

# A Multiprocessor System-on-Chip for Real-Time Biomedical Monitoring and Analysis: Architectural Design Space Exploration

Iyad Al Khatib

IMIT, ICT, KTH  
Royal Institute of  
Technology, Sweden  
+4687904111  
iyad@imit.kth.se

Francesco Poletti

DEIS  
University of Bologna  
Bologna, Italy  
+390512093782  
fpoletti@deis.unibo.it

Davide Bertozzi

ENDIF  
University of Ferrara  
Ferrara, Italy  
+390532974832  
dbertozzi@ing.unife.it

Luca Benini

DEIS  
University of Bologna,  
Bologna, Italy  
+390512093782  
lbenini@deis.unibo.it

Mohamed Bechara

ECE, FEA  
American University of  
Beirut, Lebanon  
+9613051285  
mfb05@aub.edu.lb

Hasan Khalifeh

ECE, FEA  
American University of  
Beirut, Lebanon  
+9613643891  
hik05@aub.edu.lb

Axel Jantsch

IMIT, ICT, KTH  
Royal Institute of  
Technology, Sweden  
+4687904124  
axel@imit.kth.se

Rustam Nabiev

Biomedical Engineering  
Karolinska University  
Hospital, Sweden  
+46858586288  
rustam.nabiev@kus.se

## ABSTRACT

In this paper we focus on MPSoC architectures for human heart ECG real-time monitoring and analysis. This is a very relevant bio-medical application, with a huge potential market, hence it is an ideal target for an application-specific SoC implementation. We investigate a symmetric multi-processor architecture based on STMicroelectronics VLIW DSPs that process in real-time 12-lead ECG signals. This architecture improves upon state-of-the-art SoC designs for ECG analysis in its ability to analyze the full 12 leads in real-time, even with high sampling frequencies, and ability to detect heart malfunction. We explore the design space by considering a number of hardware and software architectural options.

## Categories and Subject Descriptors

C.3 [Special-Purpose and Application-Based Systems]: Microprocessor/microcomputer applications, Real-time and embedded systems

**General Terms:** Performance, Design, Experimentation.

**Keywords:** Multiprocessor System-on-Chip, embedded system design, electrocardiogram algorithms, real-time analysis, hardware space exploration

## 1. INTRODUCTION

The advance in embedded systems and multiprocessor trends pave the way for the development of single-chip solutions for computationally-intensive bio-medical applications with huge potential health benefits

for a large number of individuals. One important application, in this respect, is the real-time remote and accurate analysis of human heart activity, which has always been a challenging problem for biomedical engineers. Heart disorders like Cardiovascular Disease (CVD) and stroke remain by far the leading cause of death in the world for both women and men of all ethnic backgrounds. In 2003, CVD alone is responsible for 29.2% of the total global deaths according to the World Health Organization (WHO), and this percentage is increasing every year [1]. More than 50% of these deaths can be saved with a reliable combination of cost-effective monitoring and accurate analysis [1].

Heart activity is electrically recorded as a set of electrocardiogram (ECG) signals which can readily reveal a number of heart malfunctions [2][3][4]. The most reliable ECG analysis technique is the 12-lead ECG, which requires the reading and analysis of twelve different signals sensed from the patient's body. The main challenge arises from the high computational demand for processing huge amounts of ECG data in parallel, under stringent time constraints, relatively high sampling frequencies, and life-critical conditions [5]. The challenges become even more complex when the patient is mobile and remotely monitored (as in cases of homecare and emergency at the point-of-need) [6], because state-of-the-art biomedical equipment for heart monitoring lack the ability to provide large-scale analysis and remote, real-time computation at the patient's location. This necessitates the transmission of huge amounts of life-critical data over a communication link to a large set of computing devices on another location [3]. In cases of mobile patients, this requires a 100% functional always-ON wireless connection since losing a few heart-beat data may be life threatening.

To overcome the aforementioned challenges and the problem of transmitting life-critical data on a wireless link (that is not reliable nor secure enough), the solution is to parallel-process the complex biomedical computations of the 12-lead ECG on a wearable multiprocessor system-on-chip (MPSoC). Hence, the solution only transmits secure remote-alarm signals and only reports on the *results* of the analysis. These result-reports, are much smaller in size (a few bytes) than the ECG data (in Mega bytes), and if transmission fails, they can be re-transmitted until reception is acknowledged by the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.  
Copyright 2006 ACM 1-59593-381-6/06/0007...\$5.00.

healthcare remote-monitoring center since they are saved on an off-chip memory for every analyzed ECG data chunk.

