22nd IEEE International Symposium on Intelligent Control
Part of IEEE Multi-conference on Systems and Control
Singapore, 1-3 October 2007

Dynamic Data Aggregation in Wireless Sensor Networks

S. Commuri and V. Tadigotla

Abstract—.**Performance improvement in Wireless Sensor Networks (WSNs) through the use of reconfigurable cluster heads is addressed in this paper. The performance of a WSN is limited by the scarce on-board power in each sensor node. Since the sensed data in the WSN is transmitted to a sink from node to node in a multi-hop fashion, elimination of the redundancies in the data and aggregation of the data from multiple sources can improve the throughput and lifetime of the network. This, in practice, is infeasible because the aggregation to be performed depends on the requirements of the end user/application and is either unknown at the time of deployment or changes over time. In this paper, the problem of implementing dynamic data aggregation in WSNs is addressed through the design of reconfigurable cluster heads (RCHs) using Field Programmable Gate Arrays (FPGAs). Our results demonstrate that different data aggregation algorithms can be efficiently implemented on the RCHs in run-time. Such an implementation provides the necessary flexibility demanded by applications, while resulting in significant reduction in the query processing time and the overall power consumption in the network.**

I. Introduction

A Wireless Sensor Network (WSN) is a collection of computational nodes, each equipped with sensing devices and radio transceivers. These sensing nodes form an ad hoc network that can be used in a variety of applications like target classification and tracking, health monitoring, and in environmental and ecological monitoring. While the applications based on WSNs appear promising, their potential is hampered by their economies of scale, available power, ability to self-organize, and the extraction of information from the several thousand sensory measurements. The limited power available at each node restricts the amount of data that can be transmitted between nodes in a sensor network. This problem is further complicated as the size of the network increases. Since the available energy restricts the amount of information that can be transferred from end-to-end in a WSN, several researchers have investigated the use of data aggregation techniques to increase the throughput and the lifetime of the network [1], [2].

Data aggregation is a technique of collecting raw data from sensor nodes, eliminating redundant measurements, and extracting the information content for onward transmission. Cluster-based and tree-based protocols were proposed in the literature to support data aggregation in WSNs. In the cluster-based approach, a group of sensors form a cluster and communicate information only to a specified cluster-head in the group. The cluster-head in turn, communicates the aggregated information to a sink or a base station. Heinzelman et. al. [3] showed that hierarchical node clustering where the cluster heads are responsible for optimum data forwarding can result in an efficient organization of a WSN. Node clustering was also proposed Younis et al. [2] to efficiently organize the network topology. In this approach, a hierarchy of nodes is developed where each cluster head is responsible for data aggregation and forwarding. Popular node-based clustering approaches are LEACH [3] and PEGASIS [4]. A hybrid combination of LEACH and PEGASIS, Hybrid Indirect Transmission (HIT), was proposed by Culpepper et al. [5]. In this approach, LEACH-like clusters are used that allow multi-hop routes between cluster-heads and non-head nodes.

Tree-based clustering protocols make use of the routing tree for relaying communications in an ad-hoc network. Routing-based data aggregation was addressed by Intanagonwiwat et al. [6] where data attributes are used to determine routing paths that facilitate data aggregation. The energy efficiency of data centric routing schemes was demonstrated by Krishnamachari et. al. [7]. Ding et. al. [8] propose an efficient energy-aware distributed heuristic to generate the aggregation tree that facilitates data-centric routing.

Cluster-based and tree-based protocols have limited application in large deployments of WSNs. The cluster-based approaches for instance, assume that the network sink can be reached by any node in only one hop. This limits the size of the network for which such protocols are applicable. PEGASIS-based protocols are suitable only for scenarios where packets can be perfectly aggregated into one packet of equal size. Tree-based data aggregation routing protocols impose a significant communication overhead to construct and maintain the tree. These limitations can be simultaneously overcome through the use of mobile reconfigurable cluster heads, where the data aggregation requirements and the interface requirements with the external infrastructure can be used to deploy RCHs whose performance is optimized for the specific application.

