A Generic Multielement Microsystem for Portable Wireless Applications

ANDREW MASON, NAVID YAZDI, ABHIJEET V. CHAVAN,
KHALIL NAJAFI, SENIOR MEMBER, IEEE, AND KENSALL D. WISE, FELLOW, IEEE

Invited Paper

An open-architecture microsystem that can be populated with a variety of sensors and actuators is described. The microsystem is designed for low-power wireless applications where small size and high sensor accuracy are important. It consists of an in-module microcontroller connected to multiple front-end transducers through an intramodule sensor bus. An external interface allows internally processed data to be output through either a hard-wired input/output port or a radio-frequency transmitter. The present microsystem is configured for environmental monitoring and measures temperature, barometric pressure, relative humidity, and acceleration/vibration. It occupies less than 10 cc, consumes an average of 530 μW from 6 V, and transmits data up to 50 m. System features such as active power management, the intramodule sensor bus, generic bus interface circuitry, and in-module sensor compensation based on bivariate polynomials are discussed.

Keywords*—Compensation, measurement system, microelectromechanical devices, microelectromechanical system, microsensor, microsystem, power management, transducer.*

I. INTRODUCTION

Driven by rapid advances in microcomputers and global connectivity, many of the most important emerging markets for microelectronics require the ability to gather information from the nonelectronic world [1], [2]. Examples include health care (diagnostic and therapeutic devices, prosthetics), automotive systems (smart vehicles and smart highways), automated manufacturing [including smart very-large-scale-integration (VLSI) process tools], environmental monitoring and control devices, defense systems, and many consumer products. Using integrated circuit technology and extensions of it, integrated sensors and microactuators are being developed to provide the necessary input/output (I/O) devices. These are increasingly being realized combined with hybrid or monolithic circuitry on a common substrate [3]–[6] and have come to be known as microelectromechanical systems (MEMS). Merging these devices with increasingly powerful digital signal-processing electronics now makes it possible to go beyond simple analog readout circuitry and form complete closed-loop microsystems in very small, highly integrated modules. These autonomous microsystems are capable of gathering data from the physical world, converting them to electronic form, compensating them for interfering variables and nonlinearities, and either acting on the information directly or transferring it to other systems [7]–[9]. Such microsystems can be expected to have a significant and pervasive impact on a large number of applications during the coming decade.

As used in this paper, a "microsystem" is defined as

> a collection of highly integrated devices that contains transducers along with appropriate interface circuitry and is capable of performing multiple tasks autonomously as well as responding intelligently to various commands from a host system.

This definition helps to identify the class of devices being considered and illustrates the differences between a microsystem and a less complex "smart" sensor. Reflecting general trends in microelectronics, many of the emerging applications for microsystems demand small portable wireless devices [10]–[12] having high accuracy and the ability to perform in distributed instrumentation systems collecting data over a broad physical area [13]. It is significant that in applications as diverse as personal health monitors, implantable neural prostheses, sensors for VLSI process control, sentries for battlefield awareness, and sensors for unmanned air reconnaissance, a single generic open system architecture applies. Such systems are customized by the sensors and actuators populating them and by the software used in their resident microcontrollers. While not every application shares the same power or telemetry range requirements, a common system framework can be shared and provides a basis for defining the standards that will be

Manuscript received November 24, 1997; revised January 14, 1998. This work was supported by the Defense Advanced Research Project Agency under Contract J-FBI-92-149.

The authors are with the Center for Integrated Sensors and Circuits, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2122 USA.

Publisher Item Identifier S 0018-9219(98)05096-8.

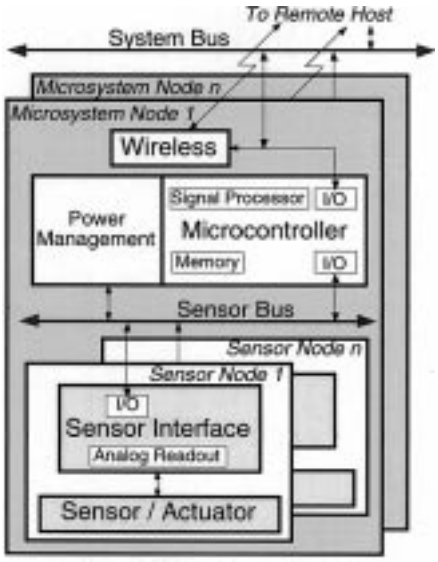


Fig. 1. Component-level architecture for a multinode microsystem formed as part of a larger distributed instrumentation system.

required in order for these systems to fully evolve [14], [15].

The eventual realization of many such microsystems will be as small (1 cc), portable, low-power (<1 mW), highly accurate sensing modules capable of bidirectional communication over considerable distances (1–2 km). While not yet meeting all of these objectives, the microsystem reported in this paper represents a significant step in this evolution. It illustrates a generic architecture for such devices along with operational features such as in-module data compensation and the use of self-testing at the transducer level. The resulting microinstrumentation system ("μCluster") is configured for environmental monitoring [9], including use as a microweather station for local, regional, and global weather forecasting.

II. Generic Microsystem Architecture

The open architecture shown in Fig. 1 partitions each individual microsystem so it can be populated to fit the requirements of different applications without limiting the transducer technologies that can be accommodated. Each microsystem can be used as part of a distributed network of such devices connected through a system bus to a remote host. This external bus may be either hard wired or wireless. Within each microsystem node is signal-processing electronics that operates under stored program control to perform preset tasks and respond to commands received from a host system. In-module memory stores the control programs as well as sensor-specific code that describes the operation of each front-end device. The control electronics connects to the transducer front end through an intramodule sensor bus, which allows read and write instructions to be issued to the transducers. Each of the front-end sensor/actuator chips contains an interface to this bus along with the transducers and their associated readout and/or control circuitry. A microcontroller provides in-module signal processing and memory, as well as power-management circuitry and the I/O hardware for both the external system bus and the intramodule sensor bus.

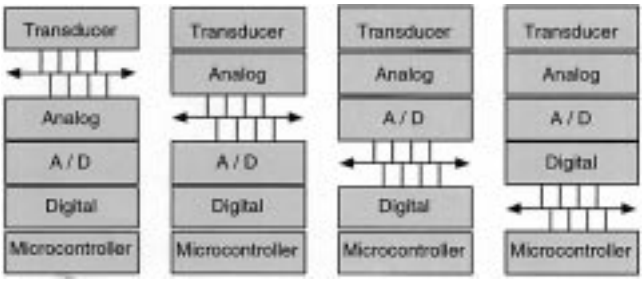


Fig. 2. Possible partitioning of a microsystem with multiple sensing nodes. In (a), we have a simple transducer hard wired to the control block. In (b), this becomes an integrated sensor (with electronics), while in (c) and (d), we evolve toward a "smart" front-end sensor.

Several bus-compatible "smart" sensors (similar to the front-end sensors here) have been reported for use as microsystems [5], [16]. However, many emerging sensors measure multiple parameters and are programmable, electronically trimmed, and self-testing. For these devices, as well as for systems containing multiple sensing chips, it is very attractive to utilize an intramodule sensor bus along with an embedded microsystem controller [17]. The microcontroller performs in-module signal processing and permits the microsystem to respond to commands from a host. It also compensates the sensor data and can make in-module decisions based on this data to offload the host system [12]. The microsystem can be programmed with more or less knowledge of the overall system task being performed, depending on the application.

For an open module architecture with a "plug and play" approach to front-end transducer population, a standard interface between the microcontroller and front-end devices is necessary to allow different sensor manufacturers to design to a common system specification. Whether the sensor is implemented as a single chip or as a two-chip hybrid [1], it is necessary to have a standard sensor bus connecting the controller to the sensing nodes. Fig. 2 illustrates the signal path for the sensor data and shows the possible partitions to accommodate a sensor bus that divides the front end from the controller. With each option, the signal path consists of the transducer itself, analog signal conditioning (amplification) circuitry, analog-to-digital conversion (ADC), digital signal processing, and the microsystem controller. In Fig. 2(a), the bus receives the direct transducer output while all signal-processing circuitry is in the control block. This option is not appropriate because of the difficulties in busing unbuffered sensor signals (representing attofarad or femtofarad capacitance changes) and because, in order to maintain an open architecture, it would require many different analog readout circuits to be built into the control circuitry. In options (b), (c), and (d), the difference is the amount of signal processing

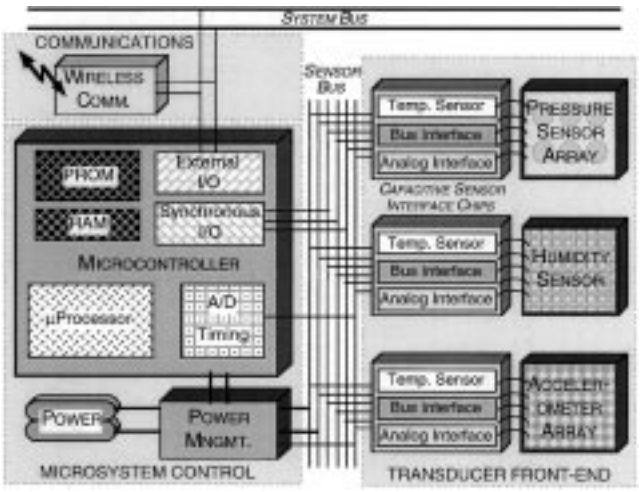


Fig. 3. Block diagram of the μCluster, a low-power wireless microsystem for portable environmental monitoring applications.

done at the front end. Many microcontrollers available today have built-in ADC's and digital signal-processing capabilities, so that option (b) is a good choice that shares available resources. However, a truly open architecture should also accommodate devices matching options (c) and (d) since these do not force additional requirements on the microcontroller. Thus, using a microcontroller with a built-in ADC, the architecture shown in Fig. 1 represents a very generic structure accommodating a variety of sensor nodes. Any front-end device with the appropriate bus interface circuitry can be added to such a microsystem, providing the desired "plug and play" capabilities.