This technical objective calls for the design of special-purpose SoC architectures, featuring increased energy efficiency while providing high computation capabilities. In this paper we introduce a novel MPSoC architecture for ECG analysis which improves upon state-of-the-art mostly for its capability to perform a number of real-time analyses of input data with high sampling frequencies, leveraging the computation horsepower and the functional flexibility provided by many (up to twelve) concurrent DSPs. The proposed architecture addresses usability, security and safety of the patients in emergency situations and long-term treatments. Comparison between our design and previous work shows the advantages of our design from a SoC performance point of view and from an application point of view.

The biochip system builds upon some of the most advanced industrial components for MPSoC design (multi-issue VLIW DSPs, system interconnect from STMicroelectronics, and commercial off-the-shelf biomedical sensors), which have been composed in a scalable and flexible platform. Therefore, we have ensured its reusability for future generations of ECG analysis algorithms and its suitability for porting of other biomedical applications, in particular those collecting input data from wired/wireless sensor networks.

The paper goes through all the steps of the design methodology, from application functional specification to hardware definition and modeling. System performance has been validated through functional, timing accurate simulation on a virtual platform. A 0.13 $\mu$ m technology homogeneous power estimation framework leveraging industrial power models is used for power management considerations.

## 2. BACKGROUND

Biomedical sensors today exhibit increased energy efficiency, therefore prolonged lifetimes (up to 24 hours), and higher sampling frequencies (up to 10 kHz for ECG) and often provide for wireless connectivity [7]. Unfortunately, a mismatch exists between advances in sensor technology and the capabilities of state-of-the-art heartbeat analyzers [5]. They cannot usually keep up with the data acquisition rate, and are usually wall-plugged, thus preventing mobile monitoring. We aim at using state of the art commercial sensors from Ambu Inc. silver/silver chloride “Blue Sensor R” [7].

### 2.1 Application Specific Background

Our application is the 12-lead ECG, which uses nine sensors on the patient's body. With 3 of these sensors, physicians can use a method known as the 3-lead ECG, which suffers from the lack of information about some parts of the heart. By interconnecting the nine sensors for the 12-lead ECG we get twelve biomedical voltage signals; hence, produce huge amounts of data especially when used for a long number of hours. Physicians use the 12-lead ECG method, because it allows them to view the heart in its three dimensional form; thus, enabling detection of any abnormality that may not be apparent in the 3-lead technique. Figure 1 shows an example of a typical ECG signal, where the most important peaks are labeled: P, Q, R, S, T, and U. Each of these peaks and inter-peak distances is related to a heart activity that is of importance for analysis, and every combination of different inter-peak intervals proves a type of heart malfunction. The higher the sampling frequency the more accurate is the analysis since there are cases of diseases, where two peaks are too close (especially the R and T peaks in the case of R-on-T phenomenon) so that it becomes hard to detect the inter-peak distances and the heart period.

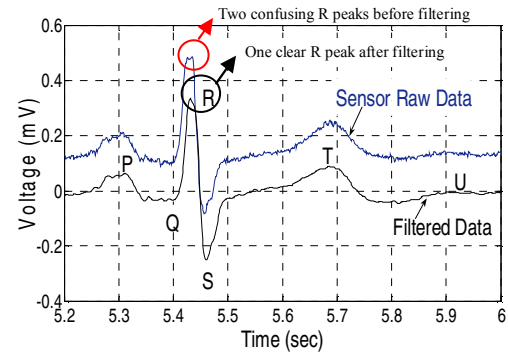
#### 2.1.1 Filtering

Data provided by biomedical sensors suffers from several types of noise: DC-offset, patient movements, and signal interference [8]. To

overcome all the problems related to sensor noise, we designed an IIR filter (implemented in hardware on a dedicated chip feeding an external SDRAM memory) that outputs its results in 16-bit binary format. Our IIR filter is of order 3, because it proves enough for our ECG analysis. Figure 1 shows an example of our filter results.

#### 2.1.2 Algorithm

Our proposed ECG-analysis algorithm is conceived to be parallel and hence scalable from the ground up. Since each lead senses and analyzes data independently, each lead can then be assigned to a different processor. So, to extend ECG analysis to 15-lead ECG for example or more, then what is required is to just change the number of processing elements in the system. The program reads a data file in chunks of four seconds. We discuss below the reason for the choice of the 4-second chunks. The data file mainly holds the values of the ECG at the lead in binary format. So by reading the data continuously every 4 seconds, we would be emulating a real sensor sending continuous data to an intermediate buffer that holds 4 seconds of data sampled at a certain frequency, typically 1000Hz. We used an autocorrelation function (ACF) based-methodology to calculate the period and other parameters of the heartbeat since it gives more accurate results than the conventional methods searching for the distance between two peaks. The autocorrelation we use as shown in (1) has a certain number of Lags (L) to minimize the computation for our specific application as discussed below. We validated our algorithm over several medical traces [9].



