2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines

Low Power Reconfigurable Controllers for Wireless Sensor Network Nodes

Vivek D. Tovinakere*, Olivier Sentieys†, Steven Derrien‡ and Christophe Huriaux†
*NVIDIA Corporation, Bangalore 560045, India. Email: vivektd@gmail.com
†INRIA/IRISA, Université de Rennes 1, 22300 Lannion, France. Email: {sentieys, christophe.huriaux}@irisa.fr
‡INRIA/IRISA, Université de Rennes 1, 35042 Rennes, France. Email: sderrien@irisa.fr

***Abstract*—A key concern in the design of controllers in wireless sensor network (WSN) nodes is the flexibility to execute different control tasks involving sensing, communications and computational resources of the node. In this paper, low power flexible controllers for WSN nodes based on reconfigurable microtasks composed of an FSM and datapath are presented. Coarse grain power gating opportunities are exploited in FSM and datapath for low power operation in reconfigurable microtasks. Power estimation results on typical benchmark microtasks show a 2× to 5× improvement in energy efficiency *w.r.t* a microcontroller at a cost of 5× relative to a microtask implemented as an ASIC with higher NRE costs.**

***Keywords*-Low power, microcontrollers, power gating, reconfigurable hardware, wireless sensor networks**

I. Introduction

Wireless Sensor Network (WSN) nodes typically need to process signals from sensors and perform transceiver tasks as part of wireless communications in the network. Adaptive nodes may need to change their roles dynamically while the internal resources of a node are also to be managed implying that flexibility of controllers is an important consideration in the design of WSN nodes. An important aspect in the design of WSN nodes that conflicts with flexibility is their energy efficiency. It is estimated that upto 25% of total power budget in a node may be consumed by controllers in active mode and a considerably higher fraction in idle mode. Hence it is important to optimize controllers integrated within a node for low power in both active and standby states.

In general, microcontrollers and FPGAs offer flexibility but are not energy efficient as dedicated circuits. WSN platforms described in [1] and [2] have used low-power microcontrollers as their driving engines for control tasks. A spectrum of low power techniques - from clock gating to power gating and subthreshold design [3], [4] - have been implemented in such microcontrollers. Nevertheless, microcontrollers are far from being optimal solutions in several applications as sequential execution of instructions for control tasks exercise significant logic of a microcontroller to cause high power consumption. Several works have focussed on low power optimizations of island-type FPGA architectures using subthreshold supply voltages, dynamic voltage scaling, power-efficient routing fabric [5], power gating [6] and different circuit techniques for LUT design. An outcome of these works has been customization of reconfigurable systems for specific classes of energy-constrained applications. In this paper, the intermediate design space between dedicated ASIC solutions and microcontrollers is explored. The focus of our work is to identify architectures for reconfigurable controllers that are amenable to low power optimizations and yet retain flexibility for a variety of control tasks in WSN nodes.

II. Microtask-based Controllers

Consider a controller that is required to execute combinations of M control tasks T_i, $i = 0, 1, 2, ..., M-1$ as specified by a task flow graph (Figure 1(a)). A microtask is a physical realization of T_i and may be obtained as combination of an FSM for control and a datapath for computations. A system level view of realization of a task flow graph as a set of microtasks is shown in Figure 1(b). A typical structure of microtask is shown in Figure 2.

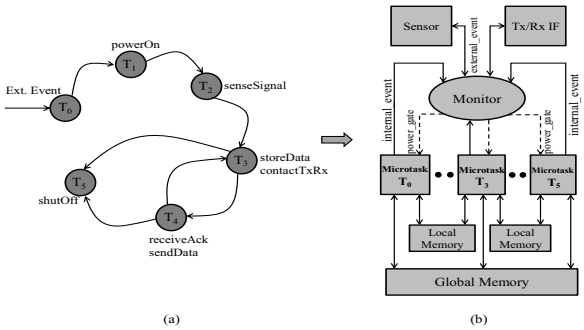

Figure 1. (a) A task flow graph and (b) system-level view of a task flow graph.

