Asynchronous Circuit Design for Wireless Sensor Nodes: A Survey

GouriWazurkar
ShriRamdeobaba College of Engineering & Management, Nagpur.

S. L. Badjate, Ph.D
S. B. Jain Institute of Technology, Management & Research, Nagpur.

ABSTRACT

Computer architecture researchers evaluate key areas such as pipelining, organization, instruction issue, branching, and exception handling when considering asynchronous and synchronous design and implementation trade-offs. Asynchronous or clockless designs are considered as an alternative to conventional synchronous digital system design. The major advantages of asynchronous are low power consumption, better modularity, higher robustness and higher speed. Virtually all processors are synchronous which are based on internal timing devices / circuits that regulate processing. As system becomes increasingly large and complex, this timing device a clock can cause big problems with clock skew and timing delay can create havoc with the overall design. It can also increase the circuit silicon and power dissipation.To overcome above limitations asynchronous design is considered aggressively. Each subsystems or functional blocks may be optimized without being synchronized to a global clock that may simplify interfacing. Thus the performance of the asynchronous system exhibits the average performance of the overall subsystems or functional block. Furthermore, asynchronous processors may yet prove to offer reduced power dissipation by inherently shutting down unused portions of the circuit.

Keywords: Asynchronous Design, Low Power Consumption, Wireless Sensor Node.

1. INTRODUCTION

Our world is becoming increasingly connected and instrumented with sensors. Improvements in microelectronics and integrated systems have made possible sensor platforms that are a few millimeters in dimension. The possible applications of sensor platforms are varied, and include: smart home systems monitoring temperature, humidity, and movement; vibration sensors for earthquake monitoring; stress/strain sensors for monitoring materials and machines; gas sensors for detection of chemical substances; biological sensors for the detection of microorganisms and environmental monitoring; habitat monitoring to study species in their natural environment, machine surveillance and preventive maintenance, precision agriculture, medicine and health care and many more [53]. One of the key issues in the design of wireless sensor node is power consumption of each component in the node and the network as a whole [26]. Hence the design of low energy processor for wireless sensor node, optimized for data monitoring operations in sensor networks is essential. Handling such a wide range of application types will hardly be possible with any single realization of a wireless sensor network (WSN). Realizing these characteristics with new mechanisms is the major challenge of the vision of WSNs. The various challenges in the design of WSN are type of service, quality of service, fault tolerance, lifetime, scalability, wide range of densities, programmability and maintainability. To realize these requirements, innovative mechanisms for a communication network have to be found, as well as new architectures, and protocol concepts. A particular challenge here is the need to find mechanisms that are sufficiently specific to the idiosyncrasies of a given application to support the specific quality of service, lifetime, and maintainability requirements. Some of the mechanism that forms the part of the WSNs are multihop wireless communication, energy efficient operation, auto-configuration, collaboration and in network processing, data centric and locality [53]. In many scenarios, nodes will have to rely on a limited supply of energy (using batteries). Replacing these energy sources in the field is usually not practicable, and simultaneously, a WSN must operate at least for a given mission time or as long as possible. Hence, the lifetimeof a WSN becomes a very important figure of merit. Evidently, an energy-efficient way of operation of the WSN is necessary. As an alternative or supplement to energy supplies, a limited power source (via power sources like solar cells, for example) might also be available on a sensor node. Typically, these sources are not powerful enough to ensure continuous operation but can provide some recharging of batteries. Under such conditions, the lifetime of the network should ideally be infinite. The lifetime of a network also has direct trade-offs against quality of service: investing more energy can increase quality but decrease lifetime. Concepts to harmonize these trade-offs are required. To support long lifetimes, energy-efficient operation is a key technique. One of the key issues in the design of wireless sensor node is power consumption of each component in the node and the network as a whole. Hence the design of low energy processor for wireless sensor node, optimized for data monitoring operations in sensor networks is essential. Asynchronous processor can be design with hardware support for performing regular operations in sensor node that can maximize lifetime of the node and minimize power [32].