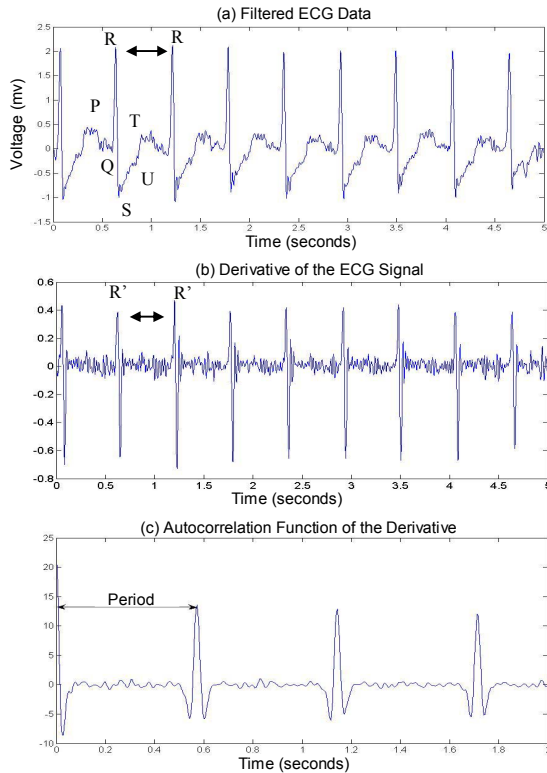
**Figure 1. ECG lead signal example: the upper curve is the Lead raw data and the lower curve is the filtered ECG lead.**

$$R_y[k] = \sum_{n=-\infty}^{n=\infty} y[n] \times y[n - k] \quad (1)$$

where,  $R_y$  is the autocorrelation function,  $y$  is the filtered signal under study,  $n$  is the index of the signal  $y$ , and  $k$  is the number of lags of the autocorrelation (L has an effect on the performance due to the high number of multiplications). We run the experiments for  $n = 1250$ , 5000 and 50,000 relative to the sampling frequencies of 250, 1000, and 10,000Hz, respectively. To run this algorithm with (1) it takes around 1.75 million multiplications. To minimize errors and execution time we use the derivative of the ECG filtered signal since if a function is periodic then its derivative is periodic. Hence the autocorrelation function of the derivative can give the period as shown in Figure 2. In order to be able to analyze ECG data in real-time and to be reactive in transmitting alarm signals to healthcare centers (in less than 1 minute), a minimum amount of acquired data has to be processed at a time without losing the validity of the results. For the heart beat period, we need at least 4 seconds of ECG data in order for the ACF to give correct results. From a technical viewpoint, real-time processing of ECG data would allow a finer-granularity analysis with respect to the traditional eyeball monitoring of the paper ECG readout.

## 2.2 Previous Work

ECG monitoring and analysis have been explored in many companies and research organizations. However, we are not aware of any single-chip real-time analysis solution for full 12-lead ECG, which is able to accurately study the heart rhythmic period and can diagnose all the peaks: P, Q, R, S, T and U and their inter-peak intervals to result in a disease diagnosis. Previous work on ECG analysis can be classified into 4 types of solutions: (i) classical stationary machine solutions [10], (ii) SoC solutions [11][12], (iii) Handheld device solutions [13] [14], and (iv) and ASIC solutions [15]. The classical solutions do not allow for patient mobility nor remote analyses since they are wall plugged, thus suffer from the need of many beds in the healthcare center. Moreover, in the classical medical technique for ECG analysis, the 12 lead signals are printed on eyeballing paper making the check of the different heart peaks and rhythms difficult and inaccurate due to its dependence on the physician's eyes. On the other hand, when using digital recording and filtering we can determine the peaks more accurately. The SoC solution in [11] does not run 12 lead analyses, but runs 1 lead per SoC. Consequently, to run 12-lead analyses with that solution means using 12 chips. One commercial solution [12] takes 8 input sensor-lines and calculates lead signals and analyzes them on the one DSP, hence it is time consuming. It only detects if the heart is healthy or unhealthy without analyzing diseases since it only detects the QRS without the P and T. Hence, it is not scalable. It uses 12 bits for the signals while we use 16 bits, thus, we add more accuracy to the analysis. The handheld solutions only read and transmit data. The ASIC solutions are just used for data acquisition before transmission.