III. Microsystem Components

The generic architecture introduced above defines a basic set of control, communication, and front-end components that will now be discussed in more detail to illustrate issues important to microsystem development. These issues will be presented as they relate to the prototype μCluster [9]. This microinstrumentation system was designed to demonstrate how the generic architecture could be used to build a microsystem that is small in size, dissipates very low power, utilizes wireless communication, and provides highly accurate, fully compensated sensor data. As shown in Fig. 3, the μCluster consists of external interfaces for both hard-wired and wireless communication, a control block containing a microcontroller and power-management circuitry, and several front-end sensing nodes. The control block is connected to the sensors through an intramodule sensor bus and sensor interface chips. The sensors, measuring temperature, barometric pressure, relative humidity, and acceleration/vibration, are all capacitive, so that the associated power dissipation is limited to the analog readout circuits used; all of these transducers are self-testing using embedded thermal or electrostatic actuation. Operating under stored program control, the μCluster performs periodic scans of these embedded transducers. The scan rate is programmable and can be adaptive, based on the rate of change in the sensor readings. The sensor data are compensated by electronically programming the front-end readout circuitry (for coarse trims of offset and slope) and in software using prestored compensation algorithms. The resulting measurements are stored internally until output to a remote host system. Between scans, the μCluster utilizes a low-power "sleep mode" to maximize battery life, but the sensor bus is still monitored for event-triggered interrupts that can wake the system.

A. The Microcontroller

The microcontroller unit (MCU) used in such microsystems should provide a microprocessor for signal processing and stored-program control, I/O interfaces to both the external system bus and the intramodule sensor bus, and an on-chip ADC and timing hardware to accommodate different sensor data formats. Adequate on-chip read-only memory (ROM), random-access memory (RAM), and electrically erasable and programmable (EEP)ROM are important, as well as overall power dissipation, which is important in spite of the relatively low duty cycles in many applications. For the present μCluster, the Motorola 68HC11 is used. It has an 8-b processor, an RS-232 compatible UART-type interface for external I/O, a synchronous serial peripheral interface adapted for sensor bus communications, an 8-b ADC for converting analog sensor data, signal timing hardware for frequency-encoded sensor data, a variety of on-chip memory, and a low-power mode that maintains RAM data while drawing less than 50 μA [18]. The active power dissipation is about 90 mW with a 2-MHz bus clock. EPROM (96 kb) is used to store control and sensor-specific program code, EEPROM (4 kb) is used to store operation and compensation data for individual sensors, and RAM (4 kb) is used for the temporary storage of sensor data and program variables.

The 68HC11 was not specifically designed for these microsystem applications; indeed, few manufacturers have so far addressed this area in spite of the fact that it promises to be a very high-volume market. A microcontroller designed specifically for such microsystems should have all of the basic subsystems noted above but should minimize hardware, including I/O ports, to keep the power consumption low ($<$10 mW active; $<$20 μW standby). Since a direct tradeoff between clock speed and power dissipation exists in the digital electronics, and because the readout time for many transducers is on the order of 100 μs or more (during which the processor will generally be on), a processor speed of 5–15 MHz should be adequate. The controller should also have an ADC with $\geq$12-b accuracy; for some applications, 14–16-b accuracy is needed. Hence, a 16-b processor is also desirable. Low-power, low-voltage circuit techniques [19] should be used along with the smallest possible device dimensions to help reduce load capacitances. The ability to shut down MCU subsystems selectively when not in use can further reduce power dissipation. Last, the ability to programmably control clock frequencies would allow transducer front-end operations to occur more slowly while processing data at the fastest possible rate. Although some of these specifications

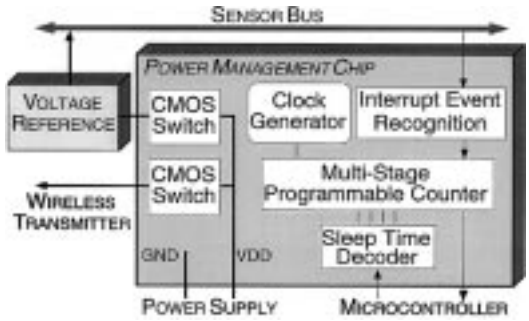


Fig. 4. Block diagram of the power-management chip used to control power flow across the μCluster and monitor system activities during low-power sleep mode.

are perhaps aggressive, the technology available today, if applied to a microsystem-specific controller, could achieve them.

B. Power Management

In battery-powered microsystems, it is important to minimize power consumption wherever possible. In the transducer front end, capacitive sensors should be used since these devices offer high sensitivity yet consume no power and can be read out rapidly using low-power circuit techniques. For wireless operation, it is important to use low-power telemetry hardware, understanding the direct tradeoff between power and communication range. While many efforts are currently under way to reduce the power consumption of MCU's [20] and wireless communication devices [21], system-level approaches to power management are also important. A common method for conserving power that was adopted in the μCluster is to power down unused subsystems. Since the MCU here is periodically shut down, a separate power-management chip (PMC) was necessary. This all-digital complementary metal–oxide–semiconductor (CMOS) circuit can implement all of the necessary functions with less than 10 μA of supply current. As shown in Fig. 4, the PMC contains an on-chip clock generator and timing hardware that can measure eight discrete time periods between 15 s and 5 min based on a 4-b code input from the MCU. This allows the MCU to enter a low-power "sleep" mode for a preset duration, after which the PMC wakes the MCU. Because many of the parameters measured by the μCluster change slowly, most front-end nodes can also be powered down while not in use. These devices are powered through a 5 V reference available on the sensor bus. This voltage reference is controlled by the PMC and can be turned on and off by input codes sent from the MCU. Likewise, the RF transmitter can be shut off by commands from the MCU. Thus, the MCU, through the PMC, controls when and where power is available on the μCluster. For some front-end devices, however, it is important to monitor the environment continuously. Such sensor nodes must be very low in power and are driven by the main system supply. If an event is detected that requires attention while the MCU is asleep, an interrupt line on the sensor bus is used to wake the MCU through the PMC. A front-end device that uses this interrupt feature is discussed in a later section of this paper.

Power-management features will certainly be necessary in most low-power microsystems [22] and will normally be integrated directly within the microcontroller along with selectively activated voltage references, low-power timing circuitry, and interrupt recognition. To reduce power consumption further, a 3 V supply is desirable but will put demanding constraints on the analog circuits in the transducer front end. Due to the limited current available from long-life battery supplies, it is necessary to reduce the maximum active (peak) load by individually enabling different circuit blocks to perform desired tasks periodically. Advances in battery materials and packaging continue to provide cells with improved energy density, weight, and volume [23], yet further improvements will be necessary as portable systems increase in performance while shrinking in size. Efforts using MEMS-based micropower supplies [24] illustrate new approaches to meeting the challenges of these small, low-power microsystems.

C. Wireless Communications

Wireless communications eliminates the need for costly wired networks and allows for rapid and simple microsystem deployment. Bidirectional communication with a range of a mile or more is needed for many applications, yet with current technology, these features would require more space and power than are available in the μCluster. However, the portable revolution now taking place is driving wireless technology with similar demands so that devices compatible with these needs should soon be available [21]. Radio-frequency (RF) technology has moved from purely analog to a mix of analog and digital components, providing a more efficient use of bandwidth, lower power dissipation, and improved noise immunity [25]. Several technologies are currently being investigated for use in RF transceivers, including CMOS, Si bipolar, Si/SiGe heterojunction bipolar transistors (HBT's), and GaAs MESFET and HBT devices. The passive components (inductors, capacitors, filters, and resonators) used often determine the size and performance of such systems, and current research efforts in MEMS aimed at performing these functions mechanically [26] should make it possible to realize the entire transceiver monolithically, reducing size, weight, cost, and power. Low-power CMOS wireless systems have been introduced [27], and commercial products such as the pager wristwatch [28] are driving the technology in the same direction as the microsystems considered here.

