Office of Technology Development
Robotics Technology Development Program

WHOLE-ARM OBSTACLE AVOIDANCE SYSTEM CONCEPTUAL DESIGN

A. L. Wintenberg, P. L. Butler*, S. M. Babcock*,
M. N. Ericson, and C. L. Britton, Jr.

Instrumentation and Controls Division
*Robotics & Process Systems Division
Oak Ridge National Laboratory

Date Published–April 1993

MASTER

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

Whole-Arm Obstacle Avoidance System
Conceptual Design

1. SYSTEM OVERVIEW

Whole-arm obstacle avoidance is needed for a variety of robotic applications in the Environmental Restoration and Waste Management (ER&WM) Program. Typical industrial applications of robotics involve well-defined workspaces, allowing a predetermined knowledge of collision-free paths for manipulator motion. In the unstructured or poorly defined hazardous environments of the ER&WM Program, the potential for significant problems resulting from collisions between manipulators and the environment in which they are utilized is great. The conceptual design for a sensing system that will provide protection against such collisions is described herein.

The whole-arm obstacle avoidance system consists of a set of sensor "bracelets," which cover the surface area of the manipulator links to the maximum extent practical, and a host processor. The host processor accepts commands from the robot control system, controls the operation of the sensors, manipulates data received from the bracelets, and makes the data available to the manipulator control system. The bracelets consist of a subset of the sensors, associated sensor interface electronics, and a bracelet interface. Redundant communications links between the host processor and the bracelets are provided, allowing single-point failure protection. The system allows reporting of 8-bit data from up to 1000 sensors at a minimum of 50 Hz. Faster update rates are possible with fewer sensors and/or lower resolution data.

While the initial prototype implementation of the system utilizes Sandia National Laboratories (SNL) capacitance proximity sensor [Novak and Feddema, 1992], the system concept allows multiple types of sensors. These sensors are uniquely addressable, allowing remote calibration, thresholding at the bracelet, and correlation of a sensor measurement with the associated sensor and its location on the manipulator. Variable resolution allows high-speed, single-bit sensing (object present or not) as well as lower-speed higher-resolution sensing, which is necessary for sensor calibration and potentially useful in control.

1

2. SYSTEM FUNCTIONS

Functions to be performed by the whole-arm obstacle avoidance system are described in the following paragraphs that cover the sensor interface, the robot control system interface, and the modes of operation.

2.1. SENSOR INTERFACE

The sensor interface electronics provide the necessary input and output interface to the sensor, which, in the case of the SNL capacitance proximity sensor, is an oscillator signal and a modulated sensor output signal. In order to accommodate alternate sensors, a digital input/output port is provided as well as direct access to an analog-to-digital converter.

2.2. ROBOT CONTROL SYSTEM INTERFACE

The robot control system interface provided by the whole-arm obstacle avoidance host computer allows command inputs including:

1) operating mode selection, including single or double thresholding, and
2) sensor identification and location (initial setup).

The interface allows the future addition of input of the robot kinematic configuration (including basic kinematics and current joint angles, which may be needed for calibration and compensation).

The interface provides the following output data:

1) scaled, variable resolution sensor data and associated sensor identification for sensors with readings within one and two thresholds (two thresholds are required to provide a caution range and a halt range) specified on an individual sensor basis or specified for all sensors (for object present indication, minimum resolution, two bit data suitable for both single and double thresholding),
2) two flags indicating one or more sensor readings within a single threshold and between two thresholds, and
3) a flag indicating a fault condition and associated information.

Three primary interface types are provided. One of the robot control system interfaces is compatible with communications standards established for the generic intelligent system controller (GISC). In addition, a function library interface (less restrictive than the GISC protocol) and a common memory interface are provided. A terminal interface is also provided for setup and testing, independent of a robot control system.

2.3. MODES OF OPERATION

The system is capable of operating in the following modes:

1) setup of sensors
2) calibration of sensors
3) variable resolution threshold reporting of obstacle detection in a scanned mode

2.3.1. Setup

In this mode, the sensors are scanned for sensor type and data. Sensor setup parameters (gains, etc.) are determined for each of the sensors based upon the sensor type and the sensor data received and sent individually to the sensors. On the basis of the data, the type of sensor, and inputs from the robot control system, two thresholds are sent to the appropriate bracelet for each of the sensors. In addition, a request for data of interest (based on the thresholds specified) may be specified, which causes the sensor data to be transmitted for all sensors with readings that satisfy either of the threshold criteria.

2.3.2. Calibration

Two-point calibration of the system is possible by placing a standard object at a known distance from the sensor in addition to checking the sensor output with no object present. Multipoint (more than two point) calibration using curve fitting or table look-up is a possible future addition. In this mode, the sensors are scanned and calibration constants are determined. The data from a single sensor or subset of sensors are available upon request from the host processor. Note that it is possible to excite a single sensor and read data from several in order to check for interferences between sensors. For the capacitance proximity sensor, calibration is partially verifiable without physical access to the sensor, i.e., without having to place an object at a known distance in front of a sensor. As a possible addition in the remote calibration mode, the system will automatically determine compensation for the effects of interferences caused by nearby linkages as the manipulator kinematic configuration changes.

2.3.3. Threshold Reporting

In the threshold reporting mode, two threshold levels (caution and halt) specified during setup are checked at the bracelet. At a minimum, two bits of data associated with the presence or absence of an obstacle within the two thresholds are reported. In addition, if either threshold indicates the presence of an object, and a request for data is specified in the setup mode, sensor data of interest is reported.

3. SYSTEM ARCHITECTURE

3.1. OVERVIEW

The system architecture is hierarchical in nature, as shown in Fig. 1. The host processor communicates with a set of bracelet processors or equivalent electronic components (if required for high radiation environments). The bracelet processor communicates with a subset of the sensor node electronics interfaces, which provide the direct interfaces to the sensors. The host processor provides a variety of interfaces to the robot control system as described below.
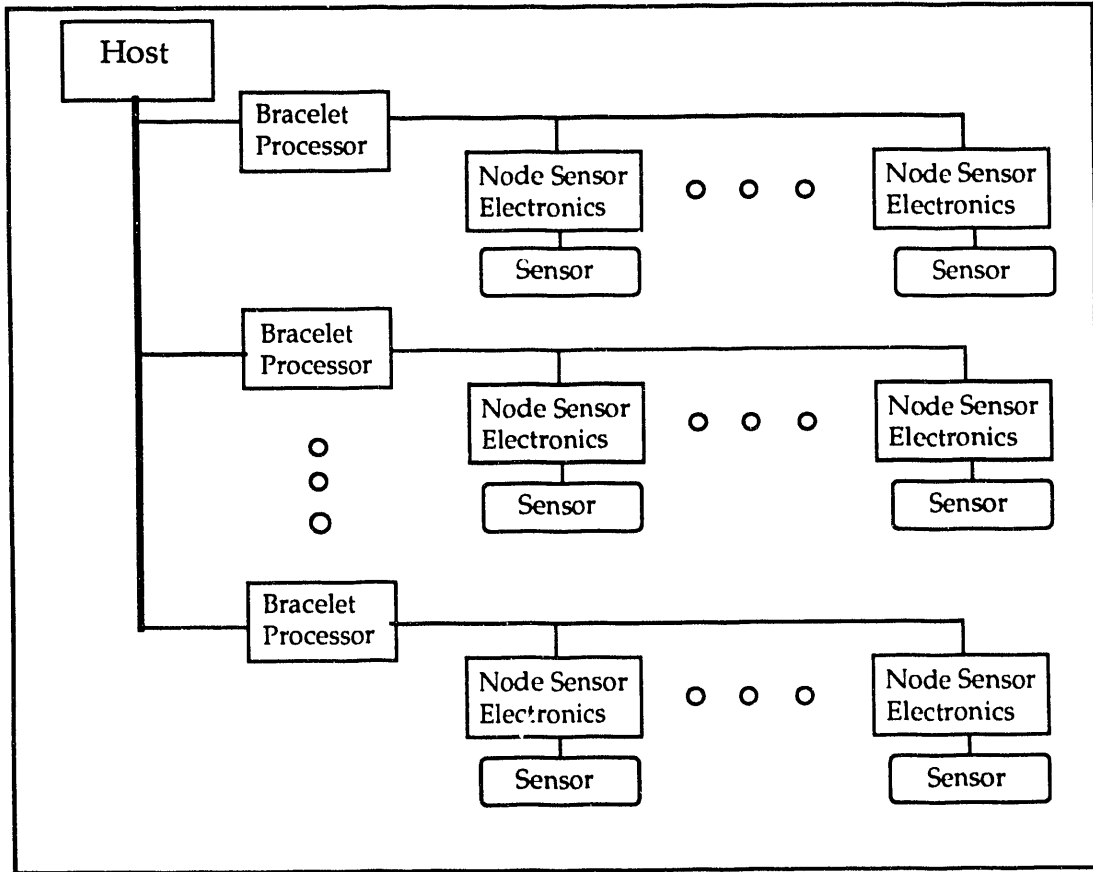
Fig. 1. Hierarchical system architecture.

The bracelet processor handles the high-speed multi-drop communications link from the host and transforms control information into a simpler control packet to the sensor nodes. Return data are combined by the bracelet processor into a single packet for high-speed transmission to the host. The bracelet processor also performs the multi-drop address decode and communications protocol handling. This protocol may include a checksum or CRC code for transmission error detection. The bracelet processor can also handle caution and halt thresholding for all sensors on the bracelet.

The node accepts low-level commands from the bracelet processor and executes them. During operation, the command is normally to acquire data. However, setup commands can be executed to set parameters such as offsets and gains.

3.2. HOST

The host system is a commercially available 680x0 VME computer that can be interfaced easily to new or existing robotic systems. The obstacle avoidance system operates independently from other robotic system modules on the host processor. Depending on the computational requirements, more than one processor may be required. This will be needed when kinematic constraints are included in the obstacle avoidance system. Because the host to bracelet communications link is based on industry standards, commercially available hardware should be available to handle the host to bracelet communications link. In fact, the communications port may be a part of the host processor board. For high-speed interfaces to the robot controller or other systems, the VME bus interface will be adequate. For lower-speed requirements, Ethernet can be used with GISC function calls.

3.3. BRACELET PROCESSOR

The bracelet processor serves as the interface between the sensor nodes and the host processor. It is responsible for communications to the host and to the sensor nodes, for data thresholding and formatting and for distribution of power and clock signals to the sensor nodes. The connections between the bracelet processor, the sensor nodes, and the host processor are shown in Fig. 2.

Fig. 2. Bracelet processor block diagram.

