A Custom Processor for Node and Power Management of a Battery-less Body Sensor Node in 130nm CMOS

Y. Shakhsheer[1], Y. Zhang[1], B. Otis[2], and B. H. Calhoun[1]
[1]Dept. of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, USA,
[2]Dept. of Electrical Engineering, University of Washington, Seattle, WA, USA,

Abstract – We present a processor, the Digital Power Manager (DPM), providing power management and node/data/processing flow control for a 130nm battery-less power harvesting body sensor node. The DPM adjusts node power consumption, responding to available energy to support operation exclusively from harvested power. The DPM consumes 2.5pJ/instruction and 0.63pJ/cycle for NOPs.

I. Introduction and Motivation

Body sensors networks (BSNs) show great promise for long term, comprehensive, inexpensive, and unobtrusive monitoring of patients and healthy individuals. However, they require careful power management due to BSN node's size, weight, lifetime, and cost requirements. The desire for low cost and a small, wearable form factor limits the amount of local energy storage, which is either a battery or a super capacitor. Harvesting ambient energy, such as from the sun, vibrations or thermal gradients due to body heat, is an appealing alternative to battery based operation, allowing for an indefinite node lifetime while achieving the required small form factor.

Recent implemented BSN nodes have shown the ability to utilize harvested energy [1][2]. To sustain operation solely from harvesting, the circuits in the node must consume less power than the power harvesting mechanism produces or temporarily store energy for future use by higher energy operations [3]. There are several issues with using harvested energy. High peak current levels cause significant voltage drops in high-impedance power harvesting sources. Additionally, high power operations, such as transmitting data wirelessly, can consume 100s of µWs [4] and may exceed the power budget set by power harvesting. Lastly, power harvesters' power output is highly environment-dependent. For example, solar harvesting provides reduced power in cloudy conditions. A power management system requires a power harvesting-specific power management scheme which must account for all these issues to prevent node death (node voltage supply falls below the voltage at which the analog or digital circuits can function). It must adjust node power consumption to the changing power harvesting input to ensure functionality.

Current BSN nodes use a generic microcontroller (MCU) [2] for a variety of functions. The MCU is typically responsible for all on-node processing, power management, and memory accesses. Additionally, MCUs use interrupts to implement data sampling and execute any kind of power management scheme.

Use of a generic MCU to control an power harvesting BSN node is inefficient. Entrusting a large amount of responsibilities to the MCU, such as power management, node control, and data flow, requires overclocking the processor or running at a higher voltage and, thus, consumes more energy. The MCU requires multiple instructions to run common node tasks, such as branch commands and control commands, requiring extra clocking energy. Lastly, in BSN applications, the MCU may spend the majority of its life time idling. Generic MCUs are incapable of running NOPs efficiently. An ideal controller provides the flexibility of a generic MCU and energy efficiency.

To address these limitations and minimize the energy consumption, we designed a custom processor, called the Digital Power Manager (DPM), that is responsible for power management, node control, data flow management, and overseeing all processing on the battery-less BSN node presented in [1]. To our knowledge, the DPM is the first implemented on-chip power management system for a fully power harvesting node. This paper is divided up as follows. Section II details the architecture of the BSN node. Section III discusses the node management role and custom instruction set architecture (ISA). Section IV discusses the power management features. Section V concludes the discussion.

II. Chip Architecture & DPM Role

A wireless BSN node 130nm system-on-chip (SoC) that provides real-time ECG (electrocardiogram), EEG (electroencephalography), and EMG (electromyography) waveform acquisition, signal analysis and processing, and wireless communication over a MICS radio is presented in [1]. The node is battery-less, powered entirely from a wearable thermoelectric generator (TEG) with an off-chip storage cap and an on-chip boost converter.