In this paper, the design of reconfigurable cluster heads (RCHs) is presented. The RCH is built on a FPGA based platform and can be configured at run-time to implement different aggregation strategies and communication protocols. The ability to dynamically implement clustering and data aggregation is shown to be especially suited to cases where the network size and data aggregation requirements are not specified at the time of deployment. This technique is also shown to result in performance

S. Commuri is with the School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK, 73019 USA (phone: 405-325-4302; fax: 405-325-3442; e-mail: scommuri@ou.edu).

V.Tadigotla is with with Xilinx Inc., Longmont, CO, 80501, USA (phone: 720-652-3105; e-mail: visu.tadigotla@xilinx.com).

enhancement that is not possible using conventional microprocessor-based implementations.

The remainder of the paper is organized as follows. Section II provides a brief overview of design of reconfigurable systems. Section III describes the experimental setup and the implementation of Reconfigurable Cluster Head. The performance criteria used to evaluate the proposed design is presented in Section IV. The experimental results and analysis are provided in Section V and the conclusions are summarized in Section VI.

II. Background On Hardware Reconfiguration

The primary requirement in the design of a RCH is the ability to implement different data aggregation operations. Since the actual aggregation operations and the number of sensor nodes in the cluster are not known ahead of time, the RCH should have the ability to adapt in real-time and implement these operations in an efficient manner. Reconfigurable systems based on Field Programmable Gate Arrays (FPGAs) provide an efficient method of realizing such RCHs.

FPGAs consist of an array of logic cells that can be configured to perform a function. This is accomplished by loading a configuration file into the FPGA. FPGAs were initially used to develop static reconfigurable systems whose functionality was determined by the configuration file loaded into the FPGA at start up [9]. Once implemented, the functionality could only be changed by powering down the system and replacing the FPGA configuration. Full reconfiguration is accomplished by erasing the contents of the FPGA and reloading a new configuration onto the device. On the other hand, Partial Reconfiguration erases only a portion of the FPGA while the rest of the device maintains its original configuration [10], [11]. Dynamic Partial Reconfiguration (DPR) is done while the device is active, i.e. certain areas of the device are reconfigured while the other areas remain operational and are not affected by reconfiguration. Detailed design of partial and dynamically reconfigurable applications using Virtex-II FPGAs can be found in [12], [13].

Virtex FPGAs of Xilinx family support partial reconfiguration and are used to demonstrate the DPR-based design presented in this paper. The key-component of each Xilinx Virtex-II Pro FPGA is the Configuration Logic Block (CLB), which consists of four slices, each of which provides two 4-input function generators, carry logic, arithmetic logic gates, function multiplexers and two storage elements. CLBs are arranged in the FPGA fabric in a grid-like pattern. Reconfiguration in Xilinx Virtex FPGA involves erasing the contents of the CLBs and configuring them with new logic. Virtex-II Pro FPGAs also has Internal Configuration Access Port (ICAP) which provides configuration access for the partial reconfiguration of the FPGA logic. The design includes a PowerPC processor core (PPC405) connected to the high bandwidth processor local bus (PLB) and a bridge connecting the PLB and the on-chip peripheral bus (OPB). The required peripheral modules are simply connected to the OPB. In addition to the PPC405, the associated bus and bus macros, the Virtex-II Pro FPGA consists of 4926 slices of configurable space in which different hardware modules can be implemented. Module based reconfiguration will then result in the creation of new logic elements in the hardware, thereby enhancing the capability of the system. Since the reconfiguration can be done in real-time while the system is operational, system components can be added in real-time to address changing needs during the lifetime of the system.

III. Experimental Setup

The DPR process described in the previous section is used to implement a reconfigurable cluster head in a WSN. The experimental setup consists of a cluster head implemented on a Xilinx Virtex-II pro device equipped with a Wi.232 DTS transceiver. The Wi.232DTS module, manufactured by Radiotronix Inc. [14], implements a wireless serial engine (WiSE) combined with FSK/DTS data transceivers to provide 152.34 Kbit/sec RF data rate, 32 channels of communication in the high power mode and 84 channels in the low power mode. The module also supports different user modes and can be optimized for fixed length / variable length data transmissions and power consumption. Integration of the transceiver with the FPGA requires instantiating an UART (Universal Asynchronous Receiver /Transmitter) module in the static portion of the FPGA and routing the output of the serial port to the output pins of the FPGA to which the transceiver is connected.