For the μCluster, the twin goals of small size and low power had a direct impact on the achievable communication range. To minimize power consumption, wireless operation was limited to data output only, and the HX1005 transmitter from RFM, Inc., a commercially available component in a very small package, was selected. This device operates with a 315 MHz carrier frequency using amplitude-shift-keyed modulation with a 1 kHz bit rate. The device has an average

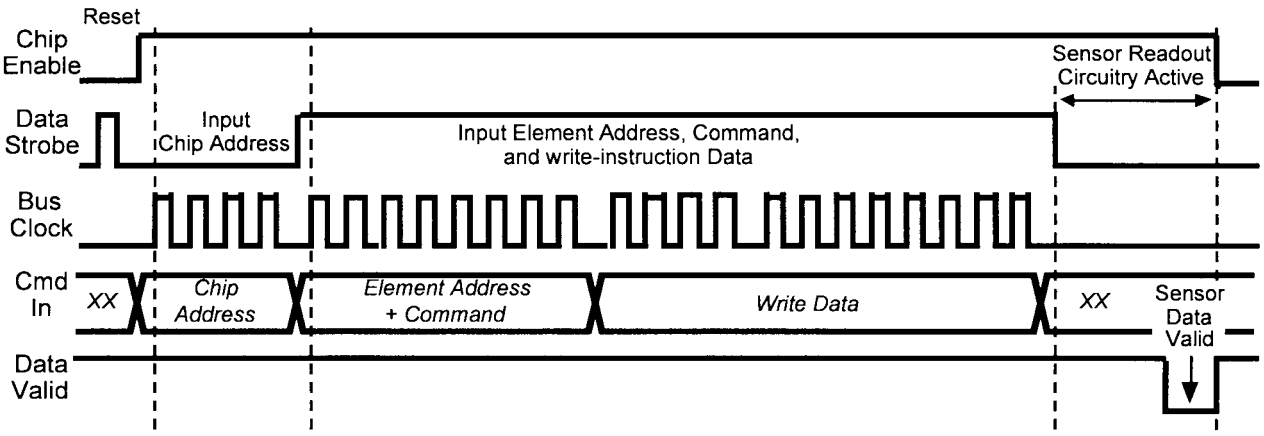


Fig. 5. Timing diagram for the serial communication signals on the microsystem sensor bus.

active power dissipation of 10.5 mW, with a range of about 50 m and a volume of 0.27 cc.

D. A Standard Sensor Bus

The sensor bus provides access to multiple front-end sensing nodes while using a limited number of interconnects. Each of these nodes can contain many addressable elements, including sensors, actuators, and programmable interface circuits. For the μCluster, a nine-line intramodule sensor bus has been adopted, although many other options are available [29]. This bus is based on the Michigan Serial Standard [30] and is very similar to the transducer independent interface (TII) of the recently developed IEEE-P1451 standard for sensor systems [31], [32]. This bus includes three power leads, four lines for synchronous serial communication, a data output line, and a data valid/interrupt signal. The three power leads are the system supply, ground reference, and a switched reference voltage for devices that can be powered down during a low-power system sleep mode. For synchronous communication with the front-end devices, a chip enable and data strobe provide handshaking for use with a programmable-frequency clock and a serial data line. The data-out (DO) line is used for sensor data output, which can be an analog voltage, a frequency-encoded pulse train, or a serial bitstream. The last line of the sensor bus is data valid (DV), which signals the MCU when valid data are present on the data-out line. Data valid is also used as an interrupt from the front-end devices that can initiate an event-triggered sensor readout when the bus is otherwise idle. A timing diagram illustrating the relationships among several of these signals is shown in Fig. 5.

As implemented in the μCluster, the data format for commands sent to the front-end nodes consists of a 4-b chip address followed by an 8-b instruction byte that contains a 3-b command and a 5-b element address. The 3-b command determines if the instruction is a sensor read request or one of several write commands. The 5-b element address is used to access multiple devices within each front-end node. In the case of a write command, data in 4- or 8-b packets will follow the 8-b instruction byte. Write commands can be used to write data to front-end actuators or to set control bits on programmable interface chips, as discussed below. In the case of a read command, the sensor bus clock can be used by front-end readout circuitry; the clock frequency is set by MCU software. When valid data are ready, the interface circuitry signals the controller through the DV bus line, placing the data on the shared DO line. Output to the DO line is disabled once the controller has received the sensor data.

IV. The Transducer Front End

The transducer front end of a microsystem consists of one or more sensing nodes. Each node contains sensors and/or actuators with appropriate readout circuitry to interface with the intramodule sensor bus. For open architectures, the front end can be populated based on the requirements of the selected application, allowing the same basic system to be used for a variety of applications. In the μCluster, each front-end sensing node has an assigned 4-b address code, allowing up to 16 such nodes to be accessed within the same microsystem. Each node can contain multiple sensors and/or actuators, limited only by the sensor bus instruction format. As described above, the data format used on the μCluster contains a 5-b element address that allows 32 readable elements and 32 writeable elements within each sensing node. Because the front-end transducers in the present μCluster are all capacitive, a generic interface chip can be used to connect all of these devices to the sensor bus. This generic interface chip and the individual sensors used for environmental monitoring in the μCluster are discussed below.

A. A Generic Capacitive Sensor Interface Chip

For environmental monitoring applications, the μCluster utilizes several capacitive sensors, which are attractive due to their low power, high sensitivity, and self-test capabilities. While numerous readout circuits for capacitive microsensors have been introduced [33]–[38], for multi-element microsystems such as the μCluster, a low-power generic interface that can accommodate the broad range of base capacitances and sensitivities associated with different transducers is needed. Such an interface has been devel-

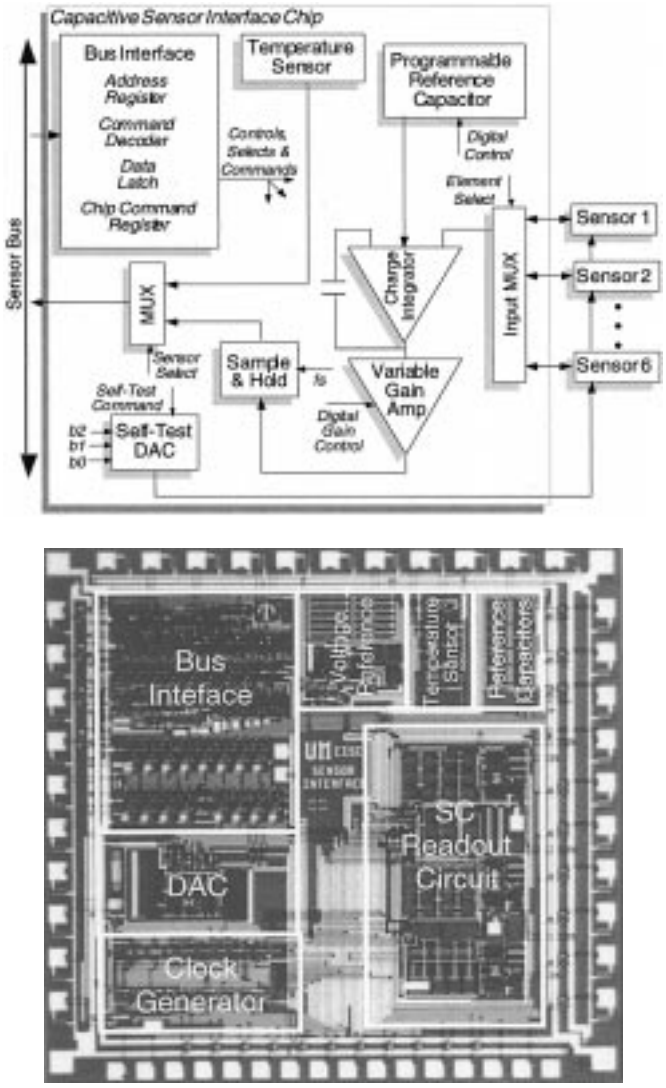


Fig. 6. A block diagram and die photograph of the generic capacitive sensor interface chip.

oped [39]. This sensor interface chip reads out capacitive devices very quickly, communicates with the microcontroller through the sensor bus, supports self-test and self-calibration, supports multiranging for a single sensor, and dissipates low power. The chip also includes a temperature sensor, as discussed below.

Fig. 6 shows a block diagram of the capacitive sensor interface chip. The serial data instructions that are transmitted from the MCU over the sensor bus are received, decoded, and stored by the bus interface unit and are applied to control the other circuit blocks illustrated in Fig. 6. Data written to the interface chip are stored in registers within the bus interface unit. To interface with capacitive sensors, this chip utilizes a low-noise front-end charge integrator to read out the difference between the sensor capacitance and a reference capacitor [34]. An input multiplexer allows the chip to interface with up to six sensor elements. Furthermore, the chip can be digitally programmed to operate with one of three reference capacitors, either external or internal. The on-chip reference capacitors are laser trimmable from 0.15 to 8 pF. Programmability of the reference capacitors allows the chip to interface with capacitive sensors having a wide range of base capacitance and also provides offset control.

The analog signal path consists of the input multiplexer, an input charge integrator, a gain stage, an output sample/hold circuit, and an output multiplexer. The gain stage can be programmed on-line to one of three gain settings. These gains can be used to accommodate sensors with different sensitivities or can be used for multiranging a single sensor, and can be fine-tuned by laser trimming the on-chip capacitors. The overall readout sensitivity can be varied from 0.23 to 73.5 mV/fF using both digital programming and laser trimming, providing an effective gain variation from 1 to 312. The output multiplexer controls access to the sensor bus for both the capacitive readout circuitry and the on-chip temperature sensor. Additionally, the interface chip supports self-test and self-calibration through a 3-b on-chip digital-to-analog converter that generates a variable-amplitude, two-phase clock for driving the sense and reference capacitors in each input charge integration cycle. The variable-amplitude clock can also be used to apply a varying electrostatic force to the sensor for self-test and self-calibration.

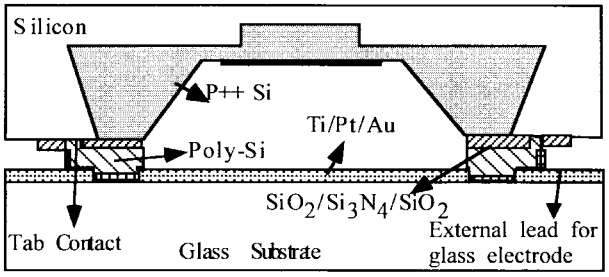
The interface chip has been fabricated using a standard 3 μm single-metal double-poly p-well CMOS process. Fig. 6 shows a photograph of the chip, which measures 3.2 $\times$ 3.2 mm. It dissipates less than 2.2 mW from a single 5 V supply and can resolve input capacitance variations of less than 1 fF with a readout time of 60 μs [39].