The chip supports many application modes. For example, it can acquire an ECG, extract heart rate, and transmit it wirelessly to a basestation. In another mode, the chip acquires ECG, checks for atrial fibrillation (AFib, a cardiac arrhythmia), and transmits the ECG of the last 8 beats over the radio if

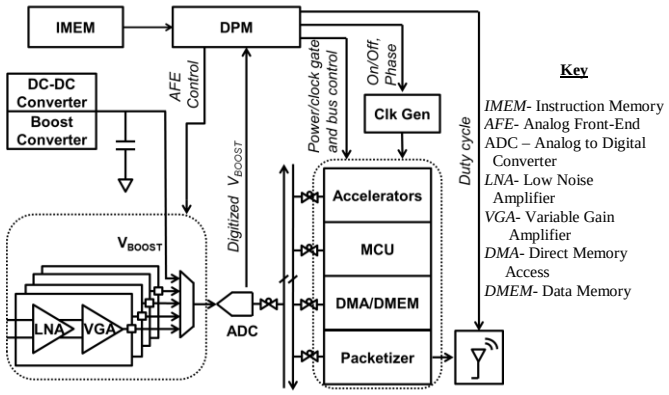

Fig. 1. Block diagram of [1]. The DPM is responsible for power management, node control, data flow management, and overseeing all on-node processing.

AFib is detected. In both modes, the SoC consumes 19µW and is powered solely by the TEG with an output of 30mV [1].

Fig. 1 shows the block diagram of the chip and the DPM's integral node management responsibilities. The DPM manages a four-channel analog front-end (AFE) for amplifying biosignals, capable of duty cycling it for saving. The DPM selects an analog channel to digitize to send to the subthreshold digital processing section for ECG/EEG/EMG processing. This section consists of accelerators and a low power subthreshold MCU based off the PIC 16C5X ISA [5], available for any generic processing needs. These accelerators include a programmable 4-channel, 30-tap FIR filter with programmable coefficient values, an accelerator for extracting the R-R interval from ECG waves, and for detecting AFib, a programmable envelope detector extracts peak information. Accelerator results can be stored in and retrieved from the 4kB data SRAM (DMEM) through a direct memory access (DMA) interface. Data can be transmitted through a sub-mW 400/433 MHz MICS/ISM band transmitter. This DPM is capable of power and clock gating each block, enabling the lowest power solution. Data is transferred between blocks via two circuit switched buses, also controlled by the DPM.

The node is powered by a power harvesting/supply regulation section that boosts an input as low as 30mV up to a regulated 1.35V supply, varying as power harvesting conditions change. The DPM adjusts the node's power consumption in response to real time measurements of available harvested energy to support battery-less operation.

The DPM's numerous responsibilities will be discussed in depth in the next sections.

III. Node Management and Data Flow

The DPM is a light-weight, always-on custom MCU. The DPM executes arbitrary instructions from a 1.5kB instruction memory. The memory is divided into a ROM that holds a hardcoded DPM AFib detection program and a RAM that is programmable for both DPM and MCU instructions. Control bits to each block are held in 95 output registers within the DPM. These signals are buffered and routed to their respective blocks, not requiring individual decoders for each block. This provides energy savings over many MCU architectures. Generic MCUs are less efficient than the DPM and require 2-6 MCU instructions to run single-cycle DPM-equivalent instructions. Using generic MCUs require more memory accesses and clock cycles to run a DPM equivalent program.

A. DPM Instruction Set Architecture

The DPM decodes a 12b instruction word issued by the instruction memory. Memory words have two primary formats, as shown in Fig. 2. The first 3b are an opcode. In format 1, the next four bits serve as a block ID number. Block ID numbers are unique to each block. For example, the ID number 1011 refers to the AFE. The rest of the word is control bits specific to the block. This format allows easy extension of the ISA to support new blocks. Opcode 011 uses format 2, which effectively uses a 7b opcode. The other bits are literals.

Format 1

OP	OP	OP	ID	ID	ID	ID	C	C	C	C	C

Format 2

0	1	1	OP	OP	OP	OP	L	L	L	L	L

Fig. 2. DPM instruction formats

The DPM utilizes a custom ISA, shown in Table I, to provide energy efficiency and flexibility, enabling low power operation on this node. The following subsections will explain these instructions. The table also shows the number of instructions it would take to run an equivalent operation on the MCU [5], assuming control bits are held in an output register. For a simple instruction such as enable (EN) in which bits are being written to an output register, the MCU executes a MOVLW and MOVWF, moving control bits to a work register and, then, to the intended output register.