In the implementation discussed in this paper, the performance improvement that can be achieved by using the RCH for dynamic data aggregation is investigated. For the sake of analysis, it is assumed that "n" sensor nodes, SN1…SNn, reside in the vicinity of the RCH. Further, it is assumed that each sensor node has an associated identifier (SN ID), and can provide the data (DATA) associated with different parameters (PID). It is also assumed that the sensor nodes can respond to specific queries, represented by Query Identifier (QID). The query can assume the form of a request for immediate data, a periodic broadcast, or an update based on an event trigger. The communication between an individual node and the cluster head follows the protocol packet shown in Figure 1. In the discussion that follows, the ideal case is considered where there are no packet losses due to collisions and each sensor node transmits to the cluster head in sequence. This corresponds to the case of maximum throughput in the network.

In order to aggregate data from sensor nodes, the RCH is configured to extract data from each of the received protocol packets and reformat the packet according to the desired aggregation. The aggregated data packet is then transmitted to the network sink or the next node in the routing sequence. The implementation of the aggregation operation on the RCH is shown in Figure 2. The received data from the sensor nodes is first compared with the ID of the cluster head to ensure that the RCH is the intended destination of the packet. CRC comparison is then performed to ensure that the data is error free and the sensor data is extracted and stored in a two dimensional array. Once

the required data packets are received, the data in this array are aggregated and the result formatted into a protocol packet for onward transmission.

In a typical application, the network sink or a host machine initiates the sensing task by transmitting a query to the cluster head, RCH. The RCH in turn, broadcasts a message and monitors the received power of all the transmissions from the sensor nodes in the vicinity. Based on the power levels of nodes, the RCH organizes the cluster and reconfigures itself to collect and aggregate the data from each of the sensor nodes. After the cluster is organized, different queries from the network host result in appropriate aggregation modules being loaded into the reconfigurable portion of the FPGA. Since the reconfiguration is performed in real-time, the response of the network can be optimized at run-time. Further, since the configuration files can be downloaded to the RCH through the network, the efficiency of the WSN can be improved even in cases where the requirements of the application are not known during deployment. The performance criteria used to evaluate the performance of the WSN is discussed in the next section.

IV. Performance Criteria For Evaluation

The criteria used to evaluate the improvement in the performance of the WSN are presented in this section. Since the goals of the proposed design are to improve throughput and to optimally utilize the available onboard power, performance criteria will be developed to evaluate the savings that can be achieved by implementing dynamic data aggregation using hardware reconfiguration. The following indices are used for evaluating the performance improvement achieved through the use of reconfigurable hardware.

A. Query Processing Time (QPT)

Time for processing queries in a WSN is important especially in applications requiring real-time data monitoring. The processing time for servicing queries in a RCH is dependent on the complexity of the data aggregation algorithm and the associated gate-delays in the hardware. The *Query Processing Time (QPT)* is measured as the overall time taken by the RCH to respond to a query. The QPT includes the time required to parse the query, dynamically implement the required hardware reconfiguration, load appropriate software modules, receive data from the sensor nodes, perform data aggregation, and format a protocol packet to send to the query source. This total execution time is measured using onboard timers and averaged over several runs to compute the QPT.

B. Implementation Efficiency

The efficiency of an implementation is measured in terms of *Hardware Occupancy* which is defined as the number of gates required to implement the circuit in the hardware. As the overall processing time associated with gate delays and the power consumption are both affected by the hardware occupancy, the gate count required for implementing different data aggregation operations should be minimized.

The gate count for each functional module used in the design can be obtained from the synthesis report generated by the Xilinx Embedded Development Kit toolset [22], [23] and is represented in terms of number of slices of the FPGA. Since the device contains a total of 4926 slices for implementing the reconfigurable design, the *Hardware Occupancy* reflects the portion of the FPGA used for the overall design.