**Figure 2. Heart period analysis: (a) ECG signal peaks P, Q, R, S, T, and U; (b) derivative amplifying the R peaks, that we label as R'; (c) autocorrelation of the derivative with clear significant periodic peaks.**

## 3. MPSOC ARCHITECTURE

In order to process filtered ECG data in real-time, we choose to deploy a parallel Multi-Processor System-on-Chip architecture. The

key point of these systems is to break up functions into parallel operations, thus speeding up execution and allowing individual cores to run at a lower frequency with respect to traditional monolithic processor cores. Technology today allows the integration of tens of cores onto the same silicon die, and we therefore designed a parallel system with up to 13 masters and 16 slaves (Figure 3). Since we are targeting a platform of practical interest, we choose advanced industrial components [16]. The processing elements are multi-issue VLIW DSP cores from STMicroelectronics, featuring 32kB instruction and data caches. These cores have 4 execution unit stages and rely on a highly optimized cross-compiler in order to exploit the parallelism. They leverage the flexibility of programmable cores and the computation efficiency of DSP cores. By the way, these features allow to reuse this platform for other biomedical applications other than the 12-lead ECG, thus making it cost-effective. Each processor core has its own private memory (512KB each), which is accessible through the bus, and can access an on-chip shared memory (8KB are enough for this application) for storing computation results. Other relevant slave components are a semaphore slave, implementing the test-and-set operation in hardware and used for synchronization purposes by the processors or for accessing critical sections, and an interrupt slave, which distributes interrupt signals to the processors. Interrupts to a certain processor are generated by writing to a specific location mapped to this slave core. The STBus interconnect from STMicroelectronics was instantiated as the system communication backbone. STBus can be instantiated both as a shared bus or as a partial or full crossbar, thus allowing efficient interconnect design and providing flexible support for design space exploration.

In our first implementation, we target a shared bus to reduce system complexity (see Figure 3) and assess whether application requirements can already be met or not with this configuration. We then explore also a crossbar-based system, which is sketched in Figure 4. The inherent increased parallelism exposed by a crossbar topology allows to decrease contention on shared communication resources, thus reducing overall execution time. In our implementation, only the instantiation of a 3x6 crossbar was interesting for the experiments. We put a private memory on each branch of the crossbar, which can be accessed by the associated processor core or by a DMA engine for off-chip to on-chip data transfers. Finally, we have a critical component for system performance which is the memory controller. It allows efficient access to the external 64MB SDRAM off-chip memory. A DMA engine is embedded in the memory controller tile, featuring multiple programming channels. The controller tile has two ports on the system interconnect, one slave port for control and one master port for data transfers. The overall controller is optimized to perform long DMA-driven data transfers and can reach the maximum speed of 600MB/s. Embedding the DMA engine in the controller has the additional benefit of minimizing overall bus traffic with respect to traditional standalone solutions. Our implementation is particularly suitable for I/O intensive applications such as the one we are targeting in this work.

In the above description, we have reported the worst case system configurations. In fact, fewer cores can be easily instantiated if needed. In contrast, this architectural template is very scalable and allows for future increase in the number of processors. This will allow to run in real time even more accurate ECG analyses for the highest sampling frequency available in sensors (10,000Hz, and 15 leads, for instance). The entire system has been simulated by means of the MPSIM simulation environment [16], which provides for cycle-accurate functional simulation of complete MPSoCs at a simulation speed of 200Kcycles/second (on average), running on a P4@3.5GHz. The simulator provides also a power characterization framework leveraging 0.13 $\mu$ m technology-homogeneous industrial power models from STMicroelectronics [17][18]. We believe that for life-critical applications, low-level accurate simulation is worth doing, although potentially slow, in order to perfectly understand system level

behaviour and have a predictable system with minimum degrees of uncertainty. Each processor core programs the DMA engine to periodically transfer input data chunks onto their private on-chip memories. Moved data corresponds to 5 seconds of data acquisition at the sensors: 10kB at 1000Hz sampling frequency, transferred on average in 319279 clock cycles (DMA programming plus actual data transfer) on a shared bus with 12 processors. The consumed bus bandwidth is about 6Mbyte/sec, which is negligible for an STBus interconnect, whose maximum theoretical bandwidth with 1 wait state memories exceeds 400Mbyte/sec. Then each processor performs computation independently, and accesses its own private memory for cache line refills. Different solutions can be explored, such as processing more leads onto the same processor, thus impacting the final execution time. Output data, amounting to 64 byte, are written to the on-chip shared memory, but their contribution to the consumed bus bandwidth is negligible. In principle, when the shared memory is filled beyond a certain level, its content can be swapped by the DMA engine to the off-chip SDRAM, where the history of 8 hours of computation can be stored. Data can also be remotely transmitted via a telemedicine link.