3.3.1. Bracelet Processor Implementation

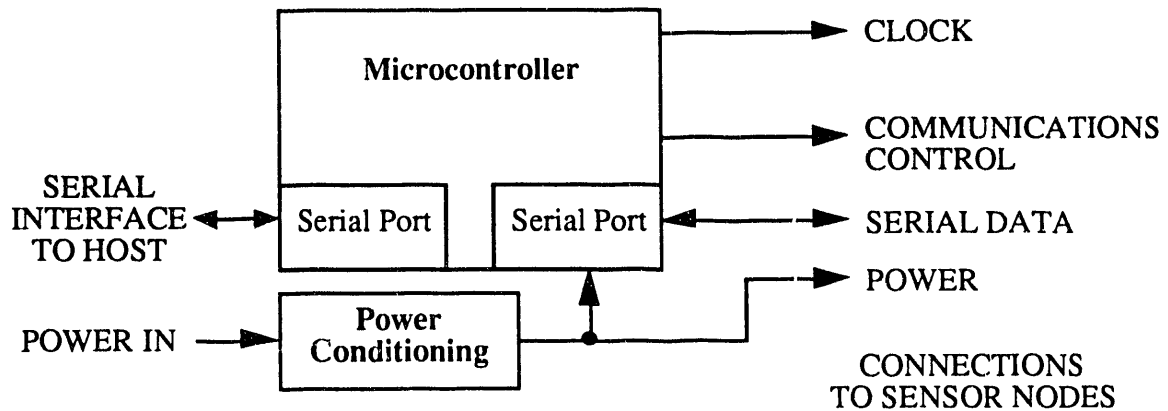The bracelet processor is based on the 8051 microcontroller family. Available variations on the basic 8051 incorporate dual serial communications ports, I/O control ports, on-chip memory, and other features that help in miniaturizing the bracelet processor module by minimizing the number of auxiliary ICs that are needed.

5

No commercially available radiation-hard microcontrollers have been located. Radiation-hard microprocessors and support chips are available, though with limited variety. The use of a rad-hard microprocessor in the bracelet processor presents two primary problems. First, several support chips must be used to build a control module that is functionally equivalent to a single microcontroller. This would significantly increase the size of the bracelet processor module. Second, rad-hard processors are generally military or space qualified and, as a result, very expensive.

To implement a rad-hard version of the obstacle avoidance system, a rad-hard application specific integrated circuit (ASIC) could be developed to replace the microcontroller-based bracelet processor. This ASIC should be only moderately complex because of the limited functional requirements of the bracelet processor (basic communications and thresholding).

3.3.2. Power Conditioning

Power supplied by the host will be conditioned at the bracelet processor level and distributed to the sensor nodes on the bracelet. To allow for voltage drops caused by potentially long cabling and to provide noise immunity, voltage regulators will be used to reduce the ±10V supplied by the host to 5V for the microcontroller and ±5V for the sensor node electronics. Current limiting will be included so that a shorted sensor node will at most disable its own bracelet.

3.4. NODE ELECTRONICS

A sensor node contains the sensor, sensor interface electronics and communications electronics. The sensor interface electronics depend upon the type of sensor; however, they would typically include control functions and some type of signal conditioning. Some of these functions would be common to several types of sensors while others would be specific to one type of sensor, and a logical division of the sensor interface electronics would be between sensor-specific functions and general functions. The communications electronics should be the same for all sensor types to allow an avoidance system to include multiple sensor types.

3.4.1. Sensor Node Electronics Concept

The sensor node is implemented using one or two ASICs. Use of ASICs allows the most compact and power-efficient implementation of the sensor node electronics. For future upgrades, a radiation-hard IC process may be used to provide radiation-hard electronics for the sensor node.

It is useful to partition the sensor node electronics into two sections: one containing all the common functions and one with functions specific to a single sensor type. The section common to all sensor types will be referred to as the sensor node core and the section specific to a single sensor type will be referred to as sensor-specific electronics. This partitioning is illustrated in Fig. 3.

3.4.1.1. Sensor Node Core

The sensor node core will contain the communications block, control logic, an analog-to-digital converter and a port for accepting digital data from the sensor. The communications block will receive commands and setup parameters from the bracelet controller, and transmit data

LINK TO
BRACELET

```
           COMMUNICATIONS

                    CONTROL
                    LOGIC                SENSOR-
                                         SPECIFIC        SENSOR
           ANALOG TO    analog           ELECTRONICS
           DIGITAL
           CONVERTER

                              digital
```
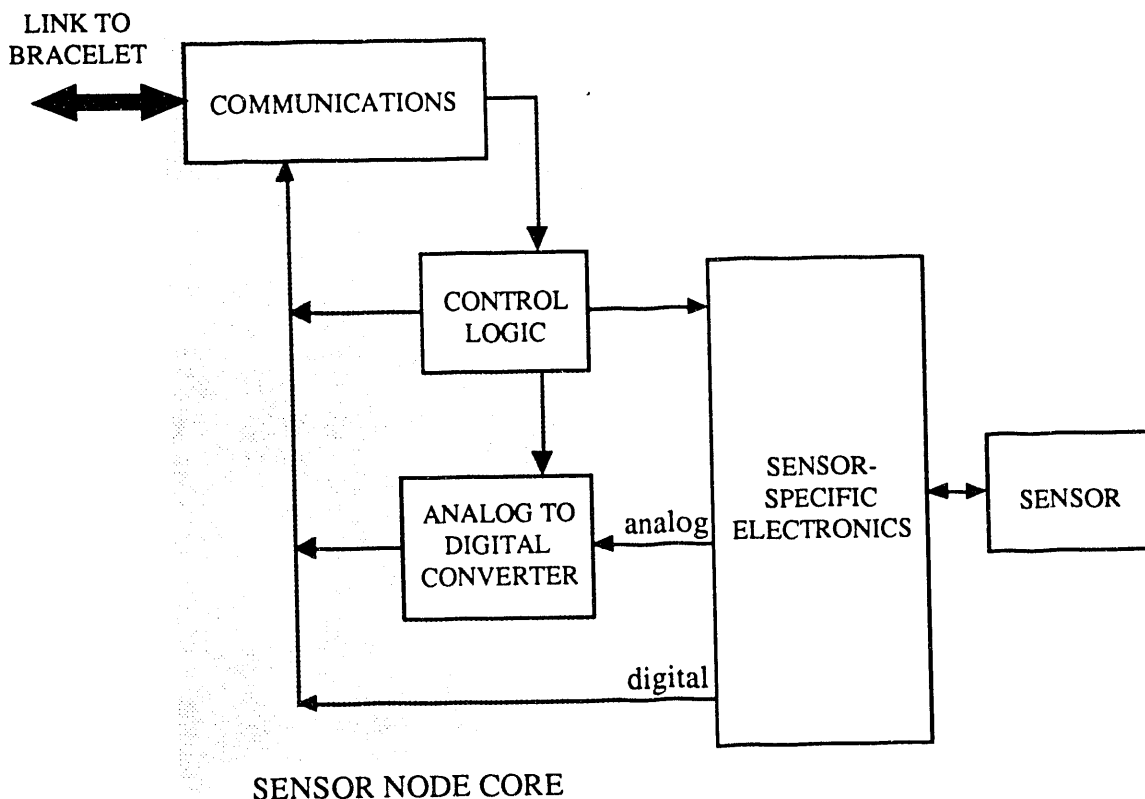
SENSOR NODE CORE

Fig. 3. Sensor node electronics.

from the node back to the controller. The control logic will control the ADC, control the routing of data to the communications block, and pass commands to any sensor-specific control logic. The ADC will take analog information from the sensor and convert it to digital format. A digital port is provided for compatibility with digital output sensors.

The sensor node core will be designed for implementation as a single ASIC. It is anticipated that the sensor-specific electronics could be included on the same chip with minimal additional production cost. Thus each type of sensor would require a unique chip. (All would have the same core and would therefore be compatible with the bracelet system.) This one-chip solution would be the most compact and cost-effective solution. However, for some sensors it may be desirable to have the sensor specific electronics separate from the sensor node core. For example, some types of sensor units may include all the needed sensor-specific electronics, or it may be desirable to prototype or make a small quantity of sensor nodes without having to develop an ASIC.

A reasonable approach now is to develop the sensor node core and the sensor-specific electronics for the capacitive proximity sensor as a single ASIC. This gives the lowest prototyping cost and the best illustration of the advantages of the single chip solution. Additional inputs and outputs from the sensor node core can be provided so that other types of sensors could be interfaced.

3.4.1.2. Analog-To-Digital Converter

Several possibilities for the analog-to-digital converter were considered. The capacitive proximity sensor requires a converter with nearly 12 bits of dynamic range. (A more detailed description of the requirements of the ADC for the capacitive sensor is given in section 3.4.2.) Other sensor types may require more or less resolution, but development of a converter with more than 12 bits of dynamic range would require a level of effort well beyond the scope of this project. Thus it was decided that an ADC with 12 bits of dynamic range would be developed.

One approach would involve the development of a 12-bit integrating converter. This development would be relatively straightforward, but would require considerable effort. The approach chosen will make use of an existing 8-bit ADC. By using multiple gains along with an autoranging feature, it is possible to use the 8-bit converter to provide the desired accuracy. The 8-bit converter design has a proven track record. It is designed for a shorter conversion time (5-10 μs) than is needed for this application, but that presents no problem. A block diagram of the ADC along with the amplifiers and autoranging circuitry is shown in Fig. 4. This autoranging ADC will provide 8 bits of resolution, but with the same dynamic range as a 12-bit ADC.

3.4.1.3. Sensor-Specific Electronics

Sensor-specific electronics may include control, excitation and signal processing functions. Control functions may include setup functions such as programming the gain of an amplifier or operation functions such as time measurement. A sensor may require excitation in the form of a constant current or a signal from an oscillator. Signal processing functions could be analog or digital and might include filtering or averaging.

3.4.1.4. ASIC Technology

The sensor node ASICs will be designed using a 2-micron p-well CMOS process. This process was chosen for a number of reasons. First, this process is readily available from the MOSIS service on a monthly basis and from ORBIT Semiconductor on a weekly basis. Second, a large number of useful designs (opamps, ADC, etc.) are available in that process. Third, the p-well process allows the use of +5V logic and ±5V analog circuitry on the same IC.

3.4.2. Capacitive Proximity Sensor Application

The prototype of the whole-arm obstacle avoidance system uses the SNL developed capacitive proximity sensor. The following paragraphs describe the sensor, the analog signal processing and digitization, and the sensor-specific electronics.

