Journal of Pharmaceuticals*, vol. 282, 10 September 2004, pp. 35–44.
- [23] H. Lanmuller, E. Unger, M. Reichel, Z. Ashley, W. Mayr, and A. Tschakert, "Implantable stimulator for the conditioning of denervated muscles in rabbit," in *8th Vienna International Workshop on Functional Electrical Stimulation*, Vienna, Austria, 10-13 September 2004.
- [24] C. Strydis and G. Gaydadjiev, "Profiling of Lossless-Compression Algorithms for a Novel Biomedical-Implant Architecture," in *To appear in International Conference on Hardware-Software Codesign and System Synthesis (CODES'08)*, Atlanta, Georgia, 19-24 October 2008.
- [25] C. Strydis, D. Zhu, and G. Gaydadjiev, "Profiling of symmetric encryption algorithms for a novel biomedical-implant architecture," in *To appear in ACM International Conference on Computing Frontiers (CF'08)*, Ischia, Italy, 5-7 May 2008.
- [26] C. Strydis, C. Kachris, and G. Gaydadjiev, "ImpBench - A novel benchmark suite for biomedical, microelectronic implants," in *To appear in International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS'08)*, Samos, Greece, 21-24 July 2008.
- [27] M. Oberhumer, "LZO v2.0.2," <http://www.oberhumer.com/opensource/lzo/>.
- [28] Y. Law, J. Dourmen, and P. Hartel, "Survey and benchmark of block ciphers for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, pp. 65–93, February 2006.
- [29] "Cell Relay Retreat: CRC-32 Calculation, Test Cases and HEC Tutorial," <http://cell.onecall.net/cell-relay/publications/software/>.
- [30] G. Contreras, M. Martonosi, J. Peng, R. Ju, and G.-Y. Lueh, "XTREM: A Power Simulator for the Intel XScale Core," in *LCES'04*, 2004, pp. 115–125.
- [31] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: an infrastructure for computer system modeling," *IEEE Computer*, vol. 35, no. 2, pp. 59–67, February 2002.
- [32] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," in *ISCA'00*, 2000, pp. 83–94.
- [33] *Intel XScale Microarchitecture for the PXA255 Processor: User's Manual*, Intel Corporation, March 2003.
- [34] C. Strydis and G. Gaydadjiev, "Suitable cache organizations for a novel biomedical-implant architecture," in *Submitted to International Conference of Computer Design (ICCD'08)*, Lake Tahoe, California, USA, 12-15 October 2008.
- [35] S. Vassiliadis, J. Phillips, and B. Blaner, "Interlock collapsing ALU's," *IEEE Transactions on Computers*, vol. 42, no. 7, pp. 825–839, Jul 1993.