B. Temperature Sensor

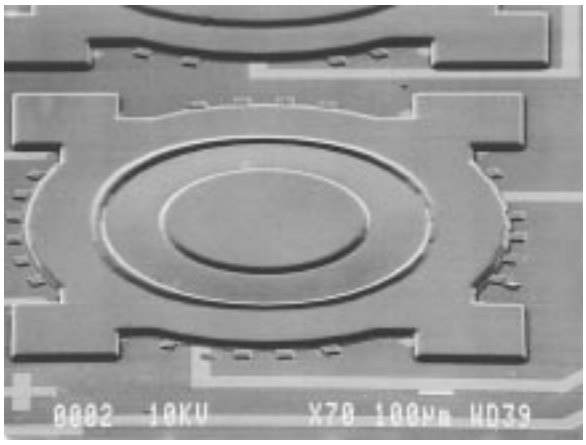
To measure temperature in close proximity to the other sensors in the microsystem, a temperature sensor has been integrated on the interface chip. This sensor provides data that can be used to compensate digitally for the temperature sensitivity of any other sensors connected through this chip. The temperature sensor utilizes the temperature dependence of the drain current of an MOS transistor in weak inversion [40]. The charging current for the capacitively loaded Schmitt input stage of a ring oscillator is set by a p-channel MOS transistor biased for subthreshold operation. Since this charging current is temperature dependent, the frequency of the oscillator provides a measure of the local temperature. A typical device displays a sensitivity of 4 ms/°C at 60°C and 33 ms/°C at -20°C with a resolution better than 0.5°C across the tested range. Although the sensor is highly nonlinear, it is easily calibrated using the digital techniques discussed later in this paper. While many integrated circuit temperature sensors exist [41], this technique provides a direct digital output and very low power dissipation, and can easily be implemented in a standard CMOS process.

C. Barometric Pressure Sensor

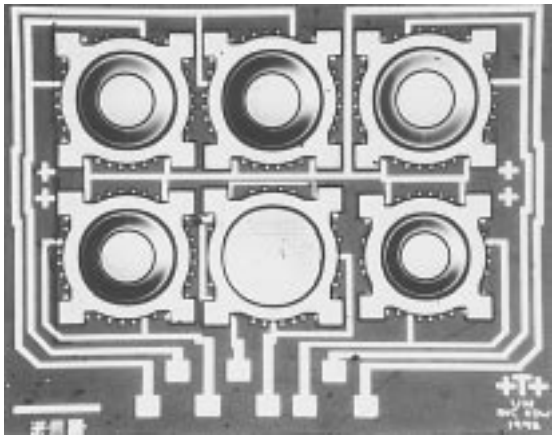
The major challenge in designing the barometric sensor is to achieve very high resolution over a wide dynamic range in both temperature (-20 to $+60$°C) and pressure



(a)



(b)



(c)

Fig. 7. A vacuum-sealed capacitive barometric pressure transducer. Shown are (a) a structural diagram, (b) a close-up scanning electron microscope (SEM) view of a single sensing element, and (c) a photograph of the complete five-transducer array with a built-in reference capacitor (bottom center).

(600–800 torr). The targeted resolution of 25 mtorr here is equivalent to about one foot of altitude shift at sea level and represents 15 b. To achieve wide dynamic range and high resolution simultaneously, a multitransducer vacuum-sealed capacitive pressure sensor has been developed [42], [43]. This device uses multiple diaphragms to segment the overall pressure range, as shown in Fig. 7. The bossed diaphragm diameters vary from 920 to 1100 μm at a thickness of 2.4 μm and a gap separation of 9.8 μm. The sensor is fabricated using bulk micromachining and a silicon-glass dissolved-wafer process [43]. To measure barometric pressure, the microcontroller first reads out the smallest of the diaphragms, which serves as a global sensor spanning the entire measurement range at relatively low resolution. After determining the approximate pressure, the controller then selects the appropriate segment transducer, each of which covers a range of about 50 torr. As pressure increases, the diaphragm deflection varies according to the fourth power of its diameter, causing the largest of the diaphragms to approach the glass first. As the gap approaches zero, a very high pressure sensitivity is achieved. When one diaphragm touches the glass, it is strain relieved, and the next diaphragm in sequence becomes the measurement device of interest. The transducer capacitance is read out using the switched-capacitor charge integrator on the interface chip. The typical pressure sensitivity of this device is 27 fF/torr (3000 ppm/torr) for a resolution of 25 mtorr with accuracy limited by nonlinearity and temperature sensitivity. Before compensation, the temperature coefficient of offset (TCO) of the transducer alone at 750 torr is 3900 ppm/°C, and the temperature coefficient of the pressure sensitivity is about 1000 ppm/°C. After compensation, the worst case TCO for the entire system is 40 ppm/°C.

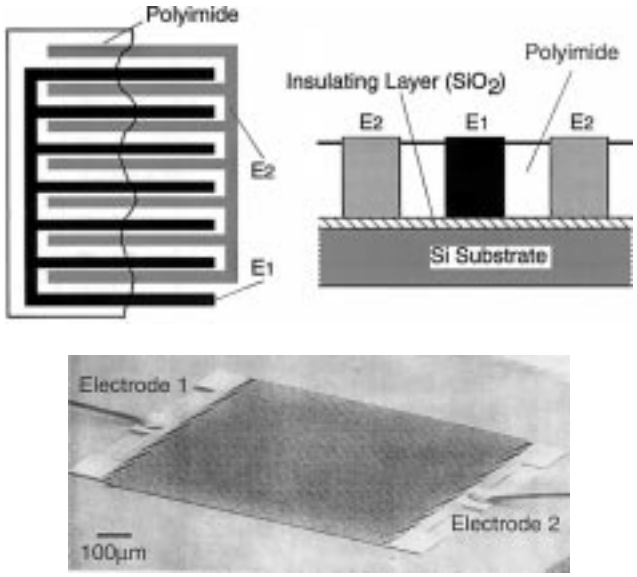


Fig. 8. Structural diagram and SEM photograph of the capacitive humidity sensor.

D. Relative Humidity Sensor

Another environmental parameter of interest is relative humidity (RH). A humidity sensor with small size (<0.5 cm^2), broad dynamic range (25–90% RH), high resolution ($\pm$2% RH), low temperature coefficient, and low power consumption is necessary to sense relative humidity changes accurately in many μCluster applications. By applying high-aspect-ratio micromolding [44] and electroplating technologies, a capacitive hygrometer with the structure shown in Fig. 8 has been fabricated. The spaces between a series of interdigitated electrodes are filled with a thin polymer film material. The dielectric constant of the polymer varies as a function of moisture, causing the

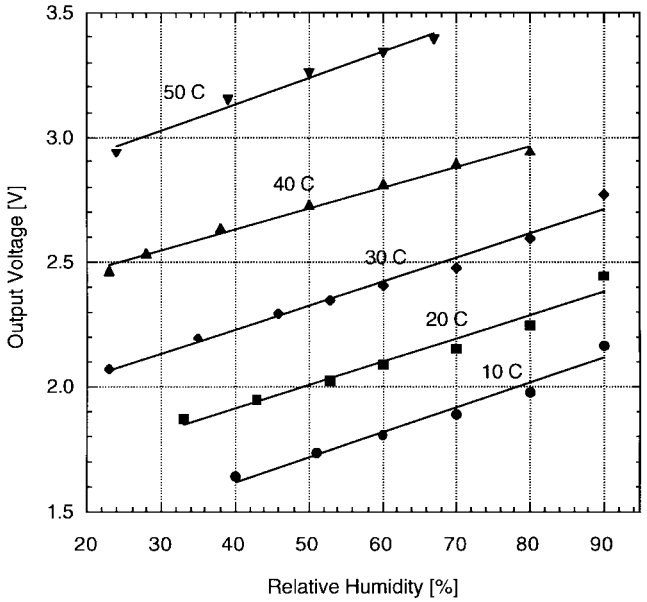


Fig. 9. Measured response of the capacitive humidity sensor at several temperatures.

capacitance between the two electrodes to change with ambient relative humidity. This capacitance change can be read out by the same interface circuit used with the other transducers on the μCluster. From a variety of humidity-sensitive polymer materials, the polyimide DuPont PI 2723 has been selected because its photosensitivity simplifies the fabrication process, while it provides good thermal stability, a linear response to humidity, and high-speed moisture absorption. Compared to other humidity sensors [45], this device utilizes high-aspect-ratio electrodes in order to obtain high sensitivity in a small area.

The hygrometer has been tested over a wide range of ambient relative humidity, temperature, and measurement frequency, and the results are shown in Fig. 9. This sensor, with a base capacitance around 6 pF, exhibits a sensitivity of 4.5 fF/%RH with good linearity over the desired humidity range. The response time is about 30 s, which is a limiting aspect of the current device. When used with the sensor interface circuit described above, the output voltage shows a linear response, a sensitivity of 9.6 mV/%RH, and a resolution of better than 2%RH. Although the polyimide alone has only a slight temperature dependence (20–50 ppm/°C), the overall sensor output shows a temperature sensitivity of approximately 30 mV/°C. This temperature sensitivity is due largely to the fact that at higher temperatures, there is more moisture in the air at the point of saturation than at lower temperatures with the same relative humidity. Thus, the capacitance of the hygrometer increases with temperature even when the humidity level is held constant. Although this effect limits the usable gain of the readout circuit, the shifts are predictable and can be digitally compensated. Current research efforts to improve this humidity sensor are focused on reducing the response time of the device and reducing moisture-induced charge leakage at the sensor-readout interface.