A microtask is dedicated to execution of a specific control task once it is generated by the design flow. A system monitor manages scheduling of microtasks according to the task flow graph. In order to exploit low-power techniques in generated controllers, the tasks are mapped onto microtasks in such a way that, during the operation of the controller based on a run-to-completion semantic, all microtasks than those required may be power-gated to suppress leakage power. However, unlike microcontrollers, the utility of generated microtasks is limited due to the specificity of application. Reconfigurable circuits offer an alternative in terms of flexibility, particularly in specific

978-1-4799-5111-6/14 $31.00 © 2014 IEEE
DOI 10.1109/FCCM.2014.68

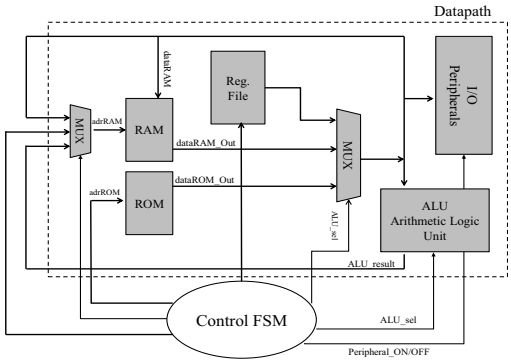

Figure 2. Structure of a typical microtask T_i.

application domains like WSNs. In this paper, low power reconfigurable architectures with routing fabric customized for FSMs and variable-precision adders in datapaths targeted towards flexible microtasks are presented. We use power gating to suppress high leakage power in nanoscale CMOS circuits. Whereas energy savings in standby mode of system operation due to power gating is obvious for low duty cycle operation, it is sought to achieve active mode leakage reduction in FSM and datapath elements by a divide and conquer approach.

III. Low Power Reconfigurable FSMs

A. Notations

Let at time unit t, the n primary inputs to the FSM be denoted by the vector $\mathbf{x}(t) = [x_0(t), x_1(t), ..., x_{n-1}(t)]$, the m outputs of FSM by $\mathbf{y}(t) = [y_0(t), y_1(t), ..., y_{m-1}(t)]$ and the state vector of N-bit state register by $\mathbf{s}(t) = [s_0(t), s_1(t), ..., s_{N-1}(t)]$. For notational convenience, we use the triplet (N, n, m) to denote the FSM parameters defined above. The next-state functions of an FSM of Moore type may be written as

$$s_i(t+1) = \sum_{k=0}^{2^{(n+N-K)}-1} m_k f_i(n(m_k), ..., s_{N-1})_k \qquad (1)$$

where K corresponds to number of variables on which $f_i(.)_k$ depends after Shannon decomposition and $\sum$ denotes logical-OR of the boolean functions. The minterm generated by first $n + N - K$ input variables of the sequence $x_i, ..., s_i(t+1)$ is denoted by m_k. Similarly, the output functions may be written as

$$y_l = \begin{cases} \sum_{k=0}^{2^{N_s}-1} m_k g_l(n(m_k), ..., s_{N-1})_k & K_{op} < N \text{ (Case 1)}, \\ g_l(s_0, ..., s_{N-1})_k & K_{op} \geq N \text{ (Case 2)} \end{cases} \qquad (2)$$

where K_{op} denote the number of inputs of a K_{op}-LUT and $N_s = N - K_{op}$. The FSM is fully reconfigurable if its realization can be configured to support any set of boolean functions f_i and g_l temporally. Assuming that each $f_i(.)_k$ can be realized with K-LUTs and $g_l(.)_k$ with K_{op}-LUTs, it can be inferred that realization of $\mathbf{s}(t+1)$ and $\mathbf{y}(t)$ requires the resources as shown in Table I.