3.4.2.1. Capacitive Proximity Sensor

The capacitive proximity sensor is an active sensor. It is driven by an oscillator, and requires active electronics to detect the current passing through the sensor capacitance. An object nearing the sensor reduces the sensor capacitance and therefore the current passing through. Thus the detected current is a function of the proximity of an object. The sensor response is proportional to the distance to the object, but is nonlinear and varies approximately as $S = C_1(1+k\text{-exp}(-d/d_0))$ where C_1 is a constant of proportionality, k is a positive constant on the order of unity, d is the

Start   Clock

Auto-ranging logic

Control logic → Done

Reference voltage

control

control

Input

Mux

8-bit ADC → Output

Comparators

Amplifiers

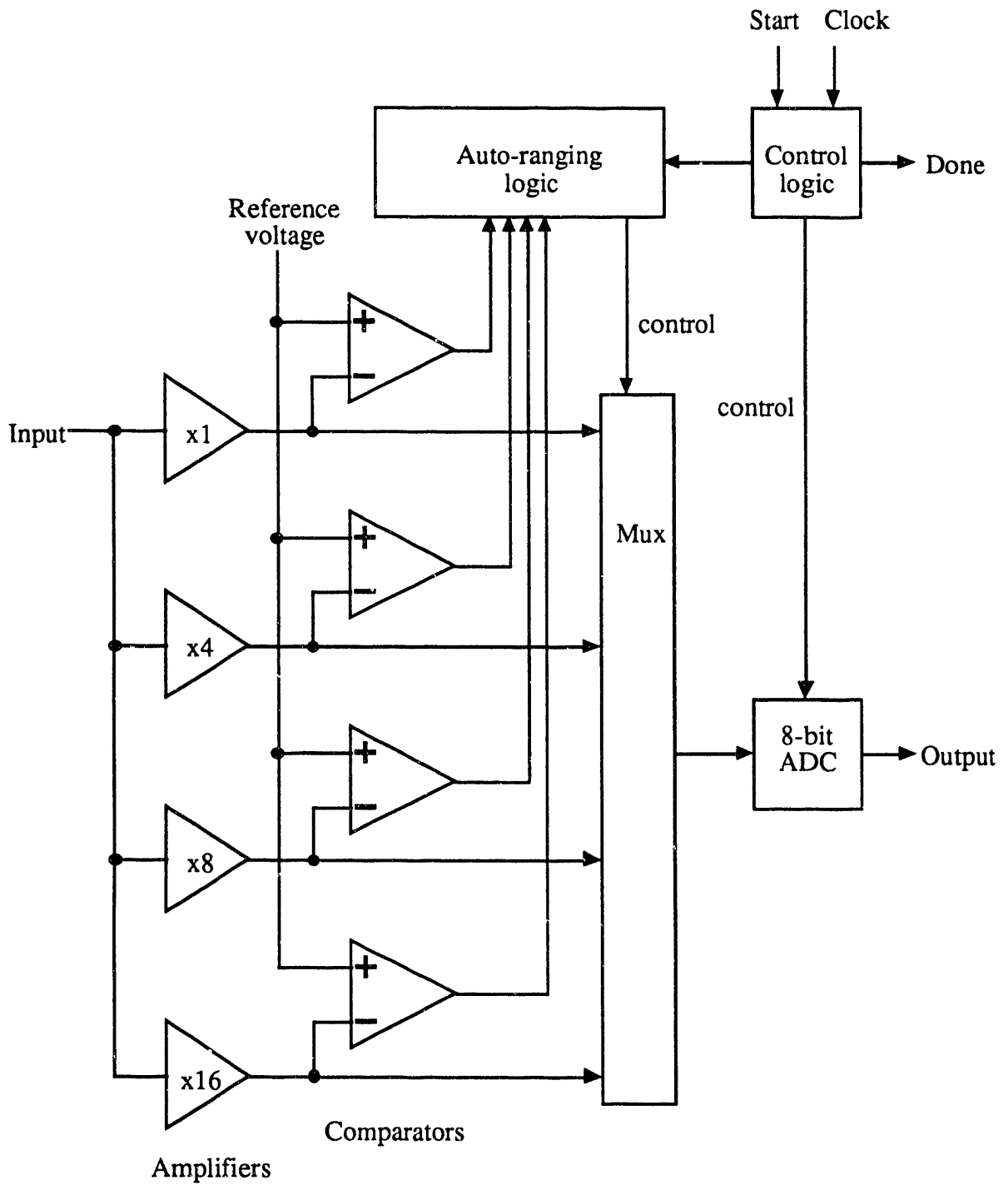**Fig. 4. Auto-ranging ADC block diagram.**

9

distance to the object and d_0 is on the order of a few inches. This type of response is shown in Fig. 5. The prototype sensor has a maximum sensor range of 16 in. in a direction normal to the sensor and approximately 6 in. in the plane of the sensor. The sensor has only minimal response to nonconductive objects such as wood.
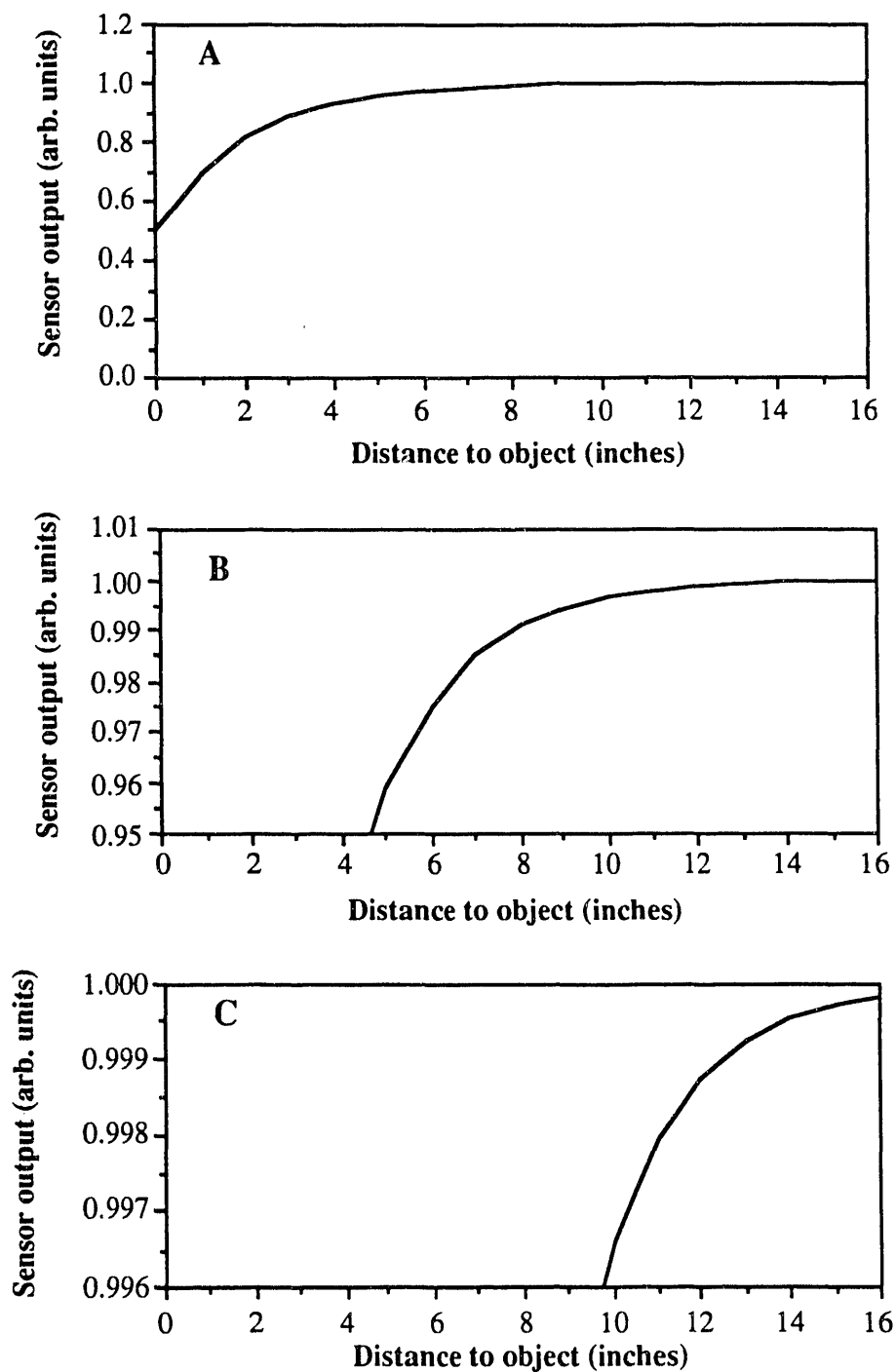

Fig. 5. Sensor response.

3.4.2.2. Analog Signal Processing and Digitization

SNL personnel, responsible for sensor development, have developed several versions of analog signal processing circuits for the capacitive sensor. These circuits use off-the-shelf ICs and are too bulky for use in the sensor node. The SNL circuit will be used as a guide for developing the analog signal processing block. The goal of the development is to duplicate the function of this circuit, but not necessarily the circuit itself.

The capacitive sensor is driven by an oscillator (about 100 kHz), and the SNL circuit uses a charge-sensitive amplifier to detect the current passing through the sensor capacitance (about 1 pF). The motion of an object near the sensor modulates the current. The amplifier output signal is synchronously demodulated and lowpassed to produce a dc or slowly varying voltage that is proportional to the proximity to an obstacle. A variable offset adjustment and additional gain are used to make better use of the dynamic range of the ADC, which is a 10- to 12-bit unit. This process is illustrated in Fig. 6.
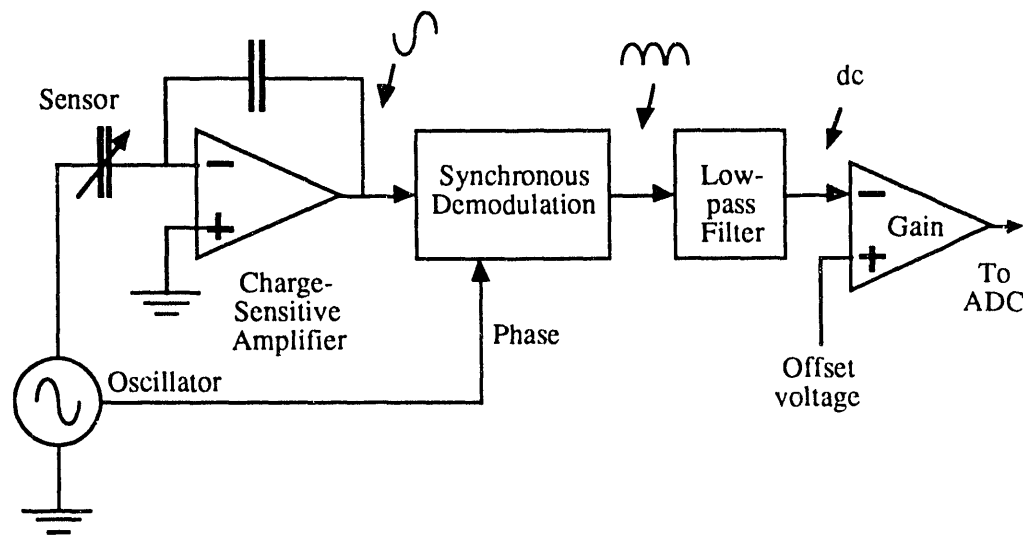
Fig. 6. Analog signal processing block diagram as described by Sandia National Laboratories.