Asynchronous or clockless designs are considered as an alternative to conventional synchronous digital system design [1] [2] [3] [4] [5]. The major advantages of asynchronous are low power consumption, better modularity, higher robustness and higher speed. Virtually all processors are synchronous which are based on internal timing devices / circuits that regulate processing. As system becomes increasingly large and complex, this timing device a clock can cause big problems with clock skew and timing delay can create havoc with the overall design. It can also increase the circuit silicon and power dissipation. To overcome above limitations asynchronous design is considered aggressively. Naturally modular design approach is followed in asynchronous designs. Each subsystems or functional blocks may be optimized without being synchronized to a global clock that may simplify interfacing. Thus the performance of the asynchronous system exhibits the average performance of the overall subsystems or functional block [12]. Furthermore, asynchronous processors may yet prove to offer reduced power dissipation by inherently shutting down unused portions of the circuit. Processors design may follow a locally synchronous, globally asynchronous approach where individual functional units use a local clock signal but are asynchronous with other functional units on the circuit die [36]. The problem with clock distribution is thereby minimized while the processor retains the advantages of a synchronous system. One possible approach may perform instruction encoding and issue asynchronously, but the instructions themselves will be distributed to synchronous execution units. In general asynchronous methodology may be beneficial to those functions that are simplistic to do sequentially but complex to do in parallel. Asynchronous methodology can exploit the simplicity provided by sequential computation while attaining performance benefits by beginning the next computation as soon as the previous one is completed, instead of having to wait for the next clock pulse.

2. Node Architecture

When choosing the hardware components for a wireless sensor node, evidently the application's requirements play a decisive factor with regard mostly to size, costs, and energy consumption of the nodes – communication and computation facilities. In realistic applications, the mere size of a node is not so important; rather, convenience, simple power supply, and cost are more important. The basic WSN node comprises of five main components controller, memory, sensors and actuators, communication and power supply as shown in Fig. 1 [53].

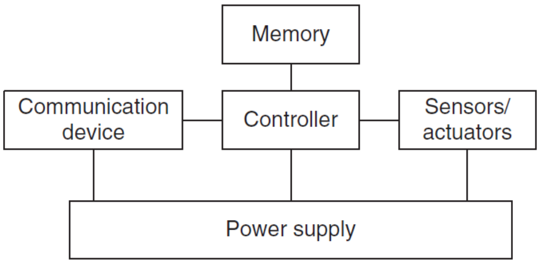

Figure 1: Basic WSN node Hardware components

Each of these components has to operate balancing the trade-off between as small energy consumption as possible on the one hand and the need to fulfill their tasks on the other hand. For example, both the communication device and the controller should be turned off as long as possible. To wake up again, the controller could, for example, use a preprogrammed timer to be reactivated after some time. Alternatively, the sensors could be programmed to raise an interrupt if a given event occurs. Supporting such alert functions requires appropriate interconnection between individual components. Moreover, both control and data information has to be exchanged along these interconnections.

3. CONTROLLER

The controller is the core of a wireless sensor node. It collects data from the sensors, processes this data, decides when and where to send it, receives data from other sensor nodes, and decides on the actuator's behavior [53]. It has to execute various programs, ranging from time-critical signal processing and communication protocols to application programs; it is the Central Processing Unit (CPU) of the node. Such a variety of processing tasks can be performed on various controller architectures, representing trade-offs between flexibility, performance, energy efficiency, and costs. Options available for the implementation of the controller are digital signal processors, microcontrollers, general purpose microprocessor, FPGAs [40] and ASIC. Controller can be implemented using FPGAs rather then using microcontrollers or microprocessor. An FPGA can be reprogrammed (or rather reconfigured) “in the field” to adapt to a changing set of requirements; however, this can take time and energy. The typical trade-off here is loss of flexibility in return for a considerably better energy efficiency and performance. On the other hand, where a microcontroller requires software development, FPGAs provide the same functionality in hardware.