Resources	Next-state functions	Output functions
# LUTs	$N \times 2^{(n+N-K)}$ K-LUTs	$m \times 2^{N_s}$ K_{op}-LUTs
Decoder Size	$(n+N-K)$-to-$2^{(n+N-K)}$	$(n+N-K)$-to-$2^{(n+N-K)}$ (shared)
AND logic OR logic	$N \times 2^{(n+N-K)}$ $2^{(n+N-K)}$	$m \times 2^{(n+N-K)}$ $2^{(n+N-K)}$ (Case 1)
Configuration Bits	$N \times 2^{n+N}$	$m \times 2^N$

Table I
Resources for Reconfigurable FSMs

B. Power Gating Opportunities in Reconfigurable FSMs

In this work, we seek to identify power gating opportunities in reconfigurable FSMs at a granularity of LUT decoding logic to reduce leakage power. It should be noted that to preserve configuration data, the reconfiguration memory needs to be in always-on power domain. Power gating

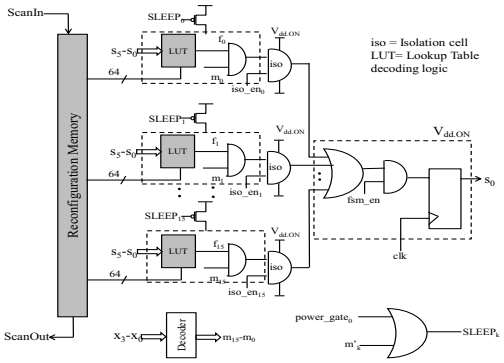

Figure 3. Power gating opportunity for aggressive active mode energy savings in a reconfigurable FSM.

opportunities can be identified in Figure 3 at the granularity of a LUT and associated AND gates. This network of sleep transistors requires as many control signals as the number of LUTs. When the reconfigurable FSM is configured as a particular FSM, its operation depends on some of the primary inputs that possibly vary slowly and therefore, depending on the values of those inputs all but one of the minterms evaluate to logic 0. The decoder outputs, then can function as control signals ($SLEEP_k = power\_gate_i + m'_k$ in Figure 3) to power-gate the respective LUT while also eliminating the need for a separate controller.

C. Overall Architecture

The overall architecture for a power-gated reconfigurable FSM is shown in a schematic form in Figure 4. Each unit of the architecture corresponds to Figure 3. N such units constitute the logic for state register bits and m units corresponding to FSM outputs. The power consumption at any given time is due only to $N + m$ LUT logic clusters apart from blocks in always-on domain.

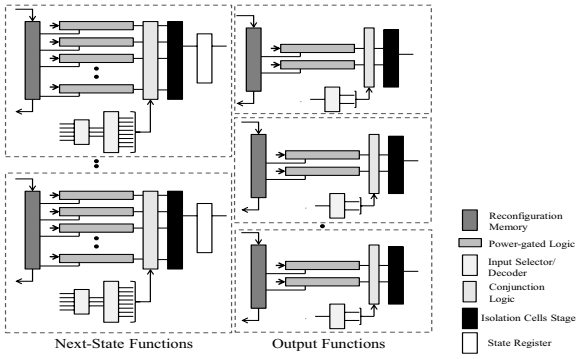

Figure 4. Schematic diagram of the overall architecture of scalable power-gated reconfigurable FSM.

D. Low Power Reconfigurable Precision Adders

In this work, we use a 32-bit adder partitioned into 4 clusters, each representing slices of 8-bit datapath along with their input and output stages. The clusters together represent the parallel prefix carry generation logic of the adder. Each cluster, along with its input and output stages are power-gated by sleep transistors. The sizes of sleep transistors are approximately 10% of the total size of all transistors in the power-gated cluster.