TABLE I: DPM ISA. ALL SINGLE CYCLE INSTRUCTIONS

Code	Description	MCU Eq
NOP	No operation	1
STALL	NOPs until new ADC sample available	1
TIMER	NOPs for a set number of cycles	1
EN	Enable/disables, sets voltage of, clock gating, and resets block specified in instruction.	2
DMA	Sets DMA control to read or write	2
ADCCHAN	Selects ADC channel input	2
CTRL	Sets flag for intended destination of instruction	2
SAVEPC	Stores current PC values	4
RESTOREPC	Restores PC value from the last saved PC value	4
SETVAR	Sets voltage of variable voltage	2
BUS1/BUS2	Controls connections to the respective bus	6
SETCLK	Sets clock to accelerators, DMA,	2
CJMP	Conditional jump based on amount of energy, amount of data stored, detection of AFib, or if the node has been programmed	4
JMP	Absolute jump to literal value	4

B. Boot Sequence

Power harvesting conditions may change at any time, potentially resulting in node death if the consumed energy exceeds the stored energy faster than harvesting replenishes it. If node death occurs, this BSN node is capable of being revived in the field through a short RF burst. In this case, the volatile instruction memory is invalid and instructions should be run from the ROM. On startup, after the storage capacitor reaches a sufficient voltage, the DPM checks a flag, which is set to 1 when programmed and 0 when rebooted. The DPM sets the program counter (PC) to point to either the instruction ROM or RAM based on the flag through use of the conditional jump (CJMP) instruction.

C. Block Control

The DPM is responsible for managing the AFE, MCU, DMEM, accelerator blocks, and transmitter through the EN instruction, allowing for fine-grained power control. The EN command is common in our BSN programs and provides an energy efficient alternative to the MCU equivalent. The DPM provides control signals to the block as show in Fig. 3. The DPM is capable of duty cycling and power gating power hungry blocks such as individual AFE channels, the transmitter, and its crystal oscillator, resulting in large energy savings.

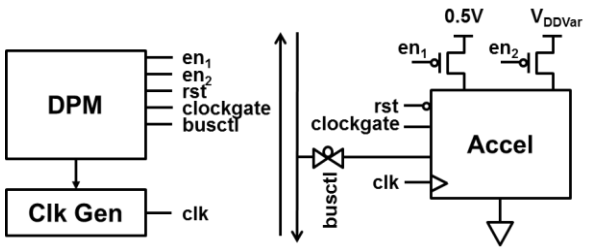

Fig. 3. Controls from the DPM to a generic accelerator block.

To save dynamic and leakage energy, the DPM can clock gate the whole data memory and individually power gate each bank by asserting bits to the NMOS footer on each bank. Individual accelerators can be clock gated and power gated. Additionally, accelerator blocks are capable of running on a 0.5V supply or a programmable voltage supply. The DPM is capable of setting the voltage of a programmable, variable DC-DC converter (V_{DDVar}) capable of providing 0.3V-1.2V by sending four control bits to the voltage regulator through the SETVAR instruction. V_{DDVar} is distributed to each accelerator, allowing blocks to utilize dynamic voltage scaling.

Most accelerators do not require fast clock speeds, such as the 200 kHz system clock, for processing. To reduce energy, the DPM issues control bits through the SETCLK instruction to the clock generator block to turn on the accelerator clock. The clock division is programmed within the scan chains.

D. NOP/Stalls

NOPs are important because the DPM spends much of its time idling since the processing is being done in the accelerators for common operating modes. This provides a good opportunity for saving energy. In NOPs, much of the DPM is clock gated, leaving only a simple timer/event block on to turn on the rest of the DPM. DPM NOPs provide ~4x savings over a DPM instruction and 2x savings over an MCU NOP.

Energy efficient NOPs are run through two DPM instructions, TIMER and STALL. For TIMER, the DPM counts down from one of 16 8b programmed-at-run-time values. STALL is an event based NOP. The node executes NOPs until the next value from the ADC is available, thus not requiring any memory accesses while the DPM awaits the next sample.

E. Flexible Datapath

BUS1 and BUS2 instructions provide flexibility to move data through the ADC, accelerators, the MCU, the DMEM, and the transmitter through executing DPM instructions. The node has two circuit switched buses that connect blocks, shown in Table II, to move data. The DPM issues bits to the transmission gates that connect each block's inputs and outputs to each bus. The DPM ensures that only two blocks are connected on each bus at one time with a hardware lock.