#### 4. EXPERIMENT DESCRIPTION

We ran experiments in order to check the limits to respect the time figure of merit since our MPSoC is a real-time application based system. So we ran experiments to check the performance of each system design with increasing frequencies (up to 10KHz). We also ran experiments to look for optimizing the algorithm together with the design by changing some algorithm parameters and looking into the overall performance of the specific biomedical application on each MPSoC design. We also ran experiments by distributing the application on different numbers of DSP cores for each design (shared bus, crossbar, and partial crossbar). The results of these experiments are presented in Section 5 below.

#### 5. RESULTS

As a first exploration, we have compared the performances of an ARM7TDMI with the ST220 DSP, in order to verify the performances of the chosen VLIW with respect to the computation kernel of our specific application. In order to have a safe comparison, we set similar dimensions of the cache memory (32KB) for the two solutions, and we run two simulations for the processing of one ECG-Lead at 250Hz sampling frequency. We run a performance-comparison between two application-specific cores. We adopt this one core solution, because our first aim is to investigate the computation efficiency of the two cores for our specific biomedical application, and de-emphasize system level interaction effects such as synchronization mismatches or contention latency for bus access. Hence, the performance of the ARM7 core serves as a reference to assess the computation efficiency of the VLIW DSP core for the same specific application. In Figure 5, we can observe that the LX220 DSP results in a better behavior in both: execution time and energy consumption. In detail, the ARM core is 9 times slower than the ST220 in terms of execution time, and it consumes more than twice the energy incurred by the DSP. These results can be explained based on three considerations:

- (i) The ST220 has better software development tools, which result in a smaller executable code. The size of the executable code for the ARM is 1.7 times larger than that of the ST220
- (ii) The ST220 is a VLIW DSP core, therefore it is able to theoretically achieve the maximum performance of 4 instructions per cycle (i.e., 1 bundle)
- (iii) A metric which is related to both previous considerations is the static instructions-per-cycle, which depends on the compiler efficiency and on the multi-pipeline execution path of the

ST220. For our application, this metric turns out to be 2.9 instructions-per-bundle.

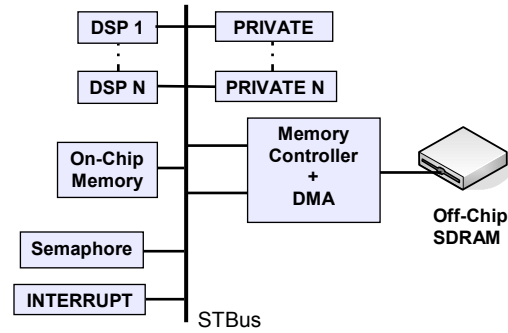


Figure 3. Single bus architecture with STBus interconnect.

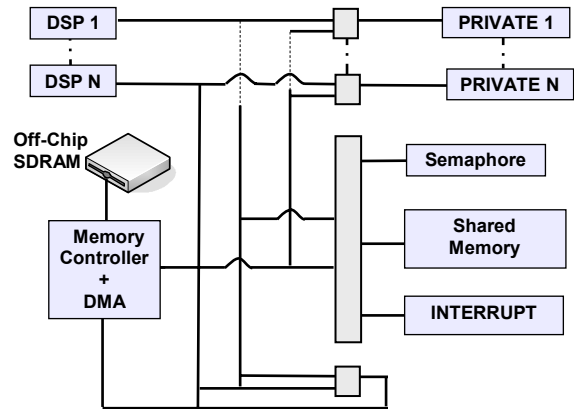


Figure 4. Full Crossbar architecture with STBus interconnect .

Let us therefore select the best processor core for our computation kernel, from the performance and energy viewpoints. We now want to optimally configure the system to satisfy the application requirements at the minimum hardware cost. We therefore measure the execution time and the energy dissipation for an increasing number of DSP cores in order to find the optimal configuration of the system. Since commercially available ECG solutions target sampling frequencies ranging from 250 to 1000 Hz, we performed the exploration for these two extreme cases for the 12-lead ECG signal. We analyze a chunk of 4secs of input data, which provides a reasonable margin for safe detection of heartbeat disorders. Note that the computation workload for the processor cores increases in a quadratic manner with increasing sampling-frequency (due to the specific application algorithm).