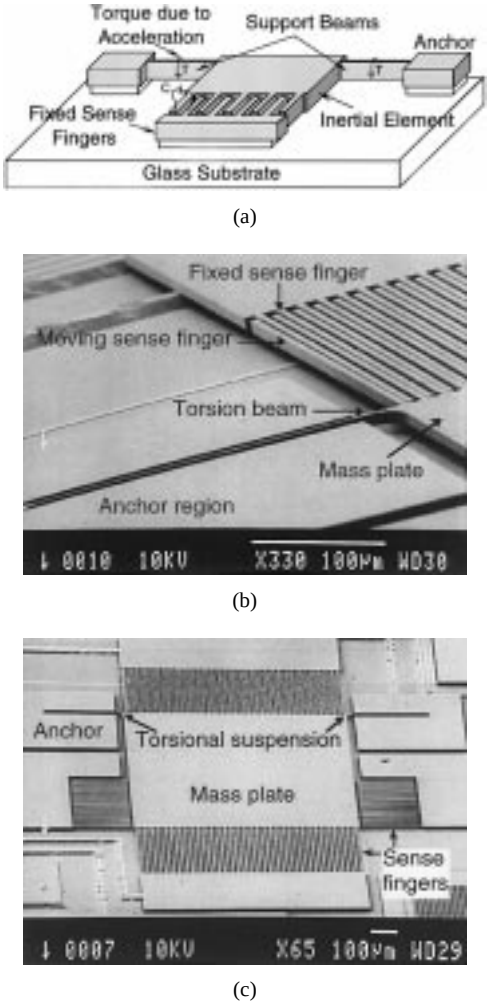


Fig. 10. (a) Structural diagram of a z-axis capacitive torsional accelerometer. (b) Close-up and (c) overall SEM views of the device are also shown.

E. Integrated Accelerometers

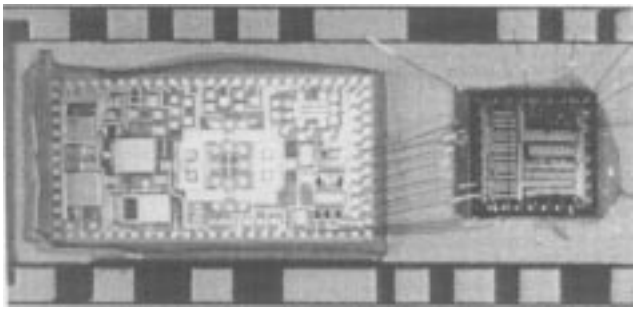
The μCluster incorporates sensors for the measurement of acceleration and/or vibration, as well as a low-power programmable threshold accelerometer array that provides an event-triggered wake-up call when the system is in sleep mode or an interrupt when it is in the normal mode of operation.

A linear torsional accelerometer is used for z-axis measurements [46]. This device, shown in Fig. 10, uses interdigitated electrode fingers and a capacitive overlap area that varies with acceleration to achieve high linearity, a high pull-in voltage, and low damping. The device is made very sensitive by using a torsional beam suspension and a very narrow air gap without limiting the measurement range. Fig. 10 also shows SEM views of this device. The sensor is fabricated using a three-mask dissolved-wafer process [47]. It consists of a 12-μm-thick boron-doped silicon inertial mass suspended 7.5 μm above the glass substrate by two narrow, high-aspect-ratio 12×3 μm torsion beams anchored to the glass substrate. A large number of 300-μm-long capacitive sense fingers are attached to the end of the proof mass, opposite the support beams. These fingers are

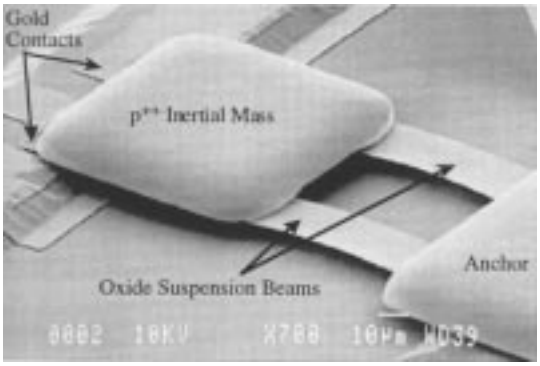
separated by a 2 μm air gap from fixed fingers anchored to the substrate. The accelerometer provides a sensitivity of 20 fF/g over a range of $\pm$4 g, and a bandwidth of 30 Hz when packaged at atmosphere. The sensor is read out using the sensor interface chip discussed above to provide a sensitivity of 300 mV/g. The fabrication process is very simple, while the device structure provides a large bandwidth without special packaging or damping control, does not rely on a large-area parallel-plate gap (which helps circumvent problems with stiction), and has a high sensitivity. In addition, its process is compatible with that of the barometric pressure sensor, allowing its integration with other sensors into a single chip.

Although continuous measurement of acceleration can be performed using the above accelerometer (e.g., to track position changes), it is also desirable to measure single acceleration events (impacts) over a much larger range using an array of acceleration-activated switches. Each of the devices in the threshold accelerometer array functions as a switch that is triggered by an acceleration greater than the threshold level of the sensor [48]. Each sensor consists of an inertial mass, which is suspended by a compliant spring and is separated from the support frame by an air gap, as shown in Fig. 11. A shock or acceleration greater than the threshold of the device causes the mass to deflect, making contact with the frame. A metal strip on the bottom surface of the mass shorts metal contacts on the frame and closes a sense circuit, acting as an event-triggered switch. The threshold level is a function of the mass of the inertial element, the spring constant of the suspension, and the air-gap dimension separating the contacts on the mass from the frame. By altering one or more of these design parameters, different thresholds can be achieved. The accelerometer array is also fabricated using the dissolved wafer process [47] and uses a p^{++} inertial mass suspended by an oxide suspension beam anchored to the glass substrate. The device employs gold contacts on both the p^{++} mass and the glass substrate, as shown in Fig. 11. The array used in the μCluster contains devices with seven different threshold levels, covering a range of 1.5–1000 g [48].

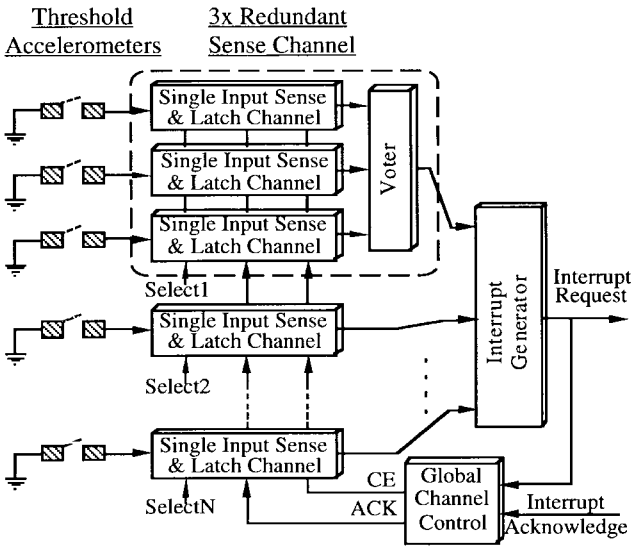
The array is interfaced to a micropower circuit chip capable of sensing switch closure and alerting the microcontroller of the occurrence of an event. Because of extremely low standby power, the array and its interface chip can be used for continuous monitoring of environmental shock or vibration while the remainder of the microsystem is in sleep mode. Fig. 11 also shows a block diagram of the interface circuit [48]. Activation of an element of the sensor array causes an instantaneous high-to-low transition at the input of the corresponding interface channel, which is amplified and latched. Higher reliability and fault tolerance are achieved by using channels with triple redundancy. Each of these redundant channels consists of three individual sense channels and a majority voter. The interface chip can also be programmed to set the overall threshold level of the array by enabling the desired elements. All the control and address signals of the chip are issued by the system microcontroller through the sensor bus and the bus interface



(a)



(b)



(c)

Fig. 11. The threshold accelerometer array and interface chip. Shown are (a) a photograph of the hybrid transducer and its interface circuit, (b) an SEM view of a single threshold accelerometer, and (c) a block diagram of the interface chip.

chip. This chip has been fabricated in a standard 3 μm single-metal double-poly p-well process. It measures 2.2 $\times$ 2.2 mm, dissipates less than 300 μW from a 5 V supply, and can operate up to 4 MHz [48]. Fig. 11 shows the hybrid assembly of the sensor and circuit chips.

V. Software Features

Having a microprocessor embedded within the microsystem provides several advantages, including the ability to alter system operation quickly and easily through changes

in software and the ability to perform digital signal processing on the sensor data. Additionally, sensor-specific software routines and/or measurement parameters can be downloaded to the microsystem, tailoring operation to the set of transducers being utilized. An extension of this concept is to provide a set of test routines that allow each device to be tested within the system to calibrate the entire data path, from the sensor to the digital data stored in the controller. This feature is especially useful for obtaining high accuracy in conjunction with the highly programmable bus interface chip presented above. The ability to test and calibrate the entire microsystem in closed-loop fashion not only improves performance but also simplifies final testing.