The sensor response is shown in Fig. 7a and 7b along with the effect of the offset adjustment and additional gain. It is important to note that the sensor response varies weakly when the object is near the maximum sensor range. Such small differences in signal level are difficult to detect. The offset adjustment and additional gain are one way to alleviate this problem. These would not be required if an ADC with considerably more than 10 bits resolution could be used or if the sensor was required only to sense distances less than 6 to 8 in. The offset and gain allow the response to extend out to 16 in. at the expense of being able to measure proximity less than about 1 in.

11
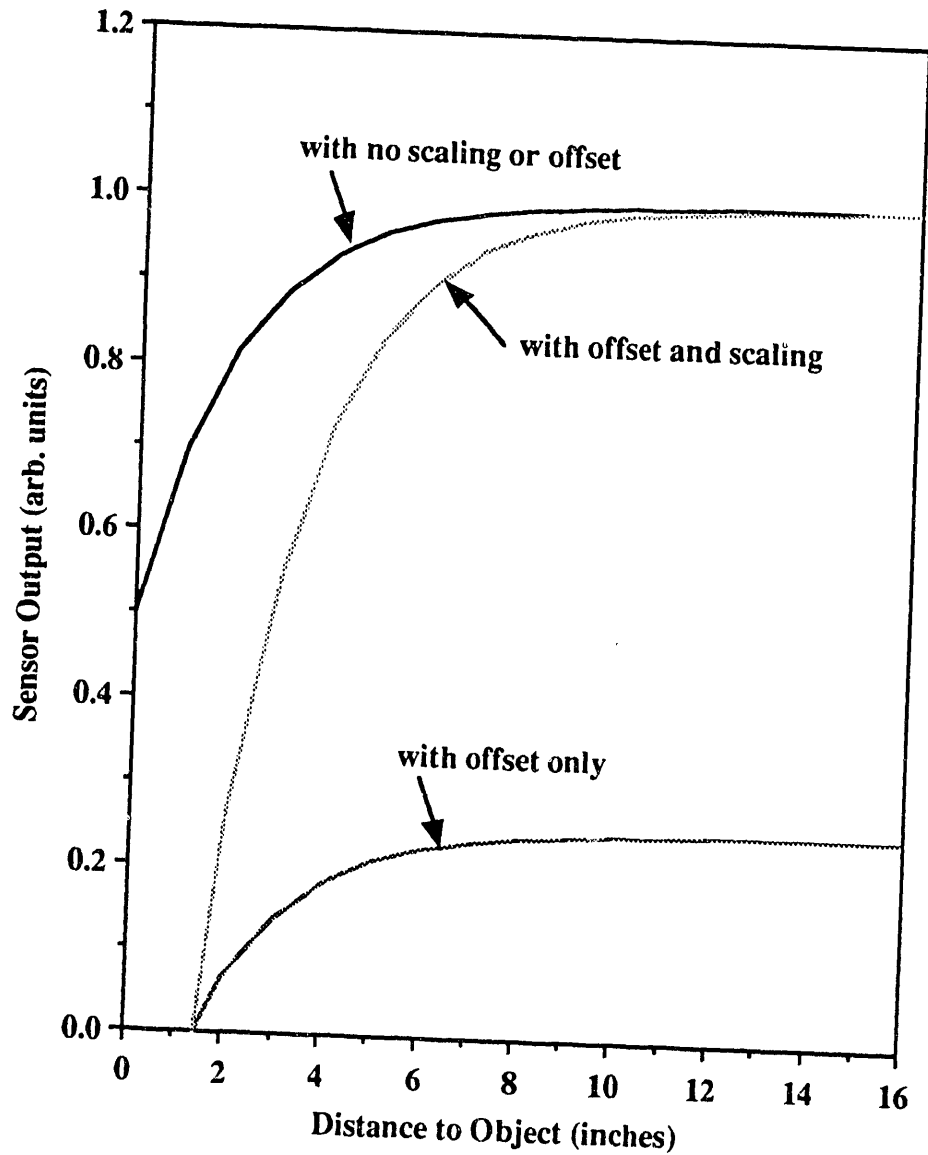
Fig. 7a. Scaled and offset sensor response.

3.4.2.3. Sensor-Specific Electronics for the Capacitive Sensor

The capacitive sensor will require an oscillator and a considerable amount of analog signal processing. The oscillator provides the drive signal for the capacitive sensor. The analog signal processing block detects the signal from the sensor, demodulates, and filters it. This process produces a dc voltage that may be digitized by the sensor node core circuitry.

3.4.2.4. Oscillator

Basic frequency for the oscillator should be about 100 kHz (based on SNL sensor results). SNL used a square wave, but a sine wave would probably be better (no harmonics), and a triangle wave would probably be acceptable. More study of wave shape is needed. The oscillator

12
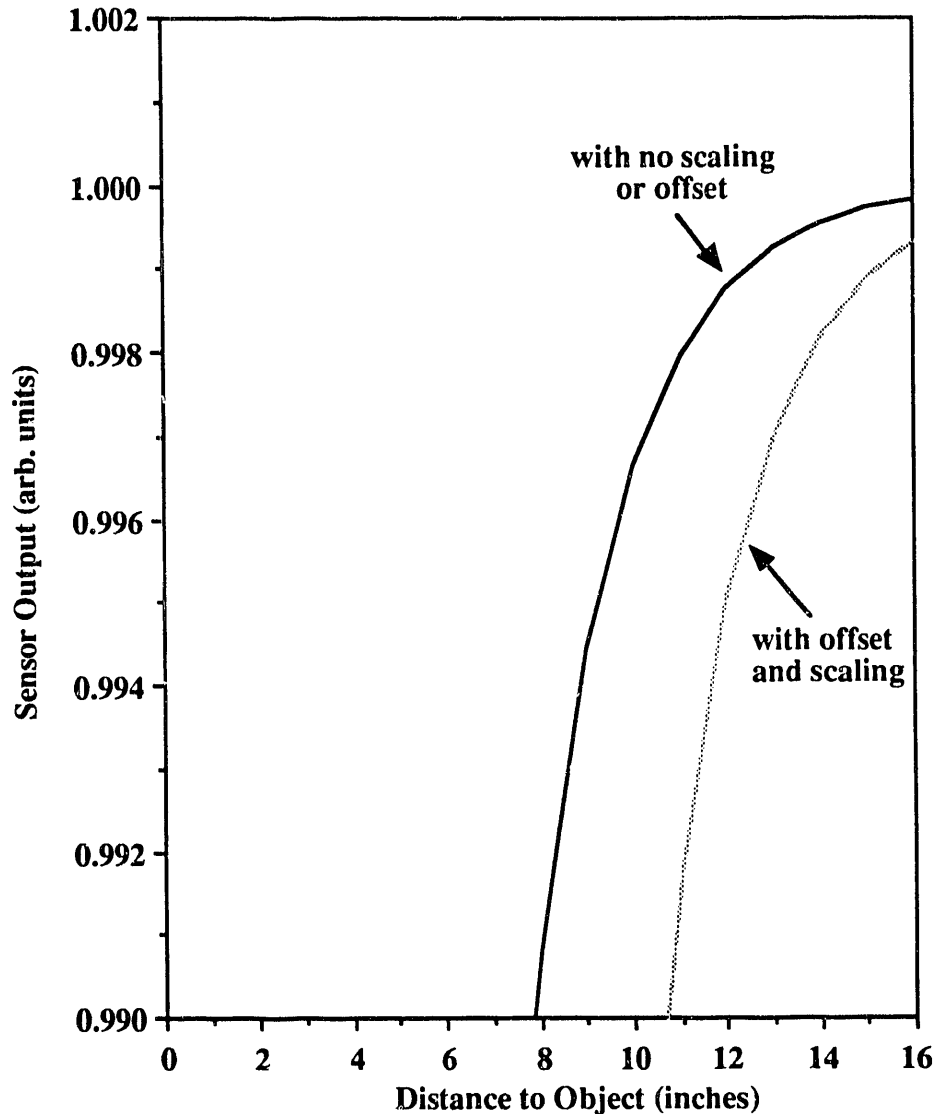
Fig. 7b. Scaled and offset sensor response magnified.

The oscillator amplitude should be as large as possible for best signal-to-noise ratio. For the ASIC, the peak-to-peak amplitude is limited to no more than 10V by supply voltages (±5V). Another requirement is that the oscillator must be capable of driving the sensor capacitance plus wiring and stray capacitances. The sensor capacitance is less than 20 pF (probably about 1 pF plus about 10 pF parasitic), and wiring would probably be on the order of 10 pF, if the sensor and the ASIC are mounted in the same package. Thus the oscillator must drive a total capacitance on the order of 20 pF.

The oscillator frequency might require being programmable to the extent that several frequencies in the range of 80 to 200 kHz are available. Several choices of frequency may be needed to implement a system of many nodes with overlapping sensitive zones. Adjacent

13

sensors need to operate at different frequencies to prevent interference. This feature may not be necessary if the oscillator in a sensor node needs only to be turned on briefly in order to make a proximity measurement.

One way to control the oscillator frequency is to synthesize it digitally from the bracelet processor clock. This clock is likely to be in the range of 2 to 20 MHz, and a wide range of frequencies in the 80 to 200 kHz range could easily be developed. One very simple method would be to multiply the clock period. As an example, a 2 MHz clock has a 0.5 μs period. Multiplying by 20 gives a 10 μs period, which is 100 kHz. Similarly, multiplying by 19 results in 105.2 kHz , multiplying by 18 gives 111.1 kHz, etc.

The oscillator amplitude stability should be very good. For example, a 1% change in oscillator amplitude would be interpreted as a 1% change in the sensor signal and that may represent a significant change in apparent proximity. Several approaches can provide a stable amplitude or correct for variations. One approach would base the oscillator on a stable reference voltage. Another would use a feedback control loop to stabilize the oscillator output level. An alternative would be to correct for variations by normalizing the detected signal with the applied signal.

The stable reference voltage approach is preferred because it is the most practical and simply implemented. A reference voltage is readily available because one is required for the ADC in the sensor node core. This voltage can be buffered and used as a supply for the oscillator output stage, which can be a simple inverter driven directly by the digital frequency synthesizer.