IV. Power Estimation and Analysis

The total average power consumption of the power-gated architecture for a particular mapping of FSM is given by Equation (3) in the next page, where N_{FSM} and m_{FSM} are number of mapped state register bits and outputs, N_{CB} denotes the total number of configuration bits, $P_{static,LUT}$, $P_{static,CB}$, $P_{static,SR}$, $P_{static,iso}$ and $P_{static,IPD}$ denote the static power of K-LUT logic, a configuration bit register, state register bit, isolation cell and input selector-decoder respectively. Further, $E_{dyn,LUT}$ and $E_{dyn,IPD}$ represent the average dynamic energy components of LUT logic and input selector-decoder due to changes in inputs. An average activity factor for transitions is assumed by means of $f_{inp,av}$, the average rate of change of inputs. We consider the worst case with wakeup transition for all state register bits and outputs to account for wakeup energy E_{wu}.

We consider a 6-LUT for lookup table implementations in reconfigurable FSM. The combinational logic of 6-LUT is synthesized using a 65nm industrial technology library. Switching energy for a single LUT is determined by applying 10000 sets of random inputs and computing the average total switched load for one input set at a supply voltage of 1.0V as

$$E_{dyn_{LUT}} = \frac{V_{dd} r_1}{N_{input}} \sum_{i=1}^{N_{input}} C_{sw_i,LUT} \tag{4}$$

where $C_{sw_i,LUT}$ is the total switched capacitance of the LUT per input set and r_1 is the steady state Virtual-V_{dd}. The various parameters of power-gated 6-LUT decoding logic is shown in Table II. The total average energy consumption

Power-Gated K-LUT Parameter	K=6	
P_{static} at V_{dd}=1V (μW)	12.56	
Mean switching snergy per state or input transition	0.059pJ	
Sleep transistor width W (μm)	0.54	12
Steady state Virtual-V_{dd} r_1 (mV)	980	998

Table II
Power-gated K-LUT parameters for power estimation.

per operation (clock cycle) is obtained from

$$E_{op,FSM} = \frac{P_{static,FSM}}{f_{clk}} + E_{dyn,FSM}. \tag{5}$$

A. Energy Efficiency and Cost of Flexibility

A metric to measure energy efficiency of different realizations is the equivalent energy per instruction of a low power microcontroller. We consider controller task implementations on openMSP430 [8], an opencores microcontroller with an instruction set isomorphic to MSP430 used in WSN platforms, to compare microtask based implementations across different tasks. The energy per operation of a microtask $E_{op,MT}$ is given by

$$E_{op,MT} = (P_{static,FSM} + P_{static,adder} + P_{static,RF})/f_{clk} + E_{dyn,FSM} + E_{dyn,adder} + E_{dyn,RF} \tag{6}$$

where $P_{static,adder}, P_{static,RF}, E_{dyn,adder}$ and $E_{dyn,RF}$ represent static power of 32-bit adder power-gated for 16-bit addition, register file and energy per operation of adder and register file respectively. The total energy for execution of a task is then

$$E_{task} = N_{st} E_{op,MT} \tag{7}$$

where N_{st} is the number of operations or state transitions required to execute the task. Hence the equivalent energy per instruction is determined as

$$E_{eff} = \frac{E_{task}}{N_{inst}} \tag{8}$$

where N_{inst} is the number of instructions required to execute the same task on the microcontroller. Table III shows the metric for control tasks realized with a microcontroller, reconfigurable microtasks and hardwired microtasks. Among the three realizations, hardwired microtasks represent the most energy efficient implementation for a specific controller since they are obtained as a result of ASIC synthesis. In this exercise, a register file of size 16 $\times$ 16 was used. It can be inferred from the table that among hardwired and reconfigurable microtasks, the latter has about 5$\times$ cost in the energy per operation of the benchmark microtask.

$$P_{total,FSM} = (N_{FSM} + m_{FSM})P_{static,LUT} + N_{CB}P_{static,CB} + 2^N(N_{FSM}2^{n-K} + m_{FSM}2^{-K_{op}})P_{static,iso} + N_{FSM}(P_{static,IPD} + P_{static,SR}) + N_{FSM}E_{dyn,IPD}f_{inp,av} + (N_{FSM} + m_{FSM})(E_{wu} + E_{dyn,LUT})f_{inp,av}. \quad (3)$$

Similarly among flexible controller implementations, it can be inferred that energy per instruction in microcontrollers is typically higher than energy per operation of reconfigurable microtasks assuming that in both realizations energy consumption is same for a control task. It should be noted that the energy per instruction metric for openMSP430 microcontroller excludes energy due to instruction and data memories making the comparison equitable.