Figure 6 shows that if we increase the number of processors, the execution time scales linearly, which proves that second order effects typical of multi-processor systems (e.g., bus contention reducing the offered bandwidth to the processor cores with respect to the requested one) has only negligible effects on system performance, proving that the system is well configured and a single shared bus communication architecture is well suited for this application. However, this does not mean that the amount of data moved across the bus is negligible: around 100KB (at 1000Hz). This data is, however, read by the processor core throughout the entire execution time, thus absorbing only a small portion of the bus bandwidth. In this regime, the bus performance is still additive.



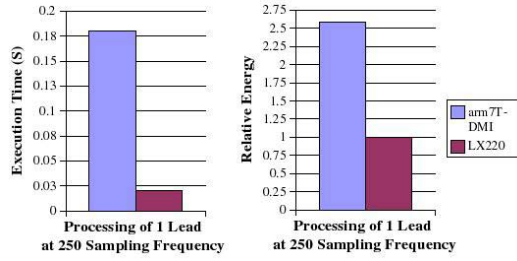


Figure 5. Comparing ARM7TDMI with a ST200 DSP performances, in processing 1 Lead at 250Hz sampling frequency.

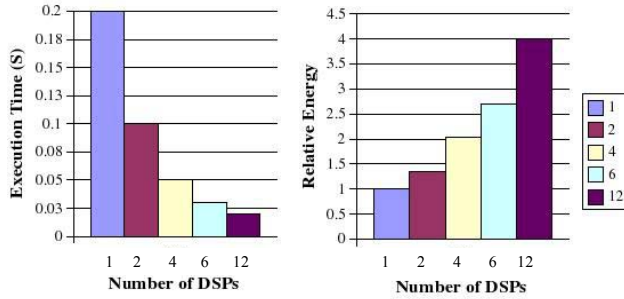


Figure 6. Execution Time and relative energy of shared bus at 250Hz sampling frequency.

Moreover, the perfect scalability of the application is also due to memory controller performance. In fact, at the beginning of the computation each processor loads processing data from the off-chip to the on-chip memory, hence, requiring peak memory controller bandwidth. The architecture of the memory controller proves capable of providing the required bandwidth in an additive fashion. By looking at the 1000Hz plot (Figure 7), we observe that 1 DSP is able to process an ECG-lead in slightly more than 3 seconds. Therefore, we still have about 1 second left (before the 4secs deadline), which is enough to perform additional analysis of the results of the individual lead-computations and converge to a decision about the heart period and malfunctions. Looking forward, we try to understand how our solution situates itself with respect to the demand for higher sampling frequencies raised by the need to perform higher accuracy analysis and the evolution of state-of-the-art sensor technology. We, therefore, measure and plot the maximum sampling frequency at which our MPSoC solution can be operated while still meeting real-time requirements. This frequency translates to poor scalability. The reason for this is mainly the interconnect performance, which does not scale any more. In fact, *bus busy* (the number of bus busy cycles over the total execution time) at the critical frequency of 2200Hz is almost 100% (99.95%), i.e., the bus is fully saturated. This is due to the fact that the amount of data being transferred across the bus increases linearly with the sampling frequency. In order to make the platform performance more scalable, we revert to a full-crossbar solution for the communication architecture. The benefits are clearly observed Figure 8, where the maximum analyzable frequency (with respect to real-time constraints) amounts now to 4000Hz, i.e. nearly twice as much as the performance with a shared bus. Moreover, we observe that average bus transaction latencies at the critical frequency are still very close to the minimum latencies, thus indicating that the crossbar is very lightly loaded. Another informative metric is the bus efficiency (number of cycles during which the bus transfers useful data over the *bus busy* cycles), which amounts to 71.83%.

We simulate the 12-processor system to get the upper bound on system performance and push the architecture to the limit. For the same reason, we restrict the analysis period to 3.5secs, which is the minimum value of the input data-chunk time-span derived from the biomedical algorithm. The results presented in Figure 8 show that

with shared-bus architecture, the maximum sampling frequency that the MPSoC can handle without going beyond the real-time constraint is only around 2200Hz.

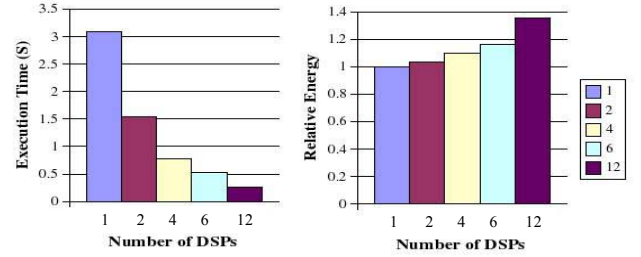


Figure 7. Execution Time and relative energy of shared bus at 1000Hz sampling frequency.