3.4.2.5. Analog Signal Processing

The analog signal processing for the capacitive proximity sensor will incorporate the blocks shown in Fig. 8. All of these blocks can be implemented in an ASIC with only a few external components. The charge-sensitive amplifier will use an opamp with external feedback resistor. Variable gain will be used to increase the signal level for more effective demodulation and to
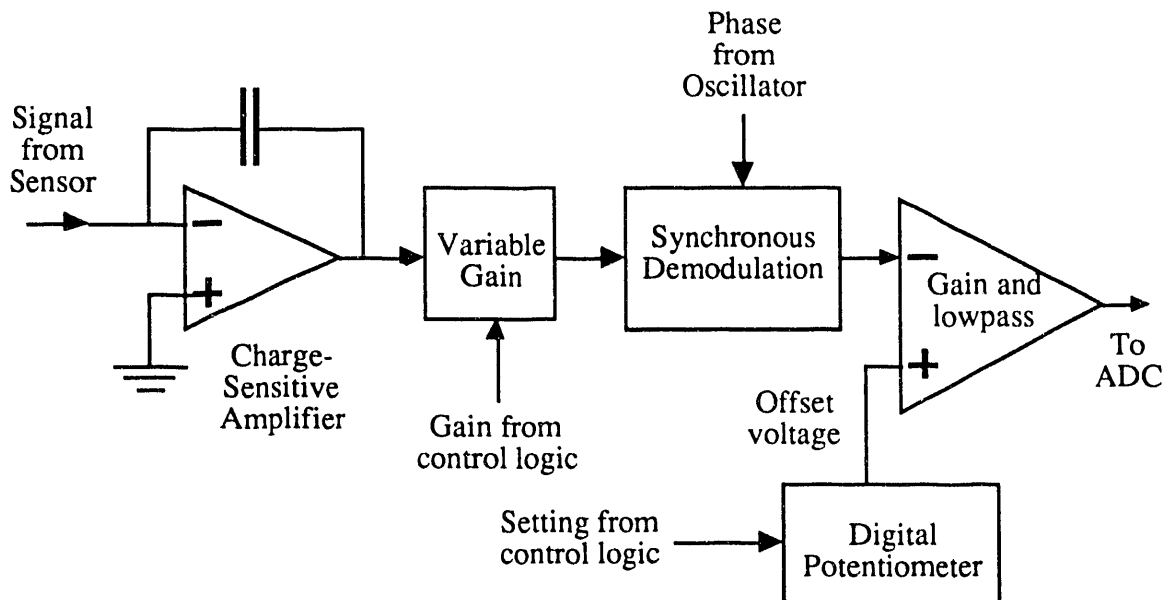

Fig. 8. Analog signal processing circuit block diagram as envisioned for application-specific integrated circuits.

14

allow for a range of sensor capacities. Several gains (x1, x2, x3 and x5) will be digitally programmable. The synchronous demodulation and low pass filtering will be achieved with a circuit, like that shown in Fig. 9. Additional lowpass filtering may be necessary and will be provided for. The offset function will be accomplished with a 5-bit digitally programmable potentiometer.

3.4.2.6. Analog-To-Digital Converter Considerations

The minimum number of bits (resolution) required of the ADC is very sensitive to the sensor response, the minimum differential distance that must be resolved, and the maximum distance. (The sensor response described here is approximate and has not been measured yet.) Fig. 7a provides an example of the sensor response for a hypothetical case where the output is $1-0.5 \exp(-d/2)$, where d is the distance to the object in inches. If the maximum distance is 16 in., and the signal processing electronics must resolve the difference between an object located 14 in. away and the same object 16 in. away, then the ADC must have at least 0.025% of full scale resolution or in other words, must be better than 12 bits. If a gain of 4 (and appropriate offset) can be used, the requirement drops to 0.1% of full scale and 10 bits, and for a gain of 16 (and appropriate offset), 0.4% resolution or 8 bits are required. The effect of an offset and a gain of 4 is illustrated in Fig. 7b. The requirements for the actual sensor may be more or less stringent.
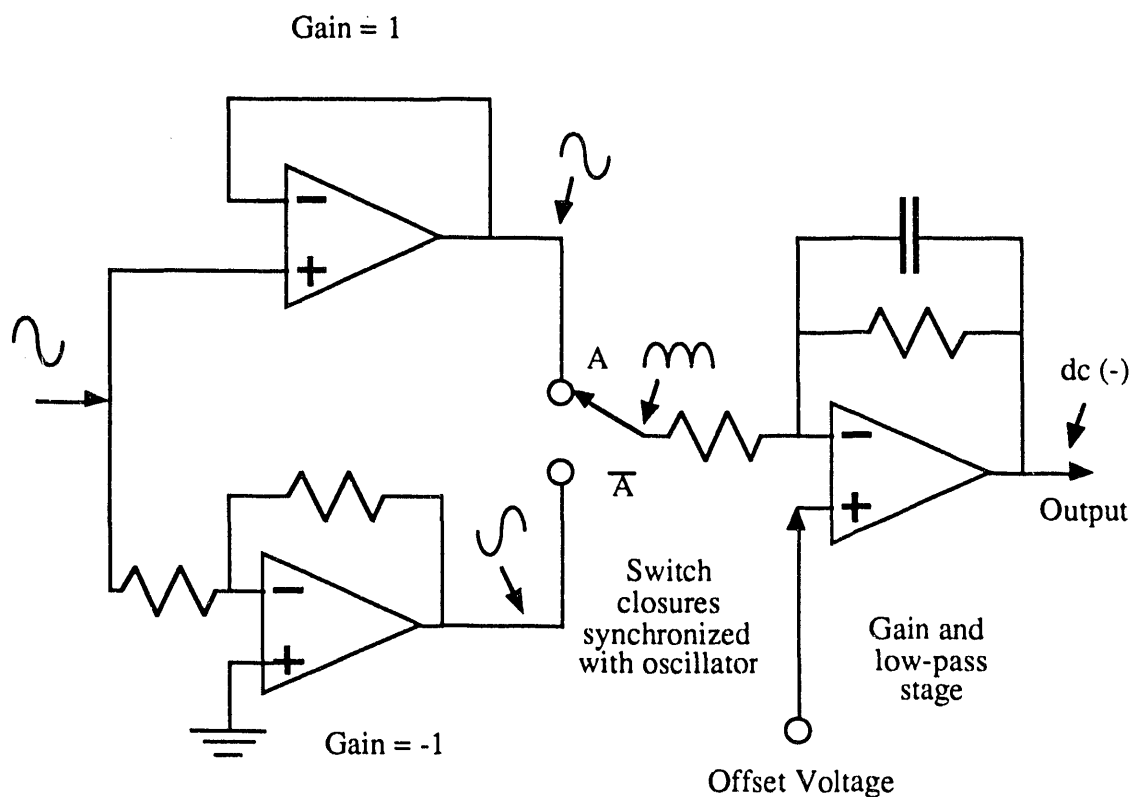

Fig. 9. Synchronous demodulation circuit block diagram concept for application-specific integrated circuits.

3.4.2.7. Operation of the Sensor Node with a Capacitive Sensor

15

The system's hardware and software would duplicate the gain and offset procedure of the SNL signal processing chain. The sensor node operation would proceed in this fashion: with no object near the sensor, the gain would be set to one and the offset to zero. (Refer to Fig. 10, curve a.) The ADC would convert the output of the analog signal processor (no object signal) and the gain would be set to the highest value that would result in an on-scale value. (ADC full scale is 3V.) (Refer to Fig. 10, curve b.) The ADC would again convert the output and the offset would be used to set the output to approximately 0.5V by adjusting the digital potentiometer. (The exact value depends upon the sensor characteristics and the range of distances of interest.) This would scale the signal so that an object far from the sensor would produce a signal near 0.5V (Fig. 10, curve c). A nearby object would produce a smaller signal. This completes the setup phase. When called upon to make a proximity reading, the autoranging ADC circuit would first determine the highest gain amplifier whose output was not off-scale and connect that output to the ADC by way of the mux, and then start the conversion process. This essentially adds more gain to the response curve (Fig. 10, curve d).
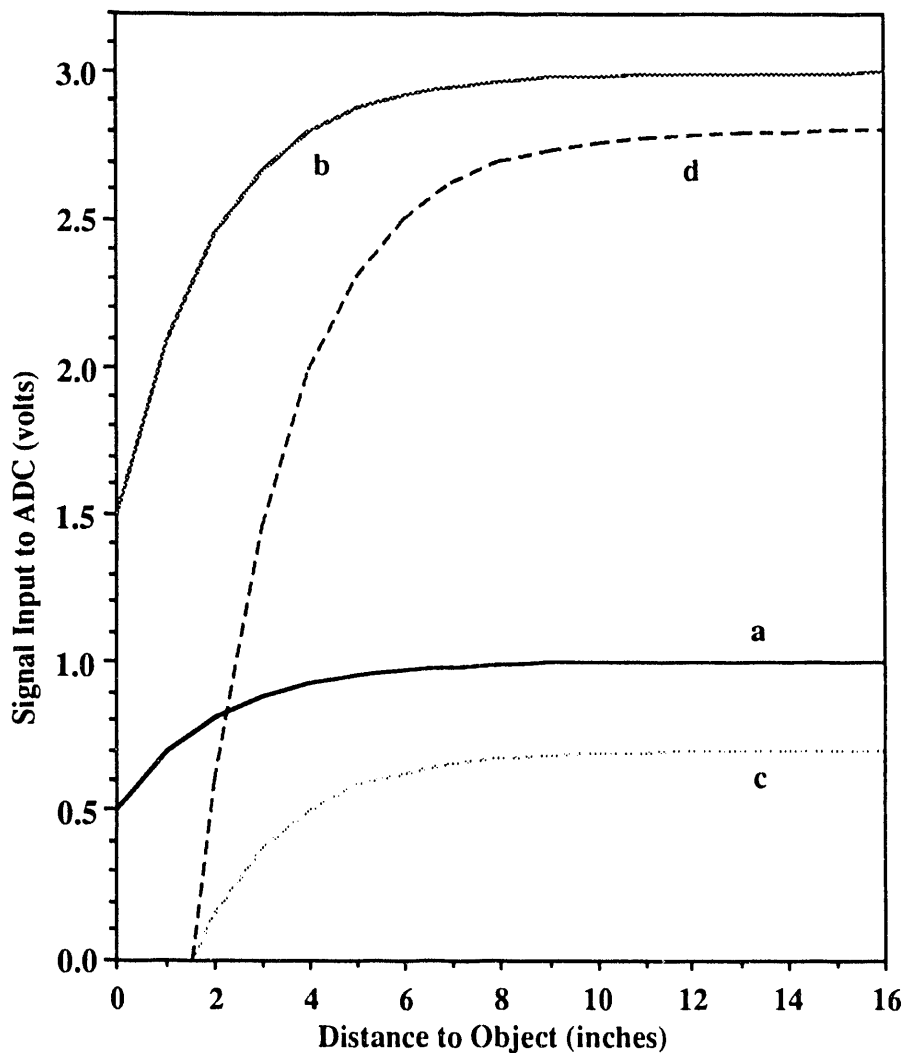

Fig. 10. Scaled and offset response.

3.5. COMMUNICATIONS

3.5.1. Host-to-Bracelet Processor

A high-speed multi-drop communications link exists between the host and bracelet processors. If possible, this communications link should use an existing protocol. The communications data rate must be able to multiplex all sensor data from each bracelet onto a multi-drop communications link. A multi-drop configuration is required to handle a number of bracelet/sensor assemblies.