Microtask [7]	Equivalent Energy per Instruction (pJ)					
	openMSP430 (without memories)		Reconfigurable Microtasks		Hardwired Microtasks	
	N_{inst}	E_{eff}	N_{st}	E_{eff}	N_{st}	E_{eff}
Crc8 (6,3,16)	30	163	71	31.60	71	8.1
receiveData (6,3,23)	66	230	332	83.53	332	15.7
Crc16 (7,4,19)	27	170	73	41.27	73	9.3
firBasic (7,3,21)	58	179	168	46.90	168	26.1

Table III
EQUIVALENT ENERGY PER INSTRUCTION IN THREE REALIZATIONS OF NODE CONTROLLERS.

Table IV gives a comparison of area estimates of hardwired microtasks and the proposed reconfigurable microtask. From a controller's perspective, a WSN application would typically require about 50 tasks to be integrated. The total area of a controller would be the sum of areas of all microtasks and associated system monitor and memories. In principle, a reconfigurable microtask can be used in place of hardwired microtasks by time-multiplexing tasks at the controller level. The advantages of power gating are present in both active and standby modes of operation.

Microtask	Hardwired Microtask(μm^2)	Reconfigurable Microtask (μm^2)
Crc8	3097	140522.2
receiveData	2858	
Crc16	3102	
firBasic	7164	

Table IV
COMPARISON OF AREAS OF 16-BIT HARDWIRED AND RECONFIGURABLE MICROTASKS.

V. CONCLUSION

A common thread in our exploration of scalable reconfigurable architectures for low power in this paper was to identify and power-gate unused logic for aggressive leakage power savings in active modes of operation. An analysis of power estimation in reconfigurable microtasks show that they offer flexibility with a significantly reduced cost in equivalent energy per instruction compared to a microcontroller. The primary drawback of LUT-based flexible circuits is the large reconfiguration memory that consumes substantial power as they must always remain in ON state. A useful approach at an architecture level is to investigate alternate reconfiguration mechanisms and is proposed for future work.

VI. ACKNOWLEDGEMENTS

The first author wishes to acknowledge University of Rennes 1, France for supporting this work carried out in the CAIRN/IRISA Lab., Lannion, France.

REFERENCES

[1] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor networks*, Picastaway, NJ, 2005, pp. 364–369.

[2] J. Beutel, O. Kasten, and M. Ringwald, "BTnodes - a distributed platform for sensor nodes," in *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems*, New York, 2003, pp. 292–293.

[3] M. Seok, S. Hanson, Y.-S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "The Phoenix processor: A 30pW platform for sensor applications," in *Proceedings of the IEEE Symposium on VLSI Circuits*, 2008, pp. 188–189.

[4] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y. Chee, and J. Rabaey, "A power-managed protocol processor for wireless sensor networks," in *Proceedings of the IEEE Symposium on VLSI Circuits*, 2006, pp. 212–213.

[5] M. Lin and A. El Gamal, "A low-power field-programmable gate array routing fabric," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1481–1494, 2009.

[6] S. Ishihara, M. Hariyama, and M. Kameyama, "A low-power FPGA based on autonomous fine-grain power gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1394–1406, August 2011.

[7] L. Nazhandali, M. Minuth, and T. Austin, "SenseBench: Toward an accurate evaluation of sensor network processors," in *Proceedings of the IEEE International Workload Characterization Symposium*, 2005, pp. 197–203.

[8] Opencores, "The openMSP430 User Guide," http://www.opencores.org, 2009.