3.1 Relation between Computation and Communication

Looking at the energy consumption numbers for controllers and radio transceivers, an evident issue is to determine the best way to invest the precious energy resources of a sensor node. Whether it is better to send data or to compute or what is the relation in energy consumption between sending data and computing? Thus the result is communicating 1 KB of data over 100 m consumes roughly the same amount of energy as computing three million instructions. It is clear that communication is a considerably more expensive undertaking than computation. This basic observation motivates a number of approaches and design decisions for the networking architecture of wireless sensor networks. The core idea is to invest into computation within the network whenever possible to safe on communication costs, leading to the notion of *in-network processing* and aggregation [53].

3.2 Asynchronous Design

Asynchronous or clockless design has long been pursued as an alternative to the conventional synchronous paradigm. The related design techniques generally claim to have the potential for lower power consumption, better modularity, higher robustness, and higher speed [12]. Given the increasing rates of transient faults, the inherent robustness of clockless designs may turn out as a crucial property. In asynchronous circuits, synchronization is indicated using a handshaking protocol. Therefore, a global clock is no longer necessary in these circuits. The absence of global clock makes the standby power consumption to be zero. A synchronous microprocessor can also enter a low-power idle state, but only with considerable effort. The global clock must be gated off to all parts of the system, except for the interrupt circuit. An interrupt must gate the global clock backon. Since the global clock still ticks in the idle state and the power overheads of clock gating are not avoided, the standby power consumption of a synchronous microprocessor can be quite significant compared to that of an asynchronous microprocessor. Asynchronous design can thus achieve near zero standby dynamic power consumption quickly and efficiently with very little overhead. The standby power consumption is extremely important in sensor nodes, because they may spend 99% of their time in an idle state. A node is temporarily waked up by sensing events, and after that, it goes back to sleep again. In such an application, standby power is dominant and asynchronous design gains big advantage in standby power saving. Asynchronous design can also minimize active power consumption because of its automatic fine-grained power management. In an asynchronous circuit, synchronization is represented by local hank-shake signals. No control signals are propagated to the blocks that are not required to perform operations, thus no activities and power consumption are imposed in these blocks. The power consumption used by an asynchronous circuit only depends on the useful operations need to do. In a synchronous circuit, on the other hand, blocks are connected with a global clock. Even with a clock-gating technique, the registers and clock-gating gates are switched with the global clock, introducing unnecessary activities.

3.3 FPGA Implementation of Asynchronous Circuits

Commercial FPGAs, such as Xilinx Spartan series FPGAs and Altera Cyclone series FPGAs [40] [8] are designed for synchronous (clocked) circuits and not for asynchronous ones. The limitations of the basic structure of these FPGAs introduce many difficulties when mapping an asynchronous circuit onto these FPGAs. The main difficulty comes from the fact that wire delays are no longer ignorable. The FPGA is organized by a large number of Slices arranged in a matrix. Each slice contains a few look-up tables and registers, in which, logics are mapped. Routing is implemented using local switchboxes (LSBs) and global switch-boxes (GSBs). An output (except for the carry signals) of a slice cannot directly connect an input of a slice nearby. The connection must be routed via switch-boxes, resulting in a long delay. This property makes it difficult to implement asynchronous elements, especially for those with feedback signals. The asynchronous circuits are designed using a bundled-data encoding scheme [21]. The correctness of a circuit using a bundled-data scheme depends on the assumption that the delay of each block is predictable and designers can use a delay-matching block to satisfy timing constraint. In this case, bundled-data asynchronous design is somewhat like synchronous design, in which, commercial timing analysis tools, like Xilinx timing analyzer, are used to find the longest delay path(s) in a circuit. Based on timing analysis data, a global clock cycle is defined. To maximize performance, asynchronous circuits are usually designed using an asynchronous pipelining technique. Fig.2 shows asynchronous bundled-data pipeline architecture. The delay matching blocks are inserted in control channel to make sure that the operations of the function blocks finish before the hand-shake control signals arrive [21].