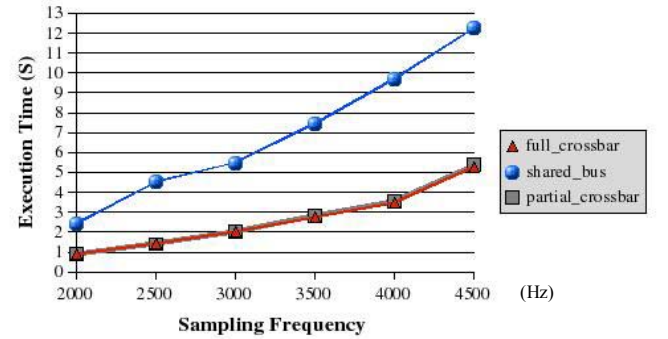


Figure 8. Critical sampling Frequencies for 3 architectures: (1) shared bus, (2) full crossbar, and (3) partial crossbar.

This good performance is an effect of the lack of contention on the crossbar branches, which is in turn due to the high performance of the memory controller and to the matching of the application traffic patterns with the underlying parallel communication architecture. As a consequence, with a full crossbar the system performance is no more interconnect-limited but computation-limited. Since the computation workload of the system grows in a quadratic manner with the sampling frequency, it rapidly increases task execution times and reduces the available slack time with respect to the deadline. We observe that the performance with a partial crossbar closely matches that of a full-crossbar (less than 2% average difference) but with almost 3 times less hardware resources. We found the optimal partial crossbar configuration (5x5 instead of a 13x13) by accurate characterization of shared bus performance. On a shared bus, we increased the number of processors and observed when the execution times started deviating as an effect of bus contention. With up to 4 cores connected to the same shared communication resource, this latter is able to work in an additive regime. Although the architecture cannot work in real-time at more than 4000Hz, we wanted to measure the execution time under non-real-time computation. In fact, from the execution time to process 3.5secs of input data, we can derive the amount of buffering that is required to store incoming data from the lead. By knowing the overall capacity of the off-chip memory, we then derive the maximum analysis time that we can afford at such high frequencies. Results are shown in Figure 9. Let us focus on the shared bus case, with 10KHz. The execution time for analyzing 3.5secs is a bit less than 1 minute (around 57secs), which is 16 times more than the real-time constraint. As a consequence, we can still decide to perform this kind of analysis, but we need to buffer 16 input data chunks while we are processing one chunk. Since an off-chip SDRAM memory can be 512MB, we can perform 3.5 minutes analysis before saturating the memory. With a full crossbar, this time amounts to around 14 minutes.

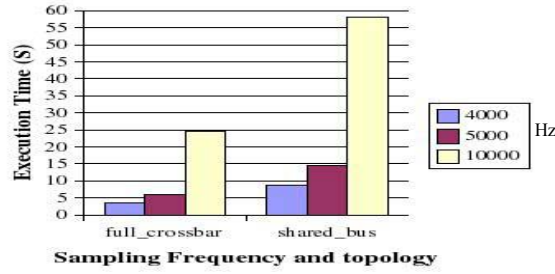


Figure 9. Analysis-time investigation with increasing frequencies (up to 10KHz) for the ECG application.

## 6. SOLUTIONS COMPARISON

Comparing our application-based MPSoC designs, we can choose the best architecture relative to the biomedical purpose. Hence, for a solution that competes with and performs better than the existing commercial solutions [12] (input sampling frequency from 250 to 1000Hz), we adopt our shared bus system architecture (Figure 3) since its advantages over existing solutions are:

- ECG SoC available solution nearest to ours is [11], but our design performs better on analysis time-consumption. Our solution is easier to deploy since the 12 leads are input to one SoC instead of having 12 SoCs (i.e. cheaper to scale with increasing leads)
- In our designs, we can do full 12-lead ECG analysis at relatively high frequencies. Our design is optimal especially that we can offer the choice of the SoC architecture (shared bus, crossbar, or partial crossbar) based on the biomedical need of a frequency range.

Table 1 shows the advantage of our three designs to the best available designs we are aware of in the research and in the market.

Table 1. Comparison between ECG analysis research SoC solution [11], available SoC commercial solution [12], and our three MPSoC designs: ST-PCB is partial crossbar, ST-CB is Full Crossbar, and ST-SB is the Shared Bus solution. Fs is the sampling frequency.