Either a wired system or a fiber-optic system can be used. A wired system is more prone to short circuits while a fiber-optic system cannot be shorted. Note that the power must be a wired line and that it is sensitive to shorts as well as breaks. In other words, the entire system should be able to recover from shorts and breaks, including those on the host-to-bracelet power lines.

Some degree of redundancy must be realized in anticipation of cable breakages and bracelet failures. One alternative is to provide two independent communications links from each bracelet to the host. If one link is broken, the other link can be used for data transmission. This scheme only provides redundant communications links, but not redundant bracelet processors and sensors.

A second possible solution to the need for redundancy is to utilize a single link that is looped around the length of the arm as shown in Fig. 11. This provides a single communications link with redundant communications paths. The problem with this scheme is that it handles only
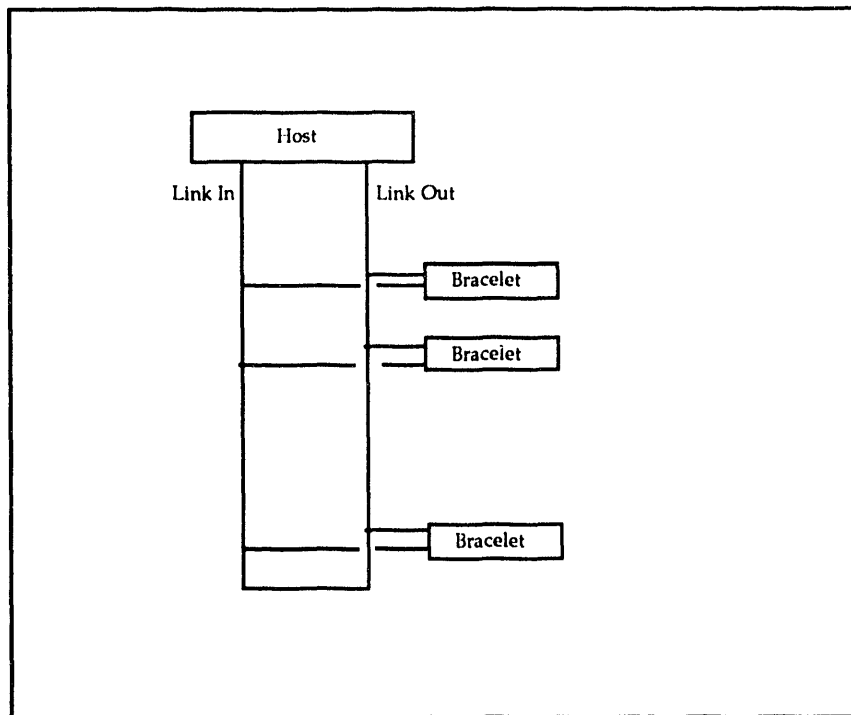

Fig. 11. Single communications link loop.

communication link breakages, but not short circuits. A single short circuit will disable the entire system. However, the problem of short circuits could be solved by the addition of an active communications circuit at each bracelet.

A third scheme, which is the preferred one, is to utilize dual communications links addressing alternate bracelets. A breakage in one link affects only the bracelets below it, on the arm. A short circuit will affect the entire link. This scheme has several advantages. One is that communications-bracelet-sensor systems are completely redundant within the cell. Another advantage is that by utilizing different communications links for separate "phases," the net data rate in a working system is greatly increased. See Fig. 12 for a rough diagram of the proposed redundant communications scheme.

The communications protocol should be a simple nonmodal protocol. All synchronization should be performed by the host processor, with the bracelets acting as slaves on the multi-drop bus. A rough sequence of communications between the host and bracelet is now discussed. The host initiates all communications to the multi-drop bracelets. This performs synchronization for all bracelets. The host then polls each bracelet for its status. The status packet contains the following information for each sensor:

> Caution flag
> Halt flag
> Error flag (set if there is a sensor error or sensor not recognized)
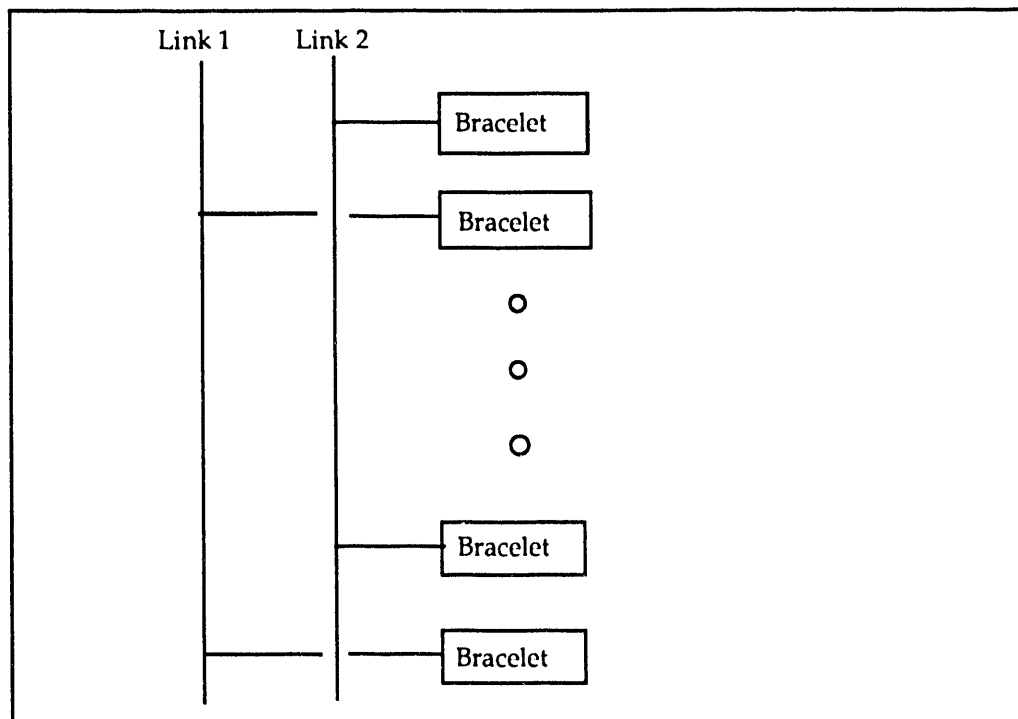> Data-follows flag (set if data is to follow at the end of the packet)

Fig. 12. Dual communications links addressing alternate bracelets.

18

After the status packet, new data are received for all sensors that have set their "data-follows flag." In this case, no address is needed because the host code assumes a predetermined order. Note that the data-follows flag is set only when a caution or error condition is met. To obtain all data continuously, the caution or halt conditions are set to a minimum, forcing continuous transfers.

A simple command structure is used to send control and parameter information to the bracelet processors. During normal operation, the command is to acquire and threshold sensor data. However, commands exist to set gain parameters, set operational modes (disable/enable), set caution, and halt thresholds, etc.

3.5.2. Bracelet Processor to Sensor Node Electronics

Communications between the sensor nodes and the bracelet controller will allow a number of functions to be performed. Transfers of data from the bracelet controller to the nodes will deliver commands and setup parameters needed for node operation. Transfers of data from a node to the bracelet controller will allow the reading of data, sensor ID, and sensor status.

The communications interface between the bracelet controller and sensor nodes will be a bi-directional serial link with the controller as master and nodes as slaves. Data will be transferred in packets of 8 bits. This approach will allow compatibility with most microcontrollers and allow simple implementation of the bracelet controller communications functions in an ASIC, should that be desired.

The bracelet processor and sensor node communication interface will be composed of a 4-signal serial interface (see Fig. 13). These signals are common to each bracelet and are defined as follows:

- SERIAL I/O - Line for synchronized bi-directional serial data transfer
- SHIFT CLK - Serial data clock
- WR/RDB - Indicates direction of serial data transfer
- STROBE - Used to latch data within sensor nodes
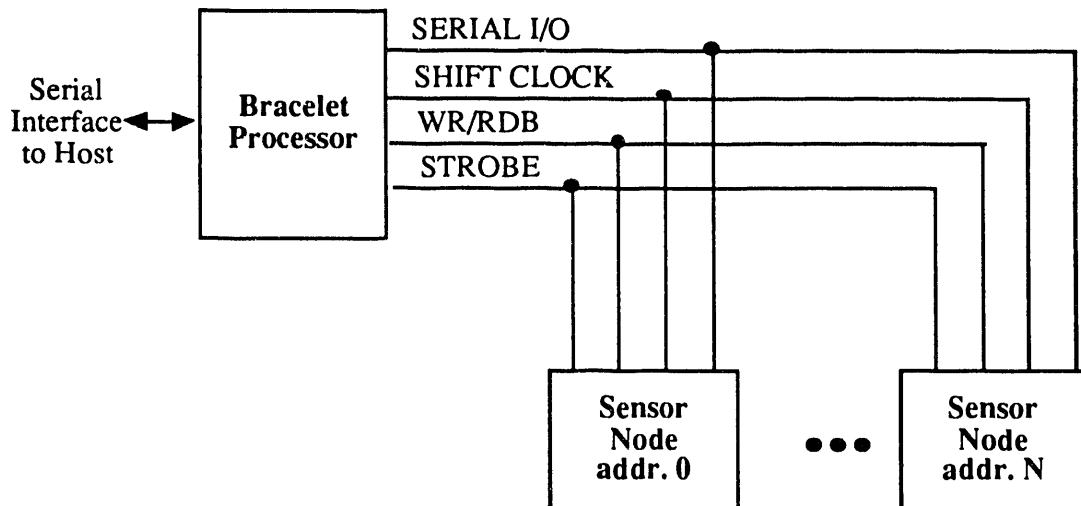
Fig. 13. Bracelet communications block diagram concept.

19

A block diagram of the serial interface to be incorporated into the sensor node ASIC (using standard cells) is shown in Fig. 14. The architecture is based on switching the serial data line between two shift registers (one input and one output), using the WR/RDB line as a direction indicator, and shifting data using the SHIFT CLK. STROBE is used to latch data to be transferred in or out of the appropriate shift register.