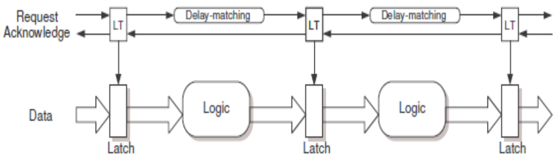

Figure. 2: Asynchronous Bundle Data Pipeline Architecture

A low-power asynchronous event-driven sensor network processor can be designed onto a commercial clocked FPGA. The processor can employ a bundled-data asynchronous encoding scheme, which makes it possible to use the clocked FPGA design tools provided by Xilinx Company. The properties of the processor may be low active and standby power dissipation and quick response time.

4. CONCLUSION

Asynchronous or clockless designs are considered as an alternative to conventional synchronous digital system design. The major advantages of asynchronous are low power consumption, better modularity; higher robustness and higher speed. Processors design may follow a locally synchronous, globally asynchronous approach where individual functional units use a local clock signal but are asynchronous with other functional units on the circuit die. The problem with clock distribution is thereby minimized while the processor retains the advantages of a synchronous system.

REFERENCES

[1] A. J. Martin et al, “Asynchronous Techniques for System on Chip Design,” in *Proceedings of IEEE*, vol. 94, no. 6, June 2006, pp 1089 - 1120.

[2] G. M. Jacobs et al, “A Fully Asynchronous Digital Signal Processor using Self Timed Circuits,” in *IEEE Journal of Solid State Circuits*, vol. 25, no. 6, Dec. 1990, pp 1526 - 1537.

[3] T. H. Y. Meng, et al, “Asynchronous Design for Programmable Digital Signal Processors,” in *IEEE Transaction on Signal Processing*, vol 39, no. 4, April 1999, pp 939 - 952.

[4] T. H. Y. Meng, et al, “Asynchronous Processor Design for Digital Signal Processing,” in *IEEE International Conference on Acoustic, Speech and Signal Processing*, 1988, pp 2013 - 2016.

[5] K. Y. Yun, et al, “High Performance Asynchronous Pipeline Circuits,” in *IEEE International Symposium on Advance Research in Asynchronous Circuits and Systems*, 1996, pp 17 - 28.

[6] L. S. Neilsen, et al, “A Low Power Asynchronous Data Path for FIR Filter Bank,” in *IEEE International Symposium on Advance Research in Asynchronous Circuits and Systems*, 1996, pp 197 - 207.

[7] C. J. Elston et al, “Hades Towards The Design of an Asynchronous Superscalar Processor, “in *IEEE Conference on Asynchronous Design Methodologies*, 1995, pp 200 – 209.

[8] R. Payne, “Asynchronous FPGA Architecture,” in *IEEProceedings of Compute Digit Technology*, vol. 143, no. 5, Sep. 1996, pp 282 – 286.

[9] W. H. F. J. Korver et al, “Asynchronous Implementation of the SCPP-A Counter flow Pipeline Processor,” in *IEEProceedings of Compute Digit Technology*, vol. 143, no. 5, Sep. 1996, pp 287 – 294.

[10] Alex Semenov et al, “Designing an Asynchronous Processor using Petri Nets,” in *IEEE Micro*, April 1997, pp 54 – 64.

[11] Martin Benes et al, “A High Speed Asynchronous Decompression Circuits for Embedded Processor,” in *7thIEEE Conference in Advance Research in VLSI*, 1997, pp 219 – 236.

[12] Tony Werner et al, “Asynchronous Processor Survey,” in *IEEE Computer*, Nov. 1997, pp 67 – 78.