In the μCluster, the main control routines are stored in EPROM with several variables allocated to EEPROM so that various operation sequences can be selected for different applications. EEPROM also contains a block of code for each sensor, which determines how it is read out and how often, how it can be self-tested, how its data should be processed (including parameters for digital compensation), and what to do if the sensor does not appear to be functioning correctly. RAM is used for program variables and the temporary storage of sensor data. The basic operating sequence for the μCluster is to scan each front-end device, calibrate (and compensate) the incoming data using one of several algorithms, output the data through the wireless transmitter, program the PMC for the desired sleep interval, shut down most front-end devices and the RF transmitter, and enter sleep mode. During sleep mode, the PMC will monitor any front-end devices that remain active and wake the system when the sleep interval has expired to initiate another scan. To obtain the data needed while minimizing the amount of energy taken from the battery supply, the sensor scan rate (or sleep-mode duration) is adaptively set based on the measurements taken. That is, when the parameters being measured are changing quickly, the scan rate is increased, but when the measured parameters appear to be steady, a very slow scan rate is used. The ability to interrupt sleep mode for an event-triggered response, as in the threshold accelerometer, is an important feature for low-power systems, providing a means for monitoring desired parameters while much of the system is shut down.

Another important software feature that affects overall performance is the digital compensation of the sensor data. For microsystems, the built-in digital signal-processing capabilities can be used to eliminate complex calibration circuitry and achieve as much as two orders of magnitude of error reduction over circuit techniques and laser trims, which have difficulty following nonlinearities [49]–[51]. Look-up tables and polynomial evaluation are the two most common methods [52] for digital compensation, although a combined piece-wise polynomial approach has also been demonstrated [53]. Look-up-table algorithms [54] offer good accuracy but are very demanding on system memory. An attractive alternative is polynomial evaluation [54], [55], which uses significantly less memory than look-up-table methods but is generally slower, with speed set by the operating frequency of the MCU. Polynomial evaluation also makes it possible to perform both calibration (offset/slope) and compensation (e.g., for temperature) simultaneously using bivariate equations, and can produce highly accurate results even for sensors with a very nonlinear response.

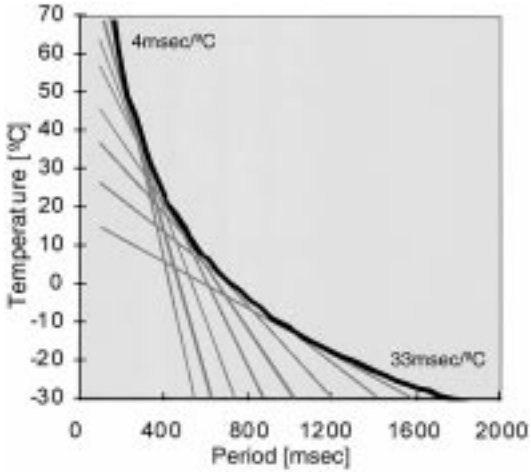


Fig. 12. Response of the integrated temperature sensor along with eight linear approximations used to segment the response range.

On the μCluster, a variety of standard calibration/compensation algorithms were established, including look-up tables, bivariate polynomials, and piece-wise polynomials. The two most interesting examples are calibration of the highly nonlinear temperature sensor and compensation of the pressure sensor. The temperature sensor on the μCluster has a response that is an inverse log function, which is very difficult to fit accurately even with high-order polynomials. However, if the response is broken into segments, the device can be accurately described by four second-order equations, each spanning 20°C. With similar accuracy, the device can be fit by eight first-order equations covering 10°C, as shown in Fig. 12. Here, the response of the sensor is shown along with each of the linear approximations used to calibrate it. For the same error, this piece-wise polynomial approach allows the device to be calibrated about 100 times faster than a single fifth-order polynomial and requires only two-and-one-half times the memory.

The barometric pressure sensor uses a bivariate polynomial for calibration and for compensation of temperature sensitivity. Considering the tradeoffs among accuracy, speed, and required memory, the fourth-order bivariate polynomial

$$\begin{aligned}\text{Pressure} = \big[&\beta_0 \pm \beta_1(V) \pm \beta_2\big(V^2\big) \pm \beta_3\big(V^3\big) \pm \beta_4\big(V^4\big) \\ &\pm \beta_5(T) \pm \beta_6\big(T^2\big) \pm \beta_7\big(T^3\big) \pm \beta_8\big(T^4\big) \\ &\pm \beta_9(TV) \pm \beta_{10}\big(T^2V\big) \pm \beta_{11}\big(TV^2\big) \\ &\pm \beta_{12}\big(TV^3\big) \pm \beta_{13}\big(T^3V\big) \pm \beta_{14}(T_2V_2)\big]\end{aligned}$$

was implemented, where V is the sensor output voltage after analog-to-digital conversion and T is the measured temperature. Using floating-point software routines, this

Table 1 μCluster Characteristics and Eventual Goals

Parameter	Result	Eventual Goal
Microsystem Volume (with PCB)	10 cc	1 cc
Packaged Volume (with batteries)	50 cc	10cc
Power Supply	External (hardwired) (or) 2, 3V/275mA-hr coin cell (or) Standard 9V battery	3V coin cell
Telemetry Range	50m	1 mile
Telemetry Frequency / Modulation	315 MHz / ASK	GHz
Average Power Dissipation*	530µWatt	<1mWatt
Portable Operating Life*	30 days	30 days
Measurement Aperture Time	70µsec	50µsec
Maximum Scan Rate (with output)	10 Hz	1kHz
Sensor Scan Interval	1 minute (typical) Adaptive and Event Triggered	
Temperature	-20 to +60ºC ±0.5ºC	-20 to +60ºC ±0.2ºC
Barometric Pressure	600 to 800 Torr ±100mTorr	500 to 800Torr ±25mTorr
Relative Humidity	25 to 90%RH ±2%RH	15 to 95%RH ±2%RH
Vibration/Acceleration	-4 to +4g ±0.1g	-4 to +4g ±1mg

**Operating Mode Dependent*

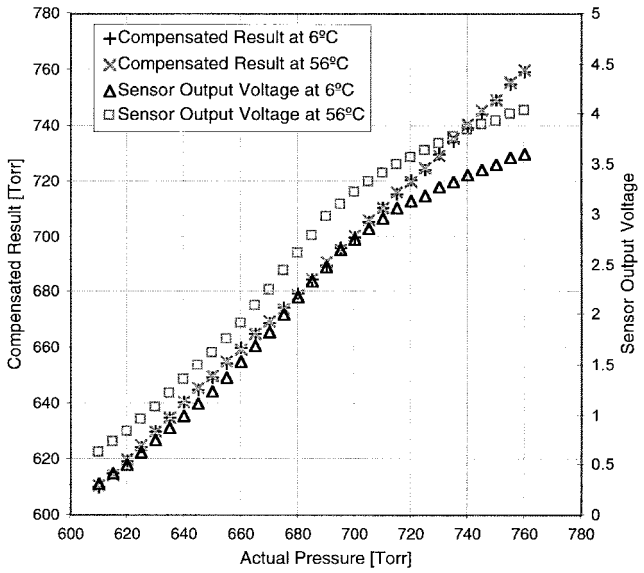


Fig. 13. Response of the barometric pressure sensor at two temperatures. The left vertical axis shows the results after calibration, which match the actual pressure very well. The right axis shows the original sensor output voltage before compensation.

equation can be evaluated in 8.3 ms, and the 15 coefficients β_i can be stored in 45 bytes of MCU memory. For similar accuracy with a look-up table, approximately 9 kB of the 13 kB available on the MCU would be required, although results could be obtained at least ten times faster. Fig. 13 shows the output voltage measured from one of the barometric segment transducers at two temperatures (right axis) and the results after compensation (left axis), which match very closely with the actual pressure. By compensating for nonlinearity and temperature sensitivity, each 50 torr sensing element is accurate to within ±100 mtorr in the range of 600–800 torr. Accuracy approaching the 25 mtorr device resolution can be obtained using higher order polynomials, but at substantial cost in memory and evaluation time.

VI. Results

As an example of a portable wireless microsystem, the μCluster has provided a great deal of information that helps identify the strengths and weaknesses of present technologies and this approach in general. Table 1 lists a number of important parameters along with the results obtained from the μCluster and some longer term goals set at the beginning of this paper. Although some of the longer term goals have not yet been reached, the μCluster was successful in demonstrating the feasibility of a generic microsystem architecture and features such as sensor bus communication, use of a programmable generic interface circuit, and in-module digital calibration and compensation for several sensors. A variety of power-management techniques have also been implemented. While the power dissipation goal of less than 1 mW was achieved, this was only possible at slow sensor scan rates using an RF transmitter with a limited range. Fig. 14 plots the power consumption of the μCluster versus scan interval and shows that the total average power of this configuration is dominated by the MCU. A sharp drop in the overall power consumption can be seen as the scan interval approaches one minute. For longer intervals, a limit of about 400 μW is reached where power consumption is limited by the sleep-mode current of the MCU and, to a lesser degree, the continuous supply to the power-management components. In comparison, the power consumption of the transmitter and the front-end devices is negligible, although at very fast scan rates the transmitter begins to play a larger role. Analysis of the same system with a 3.3 V supply shows an improvement of a factor of two, but any further reduction in power consumption can only be achieved by integrating power-management functions directly on the MCU to minimize all standby currents. As the performance demands for microsystems increase, micropower operation will be an important continuing challenge.