The input shift register is divided into three sections: sensor address, register address, and data. The sensor address is used to identify which of the bracelet sensor nodes is being addressed. This address is locally compared to a reference address (uniquely set for each sensor on a bracelet) and an address enable is asserted upon receipt of the proper address. Five register address bits will be implemented allowing for as many as 32 sensor nodes per bracelet processor. In some instances improved speed can be obtained by writing some control commands to all nodes on a bracelet simultaneously. To provide this capability one address will be reserved for a global write, which will reduce the maximum number of sensor nodes to 31. The register
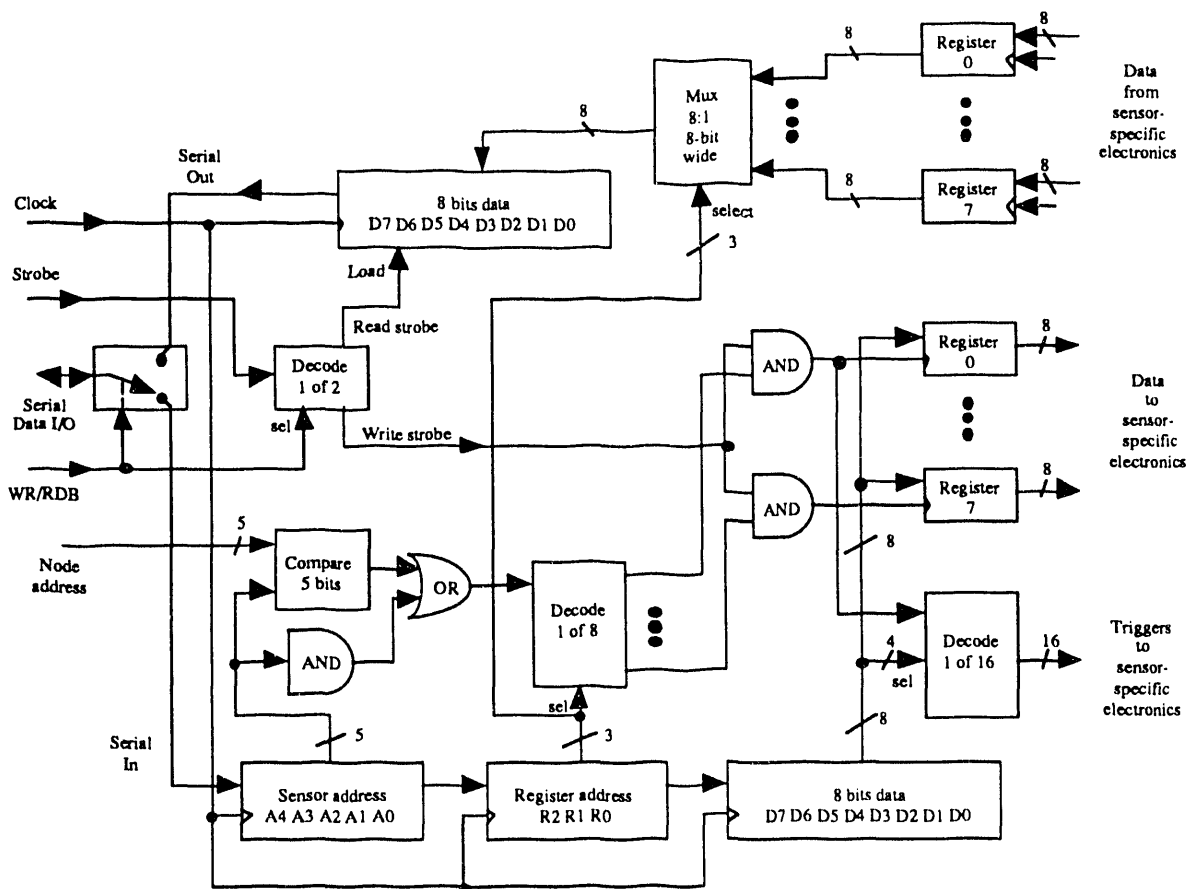
Fig. 14. Node serial communications block diagram concept
for application-specific integrated circuits.

20

address bits (3) will be used for selection of one of eight on-chip registers to be used for sensor node calibration and control. Data to be written into one of the registers is shifted into the data byte of the input shift register and subsequently loaded into one of the eight sensor node ASIC control registers.

A data write is accomplished by initiating the timing sequence detailed in Fig. 15a. Two 8-bit data bytes are shifted globally to the sensor nodes with the WR line asserted. As previously mentioned, the first byte contains the data, the second the address bits for the sensor node and selected control register. Within the addressed node, assertion of the STROBE signal transfers data from the input shift register data byte into the selected sensor node ASIC control register. All registers used in each of the sensor node ASICs will be configured using this procedure.

A data read is similarly accomplished using the timing sequence shown in Fig. 15b. A write operation is first performed to write only the address byte into the input shift register. This byte, as before, is used to select the sensor node and associated data register to read. After a byte
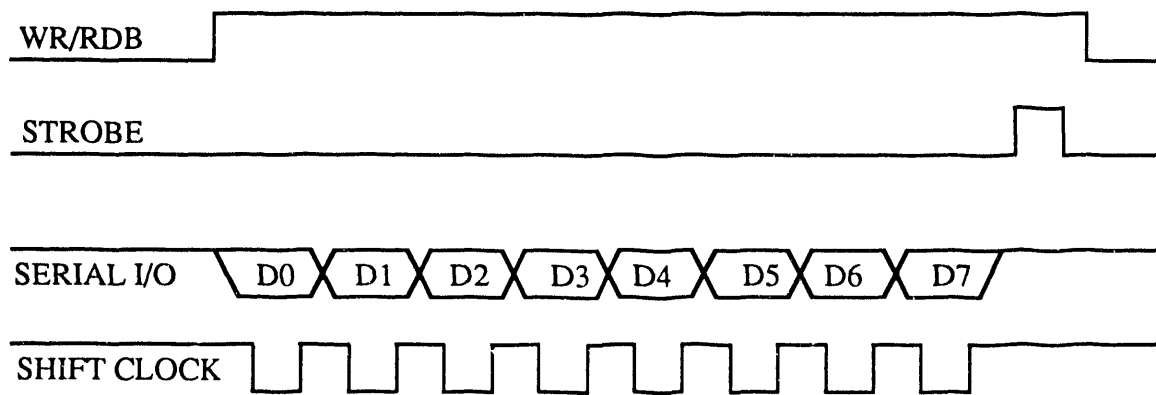
(a) Data Write
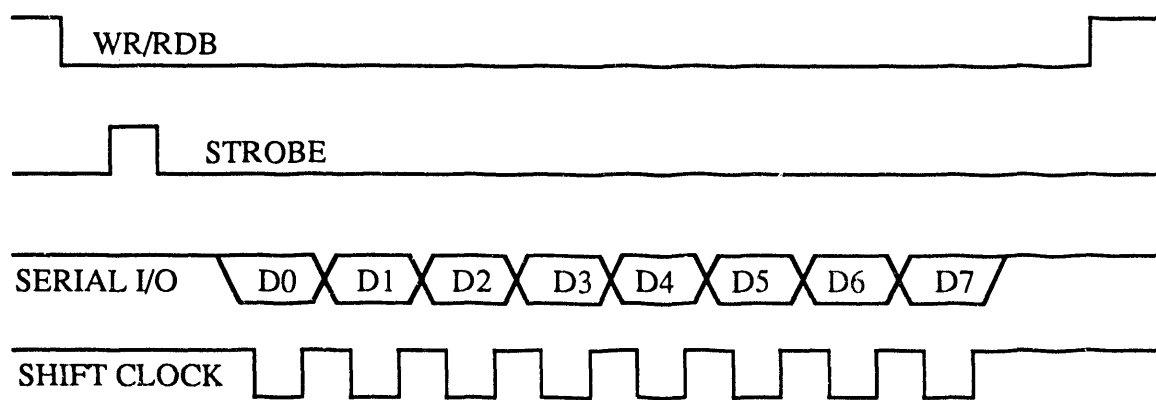
(b) Data Read (must be preceded by Data Write)

Fig. 15. Bracelet communications timing concept.

21

write, the data is then read by assertion of the WRB line (indicating a read operation), asserting the STROBE line which loads the selected byte into the output shift register, and then serially shifting the data out using the SHIFT CLK. After all data have been removed, the RDB line is de-asserted.

For some control purposes, it will be necessary to supply pulses to the sensor electronics. An addition to the data write process allows the generation of pulses with the same addressing capabilities. As in the data write, two 8-bit data bytes are shifted globally to the sensor nodes. The address is decoded to produce an enable and the STROBE signal transfers the data into the addressed register. In addition, if the register address is 0, a decoder is enabled which allows passing the STROBE signal to one of 16 outputs that are selected with the upper 4 bits of the data. Thus writing a certain data byte to register 0 will also produce a pulse that can be used to start a process in the sensor node. For example, the pulse could be used to start an analog-to-digital converter.

Selection of a serial interface was driven by the need to reduce the number of bus lines required for operation, thus making it less complex to fabricate and potentially better suited for use in harsh environments. The number of address lines selected will allow for as many as 31 sensor nodes, each having up to 8 read and 8 write registers each. This architecture will be implemented using standard cell techniques and should allow ample flexibility for the incorporation of other sensors and associated circuitry as the need develops.

4. MECHANICAL DESIGN/PACKAGING

4.1. BRACELET

The following paragraphs describe the mechanical design/packaging concept for the prototype bracelet elements and possible concepts for an environmentally hardened version.

4.1.1. Sensor Node Mechanical Concept for Prototype System

A concept for the mechanical arrangement of the sensor node is shown in Fig. 16. A small housing will contain the sensor and the electronics and provide a way to attach the node to the arm. It is anticipated that most, but not quite all, of the electronics needed for the node can be incorporated into the ASIC. A small printed-circuit board will hold these components and a connector. In the case of the capacitive sensor, it may be possible to use a multilayer printed-circuit board with the upper layers comprising the sensor and the lower layers comprising the circuitry. The ASIC and other electronic components would be surface-mounted on the bottom layer. The housing would be approximately the size of the capacitive sensor (6 cm by 6 cm) with a thickness of less than 1 cm.

4.1.2. Bracelet Processor Mechanical Concept for Prototype System

The bracelet processor would be similar to a sensor node except that instead of a sensor it would include a connector for linking it to the host. Because the bracelet processor includes a processor circuit, power conditioning circuitry, and the host communications interface circuitry, it may require a larger housing than that of a sensor node.

4.1.3. Bracelet Wiring Concept for Prototype System

A ribbon cable with mating connectors will be used to connect nodes and the bracelet processor. Mechanical configuration of a bracelet will consist simply of attaching (probably by crimping) connectors to the ribbon cable at the locations where nodes are needed. The bracelet processor will be attached at one end of the cable. For additional mechanical protection, the wiring of the bracelet may be jacketed after assembly or covered with a flexible layer of a material such as PTFE. The allowable bend radius of such a bracelet should be 2 mm or less (for the cable.)

4.1.4. Bracelet Modifications for a Hazardous-Environment System

The prototype system is designed for operation in a laboratory-type environment (moderate temperature, moderate humidity, no chemical vapors present, etc.) For an obstacle avoidance system designed to meet stringent environmental requirements, modifications to the bracelet mechanical design will be required. The modifications needed depend upon the environment expected.

