Towards a Minimal Architecture for a Printable, Modular, and Robust Sensing Skin

Austin Buchan  
Jonathan Bachrach  
Ronald S. Fearing

Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2014-31  
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-31.html

April 27, 2014
**Title:** Towards a Minimal Architecture for a Printable, Modular, and Robust Sensing Skin

**Author:**
University of California at Berkeley, Electrical Engineering and Computer Sciences, Berkeley, CA, 94720

**DISTRIBUTION/AVAILABILITY STATEMENT:**
Approved for public release; distribution unlimited

**ABSTRACT:**

**SUBJECT TERMS:**

**SECURITY CLASSIFICATION OF:**

<table>
<thead>
<tr>
<th>16a. REPORT</th>
<th>16b. ABSTRACT</th>
<th>16c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same as Report (SAR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98) Preceded by ANSI Std Z39-18
Copyright © 2014, by the author(s).
All rights reserved.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission.
Towards a Minimal Architecture for a Printable, Modular, and Robust Sensing Skin

A. Buchan, J. Bachrach, and R.S. Fearing

I. INTRODUCTION

The ability to sense and react to the environment is what separates robotic systems from simple machines. Surface interaction sensing is a vital component for robots to adapt to unstructured human and natural environments [1]. However, there is not yet to see a convenient method for covering curved and complex surfaces common to today’s research robots in tactile sensors. The problem of gathering this information from transducers distributed on the surface of a robot is key to a viable sensing solution.

Here we present an adaptation of sensitive robotic skin to the domain of rapidly prototyped systems, with the long-term goal of integrating sensing and networking with the manufacturing process using low-cost printed flexible-substrate semiconductors. In addition, this skin will inform the robot as to the shape and arrangement of the sensors on the surface, as well as recovering from and reporting changes in this topology if damaged. By exploring this approach to whole-body multi-modal sensing, we expect to improve the ability to localize and navigate in unstructured environments via detection of obstacles, targets, and environmental conditions. The shape sensing feature will allow the robots to modify their behavior and continue functioning as long as possible despite destructive modifications of the robot. For example, a legged robot with such a skin integrated throughout its body will be able to detect a leg loss, and modify its gait accordingly to continue locomotion. As a first step, we consider how the design constraints of such a sensing skin motivate the design parameters of a sensor network, and explore a particular network architecture that achieves the desired data integrity with simple underlying logic. Ultimately we want to answer the question: what are the minimum necessary conditions for constructing a serviceable, highly configurable sensor skin that is compatible with low-performance printed electronics?

This work was supported by the United States Defense Advanced Research Projects Agency under the Maximum Mobility and Manipulation (M3) program, contract W911NF1110089. [abuchan,jrb,ronf] at eecs.berkeley.edu Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA

II. PRIOR WORK

A. Robotic Sensing Skins

Prior work on robotic sensing skins include exploiting bulk resistive skin properties [2], using row-column addressing schemes to multiplex the readout of sensors [3] [4] [5], or using local processors to compress data before transmission [6]. Still others do away entirely with conventional wires to use optical [7], radio frequency [8], or ad-hoc communication methods [9]. Also considered are Tapped Delay Lines (TDLs), which use passive time delay components to time sequence sensor output on a single analog
line, and standard addressable bus communication networks such as I2C. Finally, recent work [10] in modular sensing units uses a Cellular Finite State Automata (CFSA) model of computation and data transmission in a redundantly wired surface area network.

B. Flexible Electronics

While the idea of incorporating electronics into flexible substrates has been studied for quite some time [11], several exciting recent advances in flexible semiconductor technology show promise for integration into SCM for use in a sensing skin architecture. The primary approaches include using thin silicon islands [12] with flexible interconnects [13], organic polymers [14], and Single Wall Carbon Nanotubes (SWNT) [15]. These devices can be implemented on a variety of flexible and stretchable substrates including polyimide, paper [16], silk [17], and silicone [18]. Already there are examples of medium scale integration of these devices into useful circuits [19], including work on a Carbon Nanotube ADC [20]. Several sensing modalities have been shown with flexible semiconductors including tactile [21], thermal [22], optical, and chemical [23]. Vital to our application is the ability of these semiconductors to be constructed with rapid prototyping manufacturing processes such as screen and inkjet printing [24]. We expect that continued advances in flextronics will allow implementation of sensing network logic with our SCM process in the near future.

III. Design Considerations

We have rapidly prototyped low-cost, biologically-inspired mobile autonomous systems [25]. Figure 2 shows the Smart Composite Manufacturing (SCM) process [26] pioneered for these robots. Patterns for rigid and flexible sections of the body are laser-cut, laminated, and folded into functional kinematic forms. This enables radically different mechanical designs to be realized in a matter of hours, and at a cost which can be orders of magnitude lower than that of comparable systems.