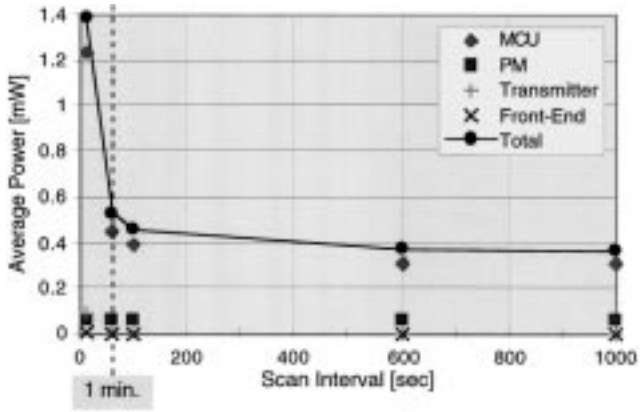


Fig. 14. Average power consumption of the different components in the μCluster as a function of scan interval.

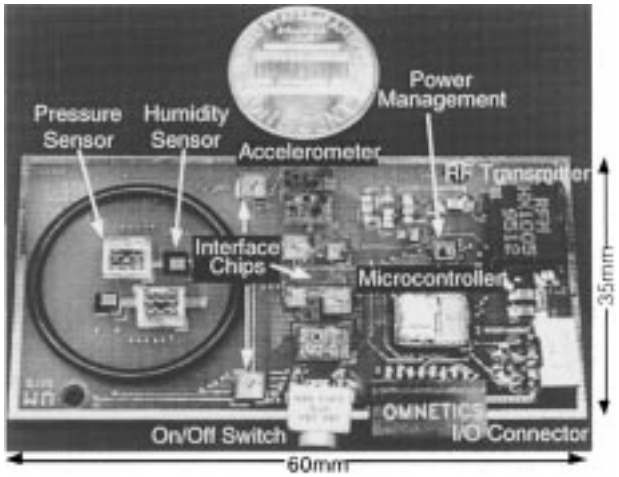


Fig. 15. A populated μCluster, showing the layout of components in the system.

For microsystems to approach the 1 cc wristwatch goal listed in Table 1, packaging technology appears to be the greatest issue. Unpackaged, the μCluster components and mounting substrate occupy 10 cc, while a packaged module with batteries fills 50 cc. Considering the components themselves, only 50% combined volume is occupied by control, communication, and front-end devices, while the remaining 50% is used for components such as the hard-wired I/O connector and the on/off switch, which could be made smaller with improvements in packaging. Battery size is also a significant concern, since a single 3 V lithium coin cell with sufficient energy for microsystem applications occupies about 1.6 cc by itself.

In constructing the present μCluster, the components were mounted on a custom three-layer printed circuit board, as shown in Fig. 15. One version of the μCluster has been used in a number of artillery exercises with the U.S. Marine Corps, where it has been benchmarked against existing meteorological equipment. The microsystem has flown on an unmanned aerial vehicle (UAV) at the Naval Research Laboratory, where μCluster data were transmitted through a satellite link while the UAV was in flight. An ocean buoy has also been modified for μCluster measurements at the Naval Command, Control, and Ocean Surveillance Center. These experiments have demonstrated the effectiveness of the μCluster and have shown the importance of similar microsystems to a variety of emerging applications.

VII. Conclusion

This paper has reported a microinstrumentation system for the measurement of environmental temperature, barometric pressure, humidity, and acceleration. It uses a generic open architecture that permits it to be customized for a given application through the choice of front-end sensors and through the control software resident in the embedded microcontroller. The MCU periodically scans the sensors, calibrates and compensates their data, and communicates the resulting information to the outside world using either a hard-wired system bus or a wireless link. The scan rate is programmable and adaptive. An associated power-management chip switches the peripheral sensors off when not in use and allows the MCU to operate in a low-power sleep mode between scans. The MCU communicates with the front-end sensors and actuators using a nine-line sensor bus, which permits up to 16 different front-end sensing nodes, each with 64 directly addressable elements, to be used in the microsystem. The system thus supports electronic trims, variable readout gain, subranging, and sensor self-test. A generic CMOS readout chip interfaces between the capacitive sensors and the sensor bus. Sensor outputs in the form of frequency-encoded digital waveforms and analog-amplitude formats are accommodated.

The microsystem module operates over an ambient temperature range from -20 to $+60$°C with a power dissipation of 530 μW at a typical sensor scan interval of 1 min. Digital compensation within the μCluster improves sensor performance significantly compared to analog trims. Temperature is measured with an accuracy of 1°C, pressure to $\pm$100 mtorr over the range from 600 to 800 torr, and humidity to $\pm$2%RH over the 25–90%RH range. Acceleration is measured $\pm$0.1 g over the 0 to $\pm$4 g range, while an array of threshold accelerometers continuously monitor for impact events with threshold levels from 1.5 to 1000 g. The microsystem occupies an unpackaged volume of only 10 cc. It operates from two 3 V lithium coin cells (275 mA-h) with a life of up to one month and with a communication range of 50 m. Such microsystems should continue to improve significantly in size and performance during the coming decade and promise to see wide application in a broad array of important products.

Acknowledgment

The authors wish to thank W. G. Baer, A. Selvakumar, U. Kang, R. De Souza, C. Yeh, and F. Ayazi for their assistance in various aspects of this project. The efforts of C. M. Kiers (Navy Science Assistance Program) and R. R. Whitlock (Naval Research Laboratory) in testing and demonstrating the μCluster in the field are also gratefully acknowledged.

REFERENCES

[1] J. H. Huijsing, "Integrated smart sensors," *Sensors Actuators A*, vol. 30, nos. 1/2, pp. 167–174, 1992.

[2] K. D. Wise, "Microelectromechanical systems: Interfacing electronics to a nonelectronic world," in *Dig. IEEE Int. Electron Device Meeting*, Dec. 1996, pp. 11–18.

[3] J. Bryzek, "MEMS: A closer look," *Sensors Mag.*, pp. 4–9, July 1996.

[4] L. Spangler and C. J. Kemp, "A smart automotive accelerometer with on-chip airbag deployment circuits," in *Dig. Solid-State Sensor and Actuator Workshop*, Hilton Head Island, SC, June 1996, pp. 211–214.

[5] H. Baltes, H. Haberli, P. Malcovati, and F. Maloberti, "Smart sensor interfaces," in *Dig. IEEE Int. Symp. Circuits and Systems*, Atlanta, GA, May 1996, vol. 4, pp. 380–383.

[6] E. Yoon and K. D. Wise, "An integrated mass flow sensor with on-chip CMOS interface circuitry," *IEEE Trans. Electron Devices*, vol. 39, pp. 1376–1386, June 1992.

[7] S. Middlehoek and S. A. Audet, *Silicon Sensors*. London, UK: Academic, 1989.

[8] K. D. Wise, "Integrated microsystems: Device and technology challenges," in *Proc. Eur. Solid-State Device Res. Conf. (ESSDERC)*, The Hague, The Netherlands, Sept. 1995, pp. 15–24.

[9] A. Mason, N. Yazdi, K. Najafi, and K. Wise, "A low-power wireless microinstrumentation system for environmental monitoring," in *Dig. Int. Conf. Sensors and Actuators (Transducers'95)*, Stockholm, Sweden, June 1995, pp. 107–110.

[10] B. J. Hosticka, "CMOS sensor systems," in *Dig. Int. Conf. Solid-State Sensors and Actuators (Transducers'97)*, Chicago, IL, June 1997, pp. 991–993.

[11] K. D. Wise, "Integrated microinstrumentation systems: Smart peripherals for distributed sensing and control," in *Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1993, pp. 126–127.

[12] C. J. Koomen, "Technologies for the multimedia city," in *Proc. European Solid-State Device Research Conf. (ESSDERC)*, The Hague, The Netherlands, Sept. 1995, pp. 25–36.

[13] J. E. Brignell, "Sensors in distributed instrumentation systems," *Sensors Actuators*, vol. 10, nos. 3/4, pp. 249–261, 1986.

[14] K. D. Wise, "Device and technology challenges for integrated sensors," in *IEEE Device Research Conf.*, Charlottesville, VA, June 1995.

[15] K. D. Wise and N. Najafi, "The coming opportunities in microsensor systems," in *Digest, IEEE Int. Conf. Solid-State Sensors and Actuators (Transducers'91)*, San Francisco, CA, June 1991, pp. 2–7, invited.

[16] K. M. Mahmoud, R. P. van Kampen, M. J. Rutka, and R. F. Wolffenbuttel, "A silicon integrated smart pressure sensor," in *Dig. Int. Conf. Solid-State Sensors and Actuators (Transducers'93)*, Yokohama, Japan, June 1993, pp. 217–220.

[17] E. J. Hogenbirk, H. J. Verhoeven, and J. H. Huijsing, "An integrated smart sensor for flow and temperature with I^2C bus interface based on thermal sigma-delta modulation," in *Dig. Int. Conf. Solid-State Sensors and Actuators (Transducers'93)*, Yokohama, Japan, June 1993, pp. 792–795.

[18] "MC68HC11E9 (HCMOS microcontroller unit) technical data manual," Motorola, Inc., Phoenix, AZ, 1991.

[19] A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[20] J. D. Meindl, "Low power microelectronics: Retrospect and prospect," *Proc. IEEE*, vol. 83, pp. 619–635, Apr. 1995.

[21] A. A. Abidi, "Low-power radio-frequency IC's for portable communications," *Proc. IEEE*, vol. 83, pp. 544–569, Apr. 1995.

[22] B. Davari, R. H. Dennard, and G. G. Shahidi, "CMOS scaling for high performance and low power—The next ten years," *Proc. IEEE*, vol. 83, pp. 595–606, Apr. 1995.