C. Power Consumption

In the RCH implementation presented in this paper, the two components of power consumption are the gates in the FPGA and the transceivers. The static power consumption in the FPGA is mainly due to the leakage current in the transistors forming the gate array. As the number of gates (transistors) on the FPGA fabric increases, static power consumption also increases. However the transistor leakage current is very small in comparison to the active current consumed by the device. Therefore, static power consumption in an FPGA is neglected in the analysis to follow. Dynamic (active) power consumption is mainly determined by the switching power of the core of FPGA and the I/O being switched, and can be expressed by the following equation:

$$P_{dynamic} = (CV^2 f)_{FPGA\,Core} + (CV^2 f)_{IO\,Core} \quad (1)$$

where C (Farads), V (Volts), and f (Hz) are the node capacitance, voltage swing and frequency, respectively. From the Xilinx Virtex II pro data sheets [13], it can be seen that the RCH consumes 0.09 μW for each configured slice in the FPGA fabric and each unused slice consumes 0.009 μW .

The power consumed by the Wi.DTS transceiver can be computed from the data sheets of the device [14]. The wireless transceiver consumes 28 mA in the low power mode and 57 mA in the high power mode during transmissions. These transceivers also consume 20 mA in the receive mode. Therefore if the transceiver is operating at 152.34 Kbit/sec, then 6.3 μW (12.6 μW) of power is consumed for transmitting 1 byte of data in the low power mode (high power mode) and 5 μW is consumed in the reception of 1 byte of data.

From the above discussion, the total energy consumption of RCH can be calculated as:

$$Total\ Energy = F\left(E(\text{active mode}), E(\text{Tx/Rx})\right) \quad (2)$$

.where $E(\text{active mode})$ is the energy associated with the dynamic power component of the FGPA and $E(\text{Tx/Rx})$ is energy consumed during the transmission and reception of data.

Cluster ID (CID)	Sensor ID (SN ID)	Time Stamp	Query ID QID	Parameter ID (PID)	DATA	CRC

Figure 1. Format of the packet for communication between the sensor node and the RCH.

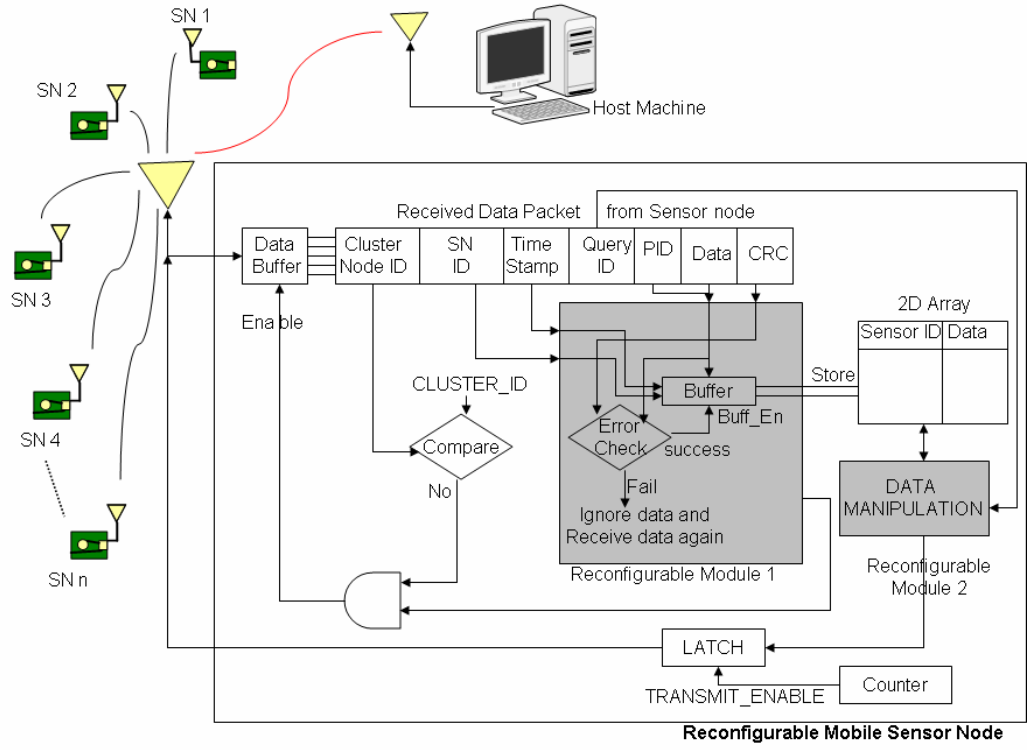

Figure 2. Reconfigurable Cluster Node deployed in a network.