Solution	Data bits	Memory	Fs (Hz)	Analysis Time	SoC inputs	Pre-filter	Application results
ST-PCB	16	Off chip = 512MB 32kB I-cache/DSP 32kB D-cache/DSP	4000	< 3.5s	12	IIR	Full 12-LEAD: heart-period; P,Q,R,S,T,U peaks;& detect disease
ST-CB	16	Same as above	4000	< 3.5s	12	IIR	Same as above
ST-SB	16	Same as above	2200	< 4s	12	IIR	Same as above
[11]	10	8KB Cache	250	No study	1	Notch	Only QRS, only decide if healthy or unhealthy
[12]	12	No info	800	No study	8	IIR	Only QRS, decide if healthy or unhealthy

## 7. CONCLUSION

We present an application-specific MPSoC architecture for real-time ECG analysis. Our solution leverages the computation horsepower of many (up to 12) concurrent DSP cores to process ECG data. This solution paves the way for novel healthcare delivery scenarios (e.g., mobility) and for accurate diagnosis of heart-related diseases. We describe the design methodology for the MPSoC and explore the configuration space looking for the most effective solution, performance and energy-wise. We present

three interconnect architectures (Single Bus, Full Crossbar, and Partial Crossbar) and compare them with existing solutions. The sampling frequencies of 2200Hz and 4000Hz, with 12 DSPs are found to be the critical points for our Shared-Bus design and Crossbar architecture, respectively.

## 8. REFERENCES

- [1] Fuster, V. Epidemic of Cardiovascular Disease and Stroke: The Three Main Challenges, *Circulation*, Vol. 99, Issue 9, March 1999, 1132-1137.
- [2] Lo, B., Thiemjarus, S., King, R., and Yang, G. Body Sensor Network- A Wireless Sensor Platform for Pervasive Healthcare Monitoring, *Adjunct Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE'05)*, May 2005, 77-80.
- [3] Code Blue, Medical Wireless Sensor Networks, <http://www.eecs.harvard.edu/~mdw/proj/codeblue>
- [4] BIOPAC Systems Inc., <http://biopac.com/>
- [5] Harland, C., Clark, T., and Prance, R. High resolution ambulatory electrocardiographic monitoring using wrist-mounted electric potential sensors, *Measurement Science and Technology*, Vol. 14, 2003, 923-928.
- [6] Heart and Stroke Foundation of Canada, The Changing Face of Heart Disease and Stroke in Canada 2000, Annual report, 1999.
- [7] Ambu, Inc. biomedical devices company, [www.ambuusa.com](http://www.ambuusa.com)
- [8] Company-Bosch, E., Hartmann, E. ECG Front-End Design is Simplified with MicroConverter, *Journal of Analog Dialogue*, Vol. 37, November 2003.
- [9] PhysioBank, Physiologic signal archives, for biomedical research, <http://www.physionet.org/physiobank/database/ptbdb/>
- [10] BIOPAC systems Inc., <http://biopac.com/>
- [11] Chang, M., Lin, Z., Chang, C., Chan, H., and Feng, W. Design of a System-on-Chip for ECG signal processing, *The 2004 IEEE Asia-Pacific Conference on Circuits and Systems*, December 6-9, 2004.
- [12] Freescale™ semiconductor, Personal Electrocardiogram (ECG) Monitor, <http://www.freescale.com/>
- [13] Hung, K., Zhang, Y. T., and Tai, B. Wearable Medical Devices for Tele-Home Healthcare, *In Proceedings of the 26th Annual International Conference on the IEEE EMBS*, San Francisco, CA, USA, September 1-5, 2004.
- [14] Jun, D., and Hong-Hai, Z., Mobile ECG detector through GPRS/Internet, *In Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems (CBMS'04)*, 2004.
- [15] Desel, T., Reichel, T., Rudischhauser, S., and Hauer, H. A CMOS Nine Channel ECG Measurement IC, *2nd International Conference on ASIC*, 1996, Oct. 1996, 115-118.
- [16] Loghi, M., Angiolini, F., Bertozzi, D., Benini, L., and Zafalon, R. Analyzing On-Chip Communication in a {MPSoC} Environment, *In Proceedings of Design and Test in Europe Conference (DATE)*, February 2004, 752-757.
- [17] Loghi, M., Poncino, M., and Benini, L. Cycle-Accurate Power Analysis for Multiprocessor Systems-on-a-Chip, *GLSVLSI04: Great Lake Symposium on VLSI*, April 2004, 401-406.
- [18] Bona, A., Zaccaria, V., and Zafalon, R. System level power modeling and simulation of high-end industrial network-on-chip", *In Proceedings of Design and Test in Europe Conference (DATE)*, February 2004, 318-323.