Through collaboration with flexible-substrate semiconductor research groups [21], we are exploring the integration of sensing electronics into the mechanical substrate of the SCM process such that the physical configuration of the composite sheet determines the sensor network behavior, and allows seamless integration from the start of the design process. As flexible semiconductors are an active research area, the yield of such processes is expected to be low for initial work. This fact drives the motivation to keep the electronics for printed sensing skin as simple as possible in terms of logic and interconnection wires.

We estimate that covering the lateral area of a 10cm by 4cm robot in 100 sensors with 8 bits of resolution and a full skin update rate of 10Hz will be sufficient to make headway on basic navigation applications (wall following, obstacle avoidance). A final design should have sensing units < 40mm², with an update rate of 10Hz for the entire skin. The skin should weigh no more than 10g (100mg/sensor), consume a total of < 100mW (1mW/sensor), and use as few connection wires as possible. For detection of damage, and ability to route over flexures, the minimum dimension of a sensing element should be < 6mm.

IV. Approach

The Ad-Hoc network approach with diffusive coupling [9] shows good resiliency to individual node failure, but does not offer the ability to extract topology information from neighbor connections. The fact that each node must be individually addressed also complicates initial configuration, and storage of the hop list limits the total size of the network. The HEX-O-SKIN cell network design [10] shows the greatest promise for implementing a modular sensing grid. The ability to reroute data around damaged cells discussed in the HEX-O-SKIN design after the detection of failure will be key to the design explored in this work. The complexity, weight, size, and power consumption of the HEX-O-SKIN cells prohibit a direct implementation for millirobots, but a similar design is used for proof-of-concept in this paper. The primary augmentation of the HEX-O-SKIN framework is the addition of a topology phase, which allows the actual connectivity of a grid to be reconstructed with minimal connections.

A. Sensor Grid and Cell Architecture

The top-level organization of the sensing skin is a regular, tessellating arrangement of identical sensing cells. Different shapes can impart different properties to the grid. The basic case of a triangular tessellation will have the fewest bits of state required to store edge information, but has the lowest edge redundancy factor. In contrast, a hexagonal cell grid has the highest redundancy and cell-packing density, but as such will have more idle logic for average arrays on surfaces. Each sensor cell unit consists of a computational logic core connected to global power bus, with bidirectional communication links to each of its neighbors in the grid. A controller needs only to connect to power and the signal line of any cell. Multiple uplinks between the grid and a supervisory controller are allowed since the choice of a parent isolates a cell to a particular master.

Figure 3 shows the organization of the major components in the individual cell architecture. A simple way to guarantee
that a cell does not participate in communication if it’s perimeter is breached by a cut is by having the trace that supplies power to the cell encircle it. The FSM logic controls the operation of the cell based on the current state and edge communication, and monitors the time between transactions to detect timeout errors. The edge I/O logic is responsible for serially communicating data between edges. For the analysis below, data is assumed to be transmitted in atomic symbol units of fixed bit-width. Error detection circuitry is necessary to guarantee the correct operation of the grid, since cells depend on accurate signaling of end of transmission from other cells. Cyclic Redundancy Checks (CRCs) are a simple and logic-efficient manner to implement this feature. The design for the cellular architecture is agnostic to the type of sensor used, as long as the sensor data is a fixed bit width and can be clocked out on demand. Multimodal sensing can be achieved by putting multiple sensors in each cell and serializing the data during the sensor readout phase, or by having a regular arrangement of sensor types in the sensor grid pattern, and inferring the sensor type from the topology information. This work simulates and implements a proof of concept based on local synchrony within a cell, but could easily be extended to a fully asynchronous logic system given the appropriate synthesis toolchain.

B. Finite State Machine

Each cell implements an identical CSFA that maps its current state and inputs to the next state and outputs. Figure 4 shows a high-level description of the state transition model. Extracting data from a sensor grid progresses in three main phases: wavefront, topology, and sensor. The wavefront stage constructs a spanning tree on the sensor grid such that each cell chooses one of its neighbors as a parent, such that forwarding data along these links will eventually reach a master uplink. The topology phase serializes this connectivity information and reports it to the master, such that the topology of the grid can be determined. Finally, the sensor phase serializes the sensor data from each cell using the same spanning tree in a pipelined manner to maximize sensor data throughput. The state transitions are designed so that if a cell fails during a data forwarding phase, the cells behind the failure will converge to stable start state at the beginning of the phase within a deterministic time. This allows the grid to be re-initialized in a synchronized manner after a failure is detected.

The wavefront phase is a pattern of activation emanating from master uplinks that provides information to each cell about the direction to an uplink. A master initiates the wavefront on a cell grid by injecting a Start token. A cell chooses the first edge receiving a Start token as its parent. After receiving the token, a cell will acknowledge its chosen parent while forwarding the token to all of its neighbors that were not part of the wavefront in the previous cycle. This allows the parent cell to recognize which of its edges have active children. The master must wait long enough for the worst case of $N_{\text{cells}}$ transaction periods before initializing the Topology phase by injecting another Start token.