[13] K. T. Christensen et al, “The Design of an Asynchronous TinyRISC TR4101 Microprocessor Core,” in *IEEE*

International Symposium on Advance Research in Asynchronous Circuits and Systems, 1998, pp 108 - 119.

[14] C. Ortega, et al, "Static Power Reduction Techniques for Asynchronous Circuits," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2010, pp 52 - 61.

[15] Clinton Kelly et al, "SNAP A Sensor Network Asynchronous Processor," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2003, pp 24 - 33.

[16] C. LaFrieda, et al, "Reducing Power Consumption with Relaxed Quasi Delay Insensitive Circuits," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2009, pp 217 - 226.

[17] RanjitManohar, "Reconfigurable Asynchronous Logic," in *IEEE International Custom Integrated Circuits Conference*, 2006, pp 13 - 20.

[18] M. Hirayama, "VLSI Oriented Asynchronous Architecture," in *13th IEEE International Symposium on Computer Architecture*, 1986, pp 290 – 296.

[19] B. R. Sheikh, et al, "An Operand Optimized Asynchronous IEEE 754 Double Precision Floating Point Adder," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2010, pp 151 - 162.

[20] Mika Nystrom et al, "A pipelined Asynchronous Cache System," in *Caltech Technical Report*, 2003, pp 1 – 10

[21] Y. Liu et al, "Designing an Asynchronous FPGA Processor for Low Power Sensor Network," in *IEEE International Symposium on Signals, Circuits and Systems*, 2009, pp 261 – 266.

[22] T. N. Prabakar, et al, "FPGA Based Asynchronous Pipelined Multiplier with Intelligent Delay Controller," in *IEEE International SoC Design Conference*, 2008, pp 304 – 309.

[23] BabakRahbaran, et al, "Is Asynchronous Logic More Robust Then Synchronous Logic," in *IEEE Transaction on Dependable and Secure Computing*, vol. 6, no. 4, Dec. 2009, pp 282 – 294.

[24] Nikoloas Minas et al, "FPGA Implementation of an Asynchronous Processor with Online and Offline Testing Capabilities," in *14th IEEE International Symposium on Asynchronous Circuits and Systems*, 2008, pp 128 – 137.

[25] J. L. Yang, et al, "A Case Study on Asynchronous VLSI Design Platform," in *IEEE Conference on Electron Devices and Solid state Circuits*, 2007, pp 1167 – 1170.

[26] ViranthaEkanayake et al, "An Ultra Low Power Processor for Sensor Network," in Proceedings of 11th International Conference on Architectural Support for Programming Languages & Operating systems, 2004, pp 27 – 36.

[27] M. Marshall et al, "A Low Power Information Redundant Concurrent Error Detecting Asynchronous Processor," in 10th*Euromicro Conference on Digital System Design Architecture, Methods and Tools*, 2007.

[28] Martin Simlastic, et al, "Clockless Implementation of LEON2 for Low Power Applications," in *IEEE Conference on Design and Diagnostics of Electronic Circuits and Systems*, 2007, pp 1 - 4.

[29] H. R. Gerber, et al, "Design of an Asynchronous Microprocessor using RSFQ-AT," in *IEEE Transaction on Applied Superconductivity*, vol. 17, no. 2, June 2007, pp 490 – 493.

[30] HarriLampinen et al, "Design of Scalable Asynchronous Dataflow Processor," in *IEEE Conference on Design and Diagnostics of Electronic Circuits and Systems*, 2006, pp 85 – 86.

[31] JianGao et al, "A Novel Asynchronous Multiply Function Multiply Accumulator," in *6th IEEE International Conference on ASICs*, 2005, pp 223 – 226.

[32] L. Necchi et al, "An Ultra Low Energy Asynchronous Processor For Wireless Sensor Networks," in *12th IEEE International Symposium on Asynchronous Circuits and Systems*, 2006, pp 1 - 8.