TABLE II: BUS CONNECTIONS FOR BOTH BUSES

Available Inputs	Available Output Destinations
MCU	MCU
FIR (x4)	FIR
DMA / Data Memory	DMA/Data Memory
Envelope Detector	Envelope Detector
ADC	Packetizer
	R-R Extraction/AFib Detection

F. Generic Processing

The DPM is not capable of generic arithmetic processing. Low power processing is done within accelerators. Processing that cannot be done in accelerators is done within the MCU, providing the node with more processing flexibility.

DPM and MCU instructions are stored in the instruction memory. These two blocks have unique ISAs, but both utilize 12b instruction words. Fig. 4 shows the method the DPM uses to determine which block should execute the instruction by checking an internal instruction flag. The flag is toggled by the DPM CTRL instruction. If the flag is 0, the DPM receives and executes the instructions and the MCU is clock gated, retaining its state if it is on. If the flag is 1, the MCU receives and executes the instruction while the DPM executes a NOP instruction. The DPM retains its state and, therefore, retains accelerator block states so data can be processed in parallel.

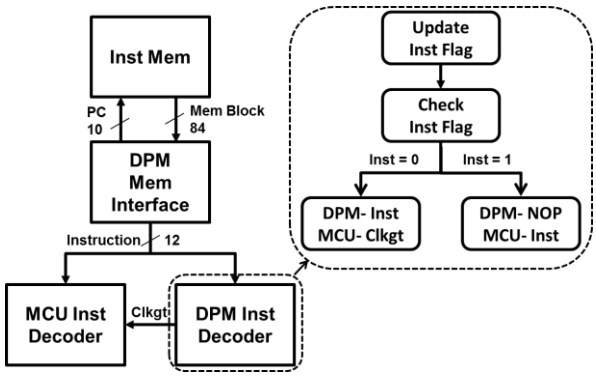

Fig. 4. The DPM determines which block should execute the instruction. The DPM decoder checks the instruction flag and makes the appropriate decision.

G. Measured Energy

The DPM provides many delay and energy advantages over the MCU. The DPM measured energy per operation on the 130nm chip at 0.5V is 2.74pJ. The DPM NOP is 0.68pJ/cycle at 0.5V. Fig. 4 shows a comparison of measured energy-delay curves for the DPM and the MCU for several instructions. Though the MCU's single cycle energy is 1.45pJ at 0.5V, the MCU requires at least two instructions to run DPM equivalent operations (see Table I). The MCU requires six instructions to produce a DPM equivalent BUS1 instruction. The DPM energy measurement includes the overhead of interfacing to the memory and the power management.

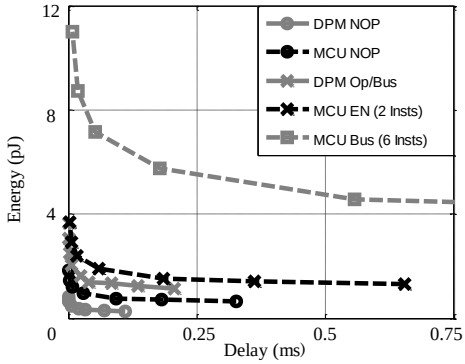

Fig. 4. Measured energy-delay curve for the DPM/MCU for NOPs, ENs, and BUS instructions.

IV. Power Management

One of the DPM's main purposes is to ensure the node remains alive by making intelligent processing and transmitting decisions based on the amount of energy on the capacitor in addition to its other responsibilities. Node death causes corruption of memory, resulting in permanent loss of critical health data stored in the DMEM and loss of non-default functionality by corruption of the instruction memory RAM.

The DPM checks the voltage on the storage capacitor, which corresponds to the energy available, and selects and implements an operating mode that limits the maximum amount of node power consumption for the given amount of energy available. If conditions are poor, the DPM throttles down processing and/or turns off transmission to save power and preserve the node; likewise, if conditions are favorable, the DPM can allow more blocks to be utilized. Turning off blocks occurs within one clock cycle. The MCU requires at least six cycles to implement the equivalent scheme, allowing the node to consume extra power when extremely limited.

A. Capacitor Voltage

The DPM utilizes the voltage on the storage capacitor to check the energy and make appropriate power management decisions. This stored energy changes with varying node power consumption and power harvesting rate. The capacitor voltage is digitized by the 8b ADC, which also digitizes biosignals. The 8b value is fed directly into the DPM immediately when a new energy value is available.