The topology phase flattens the network spanning tree represented by the parent links to a symbol string that can be used to reconstruct the tree. If a cell has remaining children, it emits the direction of the next child while sending that child a Start symbol, otherwise a Stop symbol is emitted. The cell will then forward the stream of symbols from the activated child to its parent, with the exception of replacing a Stop symbol with a Next symbol. This ensures that a Stop symbol is only emitted when a cell’s full stream terminates. This process is repeated until no children remain.

Serializing sensor information from the grid proceeds similarly to the topology step. First a cell sends its sensor data to its parent, while propagating the Start token to its first child. It then forwards the child stream of data until a Stop token is encountered. The cell discards the Stop token and starts forwarding from the next child until none remain, terminating its own stream with a Stop symbol. The stream of data from the sensor phase represents an in-order traversal of the network spanning tree. Since the grid is assumed to be regular, a simple recursive parser can determine the location in the grid of each sensor reading.

A cell will enter an error state if it reads an Error symbol, encounters corrupted data, or times out while waiting on a response from a child. Before returning to the initialization state, a cell will propagate the Error to all edges, guaranteeing that the entire grid will be re-initialized. If a cell encounters a failure rather than reading an error symbol, it will mark that edge as failed and not include it in subsequent wavefront initializations. This allows the grid to isolate failed
nodes given enough re-initializations.

C. Design Parameterization

Figure I shows how design parameter selection affects the total state required in the cell. \( N_{\text{edge}} \) is the number of cell neighbors in a particular grid implementation. \( N_{\text{sense}} \) is the number of bits of sensor information per cell. \( N_{\text{check}} \) is the number of bits in the checksum. Tradeoffs in the design can be made to favor a minimal state logic or higher likelihood to detect failures. Using the absolute minimum state for tokens and checksum information results in high likelihood that a failing cell will incorrectly pass a checksum. To counteract this larger checksums can be chosen, which will increase the total logic complexity and reduce sensor throughput. The final selection can be made to balance these effects given a specific application. Sensor throughput is calculated (with provision for start and stop bits) as

\[
\text{Throughput} = \frac{N_{\text{sense}}}{N_{\text{sense}} + 2 + N_{\text{check}}}
\]  

(1)

The likelihood of a false positive continuation due a randomly failing neighbor cell will be

\[
P(\text{FalsePositive}) = \frac{1}{2^{N_{\text{check}}}}
\]  

(2)

V. RESULTS

A. Algorithm Simulation

In order to estimate the complexity of implementing the state machine architecture using flexible-substrate semiconductors, we implemented the cell state machine in Register Transfer Logic (RTL) written in the Chisel hardware construction language [27]. Chisel produces both a high speed C++ based cycle-accurate software simulation and a low-level Verilog hardware description that maps to a standard ASIC flow for synthesis. After encoding the semantics of the state machine as described above, the simulation was used to verify that the architecture behaved as expected up to a 4 by 4 grid of midrange cells.

B. Register Transfer Logic Synthesis

We used the Synopsis Design Compiler to map the Chisel generated Verilog to the Taiwan Semiconductor Manufacturing Company (TSMC)’s 65-nm GPLUSTC CMOS standard cells. Table II shows the number of gates (standard cells) and flip-flops generated for the given number of edges. We can see that the synthesis state machine is well below the complexity of a full processor, which should allow much higher cell yield with a given flextronic process. By way of comparison, we mapped an 8-bit 6502 processor without memory to the GPLUSTC cells, which required 1479 gates and 145 flip-flops. In order to run the state machine, the device utilization would further increase since memory would be required for both instruction and scratch pad storage leading to many more cells. While these preliminary numbers reflect the performance of highly optimized silicon processes, the results verify that the desired state machine functionality can be achieved with very few bits of state, and thus total device elements, compared to a microcontroller implementation at each node.

C. Microcontroller Implementation

To simulate the minimal cell FSM, we constructed sensor cells with an 8-bit PIC16F1503 microcontroller, debugging LEDs, optical sensor, and 6 edges for connectivity of power and inter-cell communications. Based on the success of Hellard [28], Ohmura [29] and Mittendenfer’s [10] designs using the Sharp GP2S60 IR transceiver pair, it was chosen as a sensor to allow tactile and optical sensing. Thermal sensing can be achieved with the microcontroller’s internal thermal sensor. Figure 5 shows several approaches used to prototype the microcontroller based implementation of the sensor grid. The hexagonal array of cells on the PCB substrate with DIP switches is used to exhaustively test the hardware implementation cells for correct recovery from disrupting communication on edges. The operational parameters of these cells are detailed in Table III.