[33] X. Lao et al, "Low Overhead Asynchronous RISC Microprocessor A Design Experiment," in *2nd Annual IEEE Northeast Workshop on Circuits and Systems*, 2004, pp 377 – 380.

[34] P. D. Hyde et al, "A Comparative Study of the Design of Synchronous and Asynchronous Self Checking RISC Processor," in *10th IEEE International Online Testing Symposium*, 2004.

[35] Y. W. Li et al, "High Throughput Asynchronous Datapath with Software Controlled Voltage Scaling," in *IEEE Journal of Solid State Circuits*, vol. 39, no. 4, April 2004, pp 704 – 709.

[36] V. S. P. Rakapa et al, "A Mixed Clock Issue Queue Design for Gobally Asynchronous Locally Synchronous Processor Cores," in *International Symposium on Low Power Electronics and Design*, Aug. 2003, pp 372 – 377.

[37] J. H. Lee et al, "A Novel Asynchronous Pipeline Architecture for CISC Type Embedded Controller, A8051," in *45th Midwest Symposium on Circuits and Systems*, 2002, vol. 2, pp 675 – 679.

[38] Puah W. B. et al, "Rapid Prototyping Asynchronous Processor," in *IEEE International Conference on Semiconductor Electronics*, 2000, pp 223 – 228.

[39] K. S. Stevens et al, "An Asynchronous Instruction Length Decoder," in *IEEE Journal of Solid state Circuits*, vol. 36, no. 2, Feb. 2001, pp 217 – 228.

[40] S. B. Furber et al, "AMULET 2e An Asynchronous Embedded Controller," in *Proceedings of the IEEE*, vol. 87, no. 2, Feb. 1999, pp 243 – 256.

[41] Clive Max Maxfield, "The Design Warrior's Guide to FPGAs," *Elsevier Publication*, 2006.

[42] S. M. Kang et al, "CMOS Digital Integrated Circuits," *Tata McGraw Hill Edition*, 3rd Edition, 2003.

[43] D. L. Perry, "VHDL Programming by Examples," *Tata McGraw Hill Edition*, 4th Edition, 2002.

[44] J. Bhasker, "A VHDL Primer," *Pearson Education*, 3rd Edition, 2000.

[45] J. Bhasker, "Synthesis Primer," *Pearson Education*, 2001.

[46] R. P. Jain, "Modern Digital Electronics," *Tata McGraw Hill Edition*, 4th Edition, 2010.

[47] Stephen Brown, “Fundamentals of Digital Logic Design with VHDL, *Tata McGraw Hill Edition*, 2nd Edition, 2007.

[48] Neil H. E. Weste et al, “CMOS VLSI Design,” *Pearson Education*, 3rd Edition, 2006.

[49] J. M. Yarbrough, “Digital Logic Applications and Design,” *Thomson Learning*, 2001.

[50] D. P. Leach et al, “Digital Principles and Applications,” *Tata McGraw Hill Edition*, 5th Edition, 2002.

[51] R. J. Baker et al, “CMOS Circuit Design, Layout and Simulation,” *IEEE Press Series on Microelectronic System*, 1998.

[52] J. M. Rabaey et al, “Digital Integrated Circuits A Design Perspective,” *Tata McGraw Hill Edition*, 2nd Edition, 2002.

[53] Holger Karl et al, “Protocols and Architecture for Wireless Sensor Networks,” *John Wiley and sons*, 2005.

[54] K. A. Fawaz et al, “A Dynamically Reconfigurable Asynchronous Processor,” in *IEEE Symposium on Application Specific Processor*, 2010, pp 93 – 96.

[55] Taylor S et al, “Asynchronous Data Driven Circuit Synthesis,” in *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, 2010, pp 1093 – 1106.

[56] Rossi D et al, “A Heterogenous Digital Signal Processor for Dynamically Reconfigurable Computing, in *IEEE Journal of Solid State Circuits*, vol. 45, no. 8, 2010, pp 1615 – 1626.