[23] R. A. Powers, "Batteries for low power electronics," *Proc. IEEE*, vol. 83, pp. 687–693, Apr. 1995.

[24] P. B. Koeneman, I. J. Busch-Vishniac, and K. L. Wood, "Feasibility of micro power supplies for MEMS," *J. Microelectromech. Syst.*, vol. 6, no. 4, pp. 355–362, Dec. 1997.

[25] L. E. Larson, *RF and Microwave Circuit Design for Wireless Communications*. Norwood, MA: Artech House, 1996.

[26] C. T.-C. Nguyen, L. P. B. Katehi, and G. M. Rebeiz, "Micromachined devices for wireless communications," *Proc. IEEE*, this issue, pp. 1756–1768.

[27] K. Bult, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T. H. Lin, W. J. Kaiser, R. Murkai, P. Nelson, F. L. Newburg, K. S. J. Pister, G. Pottie, H. Sanchez, O. M. Stafsudd, K. B. Tan, C. M. Ward, G. Yung, S. Xue, H. Marcy, and J. Yao, "Wireless integrated microsensors," in *Dig. Solid-State Sensor and Actuator Workshop*, Hilton Head, SC, June 1996, pp. 205–210.

[28] A. Kompolt and J. Brewer, "Designing a wristwatch pager creates its own special challenges," *Wireless Syst. Design*, pp. 37–41, Mar. 1997.

[29] J. R. Moyne, N. Najafi, D. Judd, and A. Stock, "Analysis of sensor/actuator bus interoperability standard alternatives for semiconductor manufacturing," in *Proc. Sensors Expo*, Cleveland, OH, Sept. 1994, pp. 375–386.

[30] N. Najafi and K. D. Wise, "An organization and interface for sensor-driven semiconductor process control systems," *IEEE Trans. Semiconduct. Manufact.*, vol. 3, pp. 230–238, Nov. 1990.

[31] S. P. Woods, "The IEEE-P1451 transducer to microprocessor interface," *Sensors Mag.*, pp. 43–48, June 1996.

[32] L. H. Eccles, "A brief description of IEEE P1451.2," in *Proc. Sensors Expo*, Detroit, MI, Oct. 1997, pp. 81–90.

[33] K. Watanabe and W. S. Chung, "A switched capacitor interface for intelligent capacitive transducers," *IEEE Trans. Instrum. Meas.*, vol. IM-35, pp. 472–476, Dec. 1986.

[34] Y. E. Park and K. D. Wise, "An MOS switched-capacitor readout amplifier for capacitive pressure sensors," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1983, pp. 380–384.

[35] Y. Cao and G. C. Temes, "High-accuracy circuits for on-chip capacitance ratio testing or sensor readout," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 637–639, Sept. 1994.

[36] S. L. Garverick, M. L. Nagy, N. K. Rao, D. K. Hartsfield, and A. Purushotham, "A capacitive sensing integrated circuit for detection of micromotor critical angles," *IEEE J. Solid-State Circuits*, vol. 32, pp. 23–30, Jan. 1997.

[37] CSEM2003 (capacitive sensor interface with gain and offset adjustment) data sheet, Centre Suisse d'Electronique et de Microtechnique SA, Neuchatel, Switzerland, 1995.

[38] ISC721 (capacitive sensor interface circuit) preliminary data sheet, Irvine Sensors Corporation, Irvine, CA, 1997.

[39] N. Yazdi, A. Mason, K. Najafi, and K. Wise, "A low-power generic interface circuit for capacitive sensors," in *Dig. Solid-State Sensor and Actuator Workshop*, Hilton Head Island, SC, June 1996, pp. 215–218.

[40] E. Vittoz and O. Neyroud, "A low-voltage CMOS bandgap reference," *IEEE J. Solid-State Circuits*, vol. SC-14, pp. 573–577, June 1979.

[41] G. C. M. Meijer, "Thermal sensors based on transistors," *Sensors Actuators*, vol. 10, pp. 103–125, 1986.

[42] Y. Zhang and K. D. Wise, "A high-accuracy multi-element silicon barometric pressure sensor," in *Dig. Int. Conf. Sensors and Actuators (Transducer'95)*, Stockholm, Sweden, June 1995, pp. 608–611.

[43] A. V. Chavan, and K. D. Wise, "A batch-processed vacuum-sealed capacitive pressure sensor," in *Dig. Int. Conf. Solid-State Sensors and Actuators (Transducers'97)*, Chicago, IL, June 1997, pp. 1449–1452.

[44] A. B. Frazier and M. G. Allen, "Metallic microstructures fabricated using photosensitive polyimide electroplating molds," *IEEE J. Microelectromech. Syst.*, vol. 2, pp. 87–94, 1993.

[45] N. Yamazoe, "Humidity sensors: Principles and applications," *Sensors Actuators*, vol. 10, pp. 379–398, 1986.

[46] A. Selvakumar, F. Ayazi, and K. Najafi, "A high sensitivity Z-axis torsional silicon accelerometer," in *Dig. IEEE Int. Electron Device Meeting*, San Francisco, CA, Dec. 1996, pp. 765–768.

[47] Y. Gianchandani and K. Najafi, "A bulk silicon dissolved wafer process for microelectromechanical devices," *J. Microelectromech. Syst.*, vol. 1, no. 3, pp. 77–85, June 1992.

[48] A. Selvakumar, N. Yazdi, and K. Najafi, "A low power, wide range threshold acceleration sensing system," in *Proc. IEEE Microelectromech. Syst. Workshop (MEMS 96)*, San Diego, CA, Feb. 1996, pp. 186–191.

[49] O. Machul, D. Hammerschmidt, W. Brockherde, B. J. Hosticka, E. Obermeier, and P. Krause, "A smart pressure transducer with on-chip readout, calibration and nonlinear temperature compensation based on spline-functions," in *Dig. IEEE Int. Solid-State Circ. Conf.*, Feb. 1997, pp. 198–199.

[50] J. A. Lloyd, H.-S. Lee, L. Parameswaran, M. A. Schmidt, and C. G. Sodini, "An adaptive calibration technique for micromachined pressure sensors," in *Dig. Int. Conf. Solid-State Sensors and Actuators (Transducers'97)*, Chicago, IL, June 1997, pp. 295–298.
[51] J. Bryzek, "Modeling sensor performance for smart transducers," in *Proc. Sensors Expo*, Cleveland, OH, Sept. 1994, pp. 501–507.
[52] X. Li, G. C. M. Meijer, and G. W. de Jong, "A self-calibration technique for a smart capacitive angular-position sensor," in *IEEE Instrum. Meas. Tech. Conf.*, Brussels, Belgium, June 1996, pp. 774–777.
[53] P. T. P. Tang, "Table-lookup algorithms for elementary functions and their error analysis," in *Proc. IEEE Symp. Computer Arithmetic*, June 1991, pp. 232–236.
[54] S. B. Crary, W. G. Baer, J. C. Cowles, and K. D. Wise, "Digital compensation of high-performance silicon-pressure transducers," *Sensors Actuators*, vol. A21–A23, pp. 70–72, 1990.
[55] Y. Yoshii, A. Nakajo, H. Abe, K. Ninomiya, H. Miyashita, N. Sakuri, M. Kosuge, and S. Hao, "1 chip integrated software calibrated CMOS pressure sensor with MCU, A/D convertor, D/A convertor, digital communication port, signal conditioning circuit and temperature sensor," in *Dig. Int. Conf. on Solid-State Sensors and Actuators (Transducers'97)*, Chicago, IL, June 1997, pp. 1485–1488.

Andrew Mason received the B.S. degree in physics (with highest distinction) from Western Kentucky University, Bowling Green, in 1991, the B.S.E.E. degree from the Georgia Institute of Technology, Atlanta, in 1992, and the M.S. degree in electrical engineering from The University of Michigan, Ann Arbor, in 1994, where he currently is pursuing the Ph.D. degree.

He currently is a Senior Graduate Student Research Assistant with the Center for Integrated Sensors and Circuits, The University of Michigan. For the past five years, he has been involved in a project sponsored by the Defense Advanced Research Project Agency to develop a microinstrumentation cluster for environmental monitoring. On this project, he has led the efforts on system design, evaluation, and assembly, developed communication protocols and signal-processing software, and designed sensor interface and power management circuitry. His current research focuses on advanced microsystem features such as in-module digital sensor compensation, low-power system operation, and highly programmable sensor interface circuitry.

Navid Yazdi, for a photograph and biography, see this issue, p. 1659.

Abhijeet V. Chavan received the B.S. degree in electrical engineering (with highest honors) from M.S. University of Baroda, India, in 1983 and the M.S. degree in computer engineering from Syracuse University, Syracuse, NY, in 1988. He currently is pursuing the Ph.D. degree in electrical engineering at The University of Michigan, Ann Arbor.

From 1983 to 1987, he was a Design Engineer with Siemens A.G., Bombay, India, and later in Amberg, Germany, involved in design of industrial automation systems. From 1987 to 1990, he was with Coherent Research, New York, where he was involved in development of associative memory processors used for hardware acceleration. He is a Senior Project Engineer with the Delco Delphi Electronics division of General Motors (currently on study leave), where he has been involved with many automotive electronic integrated circuit designs. He has received three patents. His research interests include micromechanical system design, mixed signal circuit design, and real-time systems integration.

Khalil Najafi (Senior Member, IEEE), for a photograph and biography, see this issue, p. 1659.

Kensall D. Wise (Fellow, IEEE), for a photograph and biography, see this issue, p. 1533.