Table 1: Impact of Number of Aggregation Operations on the Performance of the Cluster Head
(R- Relay; CR - Consolidated Relay; $\overline{M}$ - MAX; $\underline{M}$ - MIN; Ave - AVERAGE)

Number of Data Aggregation Operations		SCH Implementation			RCH Implementation		
		Hardware Occupancy (slices)	Query Processing Time (μs)	Power Consumption (in mW)	Hardware Occupancy (slices)	Query Processing Time (μs)	Power Consumption (in mW)
1	R	880	1.02	64.12	880	1.02	64.12
2	R+CR	960	2.31	92.05	1014	1.041	101.09
3	R+CR+ $\overline{M}$	1244	4.57	121.31	1065	1.055	102.11
4	R+CR+ $\overline{M} + \underline{M}$	1456	6.58	149.99	1103	1.072	102.83
5	R+CR+ $\overline{M} + \underline{M}$ +Ave	1671	9.62	178.6	1147	1.089	103.17

V. Experimental Validation

Case Study: Effect of Number of Aggregation Operations

In this case study, the effect of dynamic reconfiguration on the performance of data aggregation operations is investigated using a cluster of five sensor nodes. Five different data aggregation operations, namely relay, consolidated relay, MAX, MIN, and Average, are implemented both in a Static Cluster Head (SCH) and in the Reconfigurable Cluster Head (RCH). It can be seen from Table 1 that the implementation of a simple relay operation (without DPR), requires 880 slices of the FPGA. As the number of operations increase (consolidation, MAX, MIN, AVERAGE) the required number of slices increases to 1671, whereas the maximum number of slices required is 1147 when DPR is used. The efficiency of the static and dynamically reconfigurable implementation are compared in Figure 3, which clearly shows that increasing the number of aggregation operations has a limited impact on the hardware when dynamic reconfiguration is used to implement the desired operation. The impact of the aggregation operations on the processing time is shown in Figure 4.

From Column 3 in Table 1 it can be seen that there is almost a ten-fold increase in the processing time when all the required aggregation operations are implemented in software. The processing time for each data aggregation operation is relatively unchanged when the required aggregation operations are dynamically implemented in the FPGA. It can be seen from Figure 5, that power consumption becomes a serious issue as the number of aggregation operations increase.Thus, implementation of cluster heads using dynamically reconfigurable hardware not only addresses the changing needs of applications at run time, but also leads to efficient implementation by optimizing the processing time and power consumption. This has significant impact in networks where the sensed data has to be transmitted over multiple hops to a network sink. RCHs can be leveraged to address imbalances in the WSNs and to act as a high power relay enabling the individual nodes in the WSN to conserve power.

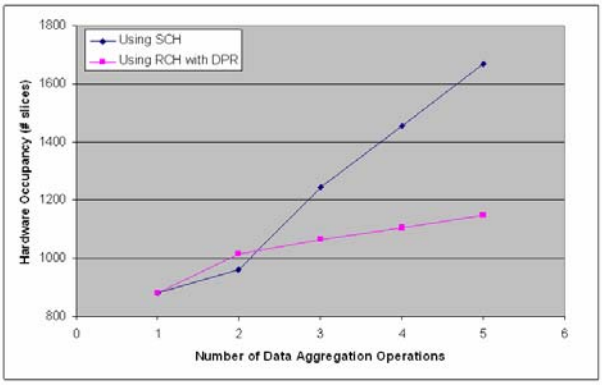

Figure 3. Effect of the Number of Aggregation Operations on Hardware Occupancy.

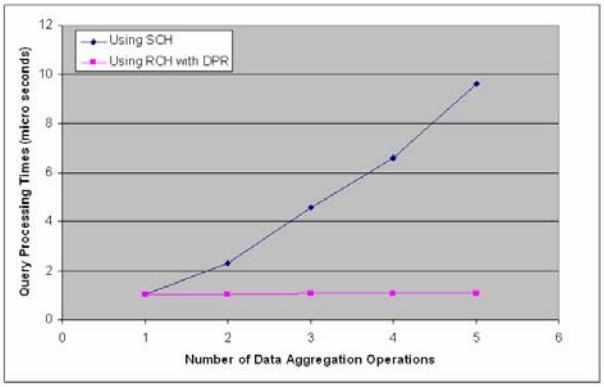

Figure 4. Effect of the Number of Aggregation Operations on the Processing Time.