The capacitor value is only digitized when the ADCCHAN instruction is issued, selecting the capacitor voltage input to the ADC. This provides the programmer with the flexibility to bypass any power management restriction if transmitting or processing the data is more important than keeping the node alive and thus, needs to bypass the DPM policies (i.e. the node detected AFib and needs to transmit the ECG to notify the doctor over the wireless radio, regardless of energy status).

B. Stoplight

The DPM has three hard-coded operating modes (red, yellow, green). An operating mode is a mode in which only a hardcoded subset of blocks (Table III) is allowed to be turned on to process or transmit data, thus capping the maximum power consumption for the mode. Note that operating modes do not turn blocks on. The EN instruction must be issued to turn on blocks that are permitted for a given mode.

The DPM stoplight issues bits that can prevent mode-restricted blocks from running instructions (Fig. 5).

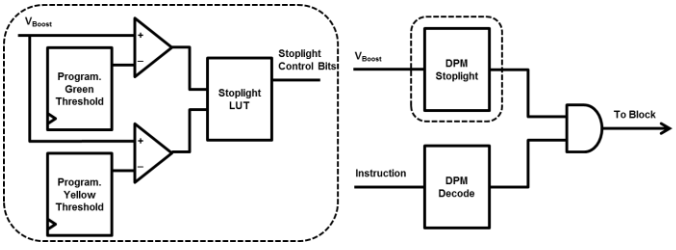

Fig. 5. Override structure of the DPM stoplight. The stoplight compares the V_{Boost} value to the threshold, selects the operating mode, and outputs control bits to the chip.

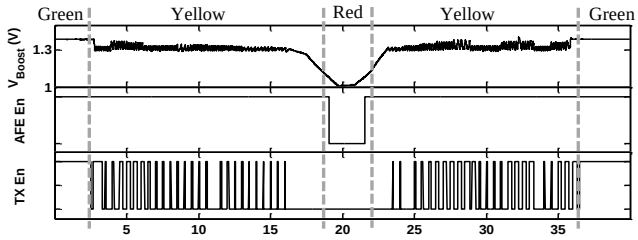

Fig. 6. Measured DPM closed-loop power management response. AFE enable (AFE En) and the transmit enable (TX En) are adjusted as the DPM changes modes.

TABLE III: OPERATING MODES

	IMEM	AFE	DMEM	Accel	Transmit
Red	On	Off	Off	Off	Off
Yellow	On	On	On	On	Duty Cycle
Green	On	On	On	On	On

This allows the DPM to modify power consumption within one clock cycle. The DPM's stoplight has the ability to power gate or clock gate individual node blocks, such as the transmitter, accelerator blocks, and analog front end channels to adjust power consumption. Fig. 6 shows the DPM stoplight overriding the previous state of the node as the capacitor voltage varies, limiting node energy consumption.

C. Threshold Values

Threshold values, programmed through the scan chains at run time, are used to determine the operating mode. The DPM compares the digitized capacitor value to two 8b threshold values (green and yellow threshold) and sets the operating mode. The current operating mode updates immediately when the stored voltage crosses a threshold. The DPM is capable of jumping from the current mode to any mode (i.e. green to red mode) or staying in its current mode at a threshold change.

IV. Conclusion

This paper presents the DPM, which is responsible for power management, node management, data flow management, and overseeing all processing on a power harvesting, battery-less BSN node. The DPM provides energy savings over generic MCUs. The DPM provides an autonomous, flexible power manager to account for varying levels of power harvesting in power harvesting BSNs. These DPM features allow this node to achieve energy efficiency in processing ECG, EMG, and EEG signals while functioning reliably from harvested energy.

References

[1] F. Zhang et al., "A Battery-less 19μW MICS/ISM-Band Energy Harvesting Body Area Sensor Node SoC," *ISSCC*, pp. 298-299, Feb. 2012.

[2] G. Chen et al., "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," *ISSCC*, pp. 288-289, Feb. 2010.

[3] B.H. Calhoun et al., "System Design Principles Combining Sub-threshold Circuits and Architectures with Energy Scavenging Mechanisms", *ISCAS*, pp. 269-272, 2010.

[4] S. Rai et al., "A 500μW neural tag with 2μVrms AFE and frequency-multiplying MICS/ISM FSK transmitter," *ISSCC*, pp. 212-213, Feb. 2009.

[5] S. Jocke, et al., "A 2.6-μW Sub-threshold Mixed-signal ECG SoC", *Symposium on VLSI Circuits*, 2009.