Figure 6(a) shows the construction of a proof-of-concept flexible sensing skin. The flexure layer was simulated using a copper and Kapton laminate which connects the edges of the separated PCM sensor cells. After laminating the 4 by 7 cell flexure layer between adhesive-backed cardboard, the composite was cut to a shape simulating a four bar linkage common in the construction of biologically inspired millirobots. Figure 6(b) shows a sequence of several features of the sensing architecture. First the uplink cell location is switched, and after re-initializing, the controller reads the same topology. Next, a partial cut is made on an edge that was included in the original network. As expected, the skin reports an error, and the controlling software re-initializes the grid. Afterwards the software recovers the same shape, but displays an updated spanning tree grid to reach the cells past the cut. Finally, a cut is made on an edge that severs cells from the end of the grid. After this reset, the updated shape of the grid is

\[\text{Table I: Theoretical performance of selected designs}\]

<table>
<thead>
<tr>
<th></th>
<th>Minimal</th>
<th>Midrange</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_{\text{edge}})</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>(N_{\text{sense}})</td>
<td>1</td>
<td>8</td>
<td>128</td>
</tr>
<tr>
<td>(N_{\text{check}})</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Throughput</td>
<td>20%</td>
<td>47%</td>
<td>88%</td>
</tr>
<tr>
<td>(P(\text{False Pos.}))</td>
<td>0.25</td>
<td>3.9 \times 10^{-3}</td>
<td>1.5 \times 10^{-5}</td>
</tr>
</tbody>
</table>

\[\text{Table II: Performance of synthesized designs in 65nm Si Process}\]

<table>
<thead>
<tr>
<th></th>
<th>Minimal</th>
<th>Midrange</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy (flip-flops)</td>
<td>25</td>
<td>29</td>
<td>35</td>
</tr>
<tr>
<td>Gates</td>
<td>256</td>
<td>423</td>
<td>626</td>
</tr>
<tr>
<td>Area ((\mu\text{m}^2))</td>
<td>752</td>
<td>1009</td>
<td>1628</td>
</tr>
<tr>
<td>Power ((\mu\text{W}))</td>
<td>2.25</td>
<td>3.16</td>
<td>4.28</td>
</tr>
</tbody>
</table>

\[1\text{Arlet, 6502 Verilog Model, http://ladybug.xs4all.nl/arlet/fpga/6502/}\]
displayed while continuing to read sensor information from the remaining cells.

VI. CONCLUSIONS AND FUTURE WORK

Table III compares the features of modular sensor cell implementations, with a projection based on state of the art flexible electronic technologies. Estimated HEX-O-SKIN cell cost is based on off-the-shelf component costs at quantities of >1000 pieces. The projection to flexible electronics is based on the applying the flip-flop and gate counts for the midrange design synthesis to SWNT printable electronic techniques [15]. Complimentary logic using 18 transistors per flip-flop and 4 transistors per gate is assumed. Device parameters are $f=10$kHz, $L=20\mu$m, $W=50\mu$m, $V=10$V, activity factor $\alpha=0.5$, and a total area of $5000\mu$m$^2$ per transistor including wiring. This conservative estimate for total area per transistor also allows for ADC circuitry. Power per device is calculated with standard formulas for switching and quiescent draw: $P_{\text{tot}} = \alpha f CV^2 + V I_{\text{off}}$. The cost for the flextronics projection is based on the $0.17$/in$^2$ value quoted in [24]. Since the primary material cost is conductor wiring, this number should be valid for both organic and SWNT electronics. The weight assumes a $25.4\mu$m PET substrate as the main contribution.

The comparison shows that a flextronics implementation with current technology could achieve the desired capabilities with reasonable cost, weight, data throughput, and sensor density. Meeting the power consumption constraint for our millirobots will require further progress in flextronics performance. The microcontroller implementation from this work would also require significant optimization in the data rate for use in an application, but successfully shows the functionality and robustness of the algorithm with a physical, flexible system.

The CFSA architecture for a surface-area sensor network described here accomplishes the functional goals of robust sensor data and topology information extraction. The design is uniquely suited to integration with the SCM process for constructing mobile robots using printable flexible electronics due to the very low complexity of digital logic required by the sensing cell units. Future work will include a full design space exploration for the logic implementation, including asynchronous models of inter-cell communication. Implementation of the logic on an FPGA will serve to characterize high-speed operation, but is unlikely to be used on our robots due to their large size, weight, and power consumption. Further, techniques will be analyzed for mitigating disruptive cell failure due to device non-idealities.

VII. ACKNOWLEDGMENTS

Thanks Chuan Wang and Joe Corea for input on flextronics. Thanks to Colin Scott for guidance provided on network failure analysis, as well as Yunsup Lee and Huy Vo for help with Chisel synthesis and analysis.

REFERENCES