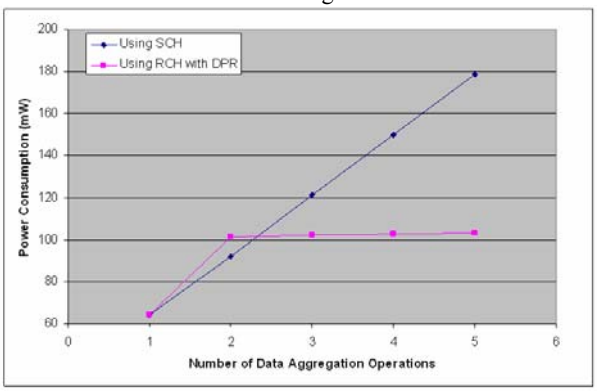

Figure 5. Effect of the Number of Aggregation Operations on the Power Consumption

VI. Conclusions

The implementation of dynamic data aggregation in sensor networks was presented in this paper. Cluster heads based on dynamically reconfigurable Field Programmable Gate Arrays (FPGAs) were shown to result in implementations that can address changing application requirements not known during deployment. The results demonstrate that different data aggregation algorithms can be efficiently implemented on the RCHs at run-time. Such an implementation results in significant reduction in query processing time and also reduces the overall power consumption in a network, thereby increasing the useful life of a WSN. The analysis presented in the paper shows that the power consumption and the query processing times increase rapidly as the number of aggregation operations increase. The size of available hardware and the time required to process queries in software also restrict the number of possible data aggregation methods that can be implemented in static networks. The use of dynamically reconfigurable hardware proposed in this paper efficiently circumvents this problem. Design of such RCHs can alleviate the interface issues in heterogeneous networks, while simultaneously improving network performance.

REFERENCES

1. F. Ordonez and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE Journal on Selected Areas in Communications, special issue on Fundamental Performance Limits of Wireless Sensor Networks*, vol. 22, no. 6, pp. 1121-1129, August 2004.
2. O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366 – 379, October-December 2004.
3. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.
4. S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 924-935, September 2002.
5. B. J. Culpepper, L. Dung, and M. Moh, "Design and analysis of hybrid indirect transmissions (HIT) for data gathering in wireless micro sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, pp. 61-83, January 2004.
6. C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," *Proc. 22nd Int. Conf. Distributed Computing Systems*, July 2-5, Vienna, Austria, pp. 457-458, 2002.
7. B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," *Proc. 22nd Int. Conf. Distributed Computing Systems*, July 2-5, Vienna, Austria, pp. 575–578, 2002.
8. M. Ding, X. Cheng, and G. Xue, "Aggregation Tree Construction in Sensor Networks," *Proc. Of IEEE Vehicular Technology Conference*, October 6-9, Orlando, Florida, USA, vol. 4, pp. 2168 – 2172, 2003.
9. A. Upegui and E. Sanchez, "Evolving hardware by dynamically reconfigurable Xilinx FPGAs," *IEEE Conference on Evolvable Systems,* September 12-14, Sitges, Spain, pp. 153-162, 2005.
10. V. Tadigotla, S. Commuri, "Design and implementation of reconfigurable mobile sensor systems," *WSEAS Transactions on Systems*, vol. 6, no. 2, pp. 400 – 408, February 2007.
11. S. Commuri, V. Tadigotla, and L. Sliger, "Task-based hardware reconfiguration in mobile robots using FPGAs," *International Journal of Intelligent and Robotic Systems*, 2007 (to appear).
12. Xilinx Inc., Two flows for Partial Reconfiguration: Module Based or Difference based, *Application Note XAPP290*, version 1.2, Xilinx, 2004.
13. *Virtex – II platform FPGA user guide*, version 1.8, Xilinx Inc., 2005.
14. *Wireless Serial Engine*, http://www.radiotronix.com/datasheets/quickstarts/Wi232DTS_QS.pdf.