RIBBON CABLE
CONNECTOR
HOUSING
SENSOR
B
CONNECTOR
PC BOARD
ASIC
ELECTRONIC
COMPONENTS
A
PC BOARD
ASIC

A. Top view
B. Side view -- cross section
C. Top view -- cross section

C
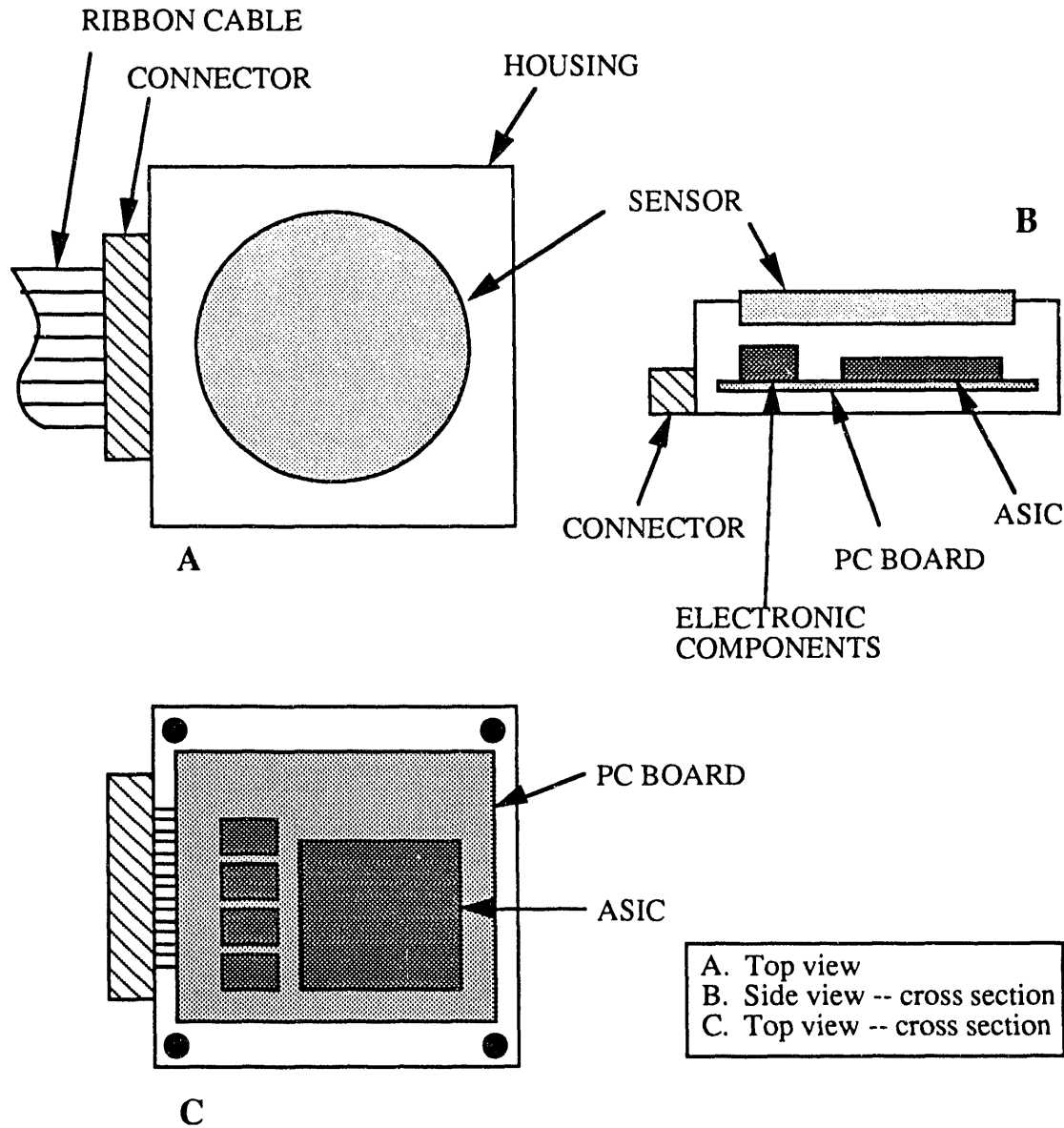
Fig. 16. Conceptual mechanical layout of sensor node for bracelet system.

4.1.4.1. Chemical Vapors and Humidity

A con. non failure of electronics exposed to chemical vapors or high humidity is loss of circuit continuity caused by corrosion. Connectors are especially susceptible to this problem. There are three likely options to prevent corrosion. The first option would be to upgrade the bracelet (processor, nodes, and cabling) to use hermetically sealed connectors. A second option would be to encapsulate connectors such as those used in the prototype system. Hermetically sealed connectors would be considerably more expensive, more difficult to install, and probably larger, but they should be extremely reliable. Encapsulation of lower-cost connectors may require considerable labor, may be less reliable, and would make repairs or reconfiguration of a bracelet very difficult.

A third option to protect the electronics from a corrosive environment would be to develop a flexible printed circuit board on which the capacitive sensor could be included as a structure and the sensor node electronics could be mounted. After mounting, an additional layer could be added to seal the bracelet. Drawbacks to this approach would be that the bracelet is not reconfigurable and repairs would likely be difficult. Bracelets of this type may need to be customized to a particular arm.

Each of the three options has advantages and disadvantages. Of the three options, the second, involving encapsulation, appears to be the most desirable for an "add-on" whole-arm obstacle avoidance "kit," assuming the labor associated with encapsulation can be minimized, an encapsulation material can be identified that can withstand the radiation and temperature requirements while remaining relatively flexible, and the potential reliability issue can be resolved. Further investigation of the three options is needed.

4.1.4.2. Radiation and Temperature

Materials used in the bracelet construction would be chosen to be able to withstand the expected range of operating temperatures and the expected radiation dose.

4.2. HOST-TO-BRACELET COMMUNICATIONS LINK AND POWER CABLING

The host-to-bracelet communications link and power cabling for the prototype system will be commercially available materials. In a retrofit situation (which is the case of the demonstration of the prototype system), these will be attached to the outside of the links with loops to allow flexing at the joints. In a newly designed manipulator they should be routed internal to the linkages if possible. The communications link can be a fiber-optic link if the environmental requirements (radiation primarily) can be met.

5. SOFTWARE CONCEPTUAL DESIGN

5.1. HOST

A host processor provides the interface between the bracelet processors and the manipulator control system. All setup and calibration of sensors and data handling is controlled by this host processor. The host acts as a primary link between the bracelet communications links and an existing robot control system. Several host interfaces provide this interface with the highest degree of flexibility. These interfaces are a common memory interface, a function library interface, and a GISC communications interface. Also provided is a terminal interface that can be used for setup and configuration. By using the software scheme described below, a collision avoidance system can be created to be an integral part of a manipulator control system or be created to be a GISC subsystem, independent of the manipulator control system. A "terminal interface" also provides off-line configuration for setting up sensor parameters and testing. Appropriate parameters are stored in a disk file.

The host software provides an interface from the host to bracelet communications protocol and a common memory/function library interface. As much as possible, details of the specific communications mechanisms are hidden by the common memory/function library interface. A host processor task is set up to control the bracelet communications link and to place returned data and flags in the common memory. Control functions (such as setting parameters) enter data into a communications queue that is sent out during the next communication cycle.

The host software task is set up to run at a specified loop rate. This software will initiate communications with the bracelets and wait for a response. As the bracelet data is returned, the host task parses it and places it in the common block. If a command comes from another task, possibly through a data queue, it is sent to the bracelet processors at the start of each loop. When no new command is pending, a standard acquisition command is sent. Since all data is placed into the common block, other tasks can access the caution and halt flags directly. The host task also combines the flags for all bracelets into a flag set for each bracelet and a single flag set for all bracelets.

5.1.1. Common Memory Interface

A common memory interface is provided to allow the system read access to individual sensor information directly. A hierarchical structure is used to contain globally available obstacle avoidance data. A single structure conveys information related to an entire arm. This information may include the following:

> Number of sensors
> Number of bracelets
> Acquisition loop rate
> Overall caution state
> Overall halt state
> Number of sensors in caution state
> Number of sensors in halt state
> Current time

26

For each bracelet, a structure is defined to include the following information:

> Bracelet location on arm
> Bracelet status (active/inactive)
> Number of sensors on bracelet
> Number of different types of sensors on bracelet
> Bracelet caution state
> Bracelet halt state

For each sensor, a C language structure defines various attributes. These attributes will include the following information:

> Sensor location and orientation on the manipulator
> Bracelet number
> Sensor type
> Caution detection threshold
> Halt detection threshold
> Current state of sensor (active/inactive/not functional)
> Current state of caution detection threshold
> Current state of halt detection threshold
> Last time of data update
> Sensor data value
> Any miscellaneous sensor specific configurations

5.1.2. Function Library Interface

A function library is provided to allow software code to access all parameters of the obstacle avoidance system. For read-only information, the functions are built on top of the common memory interface discussed above. In addition to these "read-only" functions, more functions are provided to assert control over the obstacle avoidance system. A rough outline of these functions is presented below:

> System functions
> > Initialize system (reads position and orientation information from disk file)
> > Save configuration (to disk file)
> > Read configuration (from disk file)
> > Set acquisition loop rate
> > Reset system avoidance flag
>
> Bracelet functions
> > Initialize bracelet
> > Save configuration (to disk file)
> > Read configuration (from disk file)
> > Reset bracelet avoidance flag
>
> Sensor functions
> > Initialize sensor (also reads sensor type)
> > Save configuration (to disk file)
> > Read configuration (from disk file)
> > Set sensor position and orientation
> > Set sensor state (active/inactive)
> > Set caution threshold
> > Set halt threshold
> > Set miscellaneous sensor specific configuration

27

5.1.3. GISC Communications Interface

The GISC communications interface provides an obstacle avoidance subsystem that can be part of a larger GISC system. All functions provided in the function library mentioned above are also provided as GISC functions. This provides an easy implementation path for GISC compatibility and also provides consistent functionality between the GISC, function library, and common memory interfaces.

5.1.4. Terminal Interface

An ASCII terminal interface provides access to all of the functions in the function library and all data values in the common memory for debugging and configuration purposes. This terminal interface is built on top of the function library interface or the GISC interface. This provides complete testability for the highest level interfaces while minimizing the effort required to create a terminal interface. System, bracelet, and individual sensor information can be viewed and modified in a user-friendly manner.

5.2. BRACELET INTERFACE ELECTRONICS SOFTWARE

The bracelet processor provides the interface between the host processor and the sensor nodes. It serves as both a slave to the host processor, and as a master to the sensor nodes. The bracelet processor software must accomplish two tasks: communications and data reduction. Most of the data reduction will be performed by the host processor, minimizing the complexity required at the bracelet processor level.

5.2.1. Communications

Communications may occur between the host and the bracelet processor, the bracelet processor and the sensor nodes, or the host and the sensor nodes. In the last case, the bracelet processor serves only to route communications between the host and the sensor nodes. Because different communications protocols are used, the bracelet processor must reformat data before retransmitting data to the intended recipient. Thus writing data to a sensor node involves three steps on the part of the bracelet processor: receiving data from the host, reformatting the data, and transmission of data to the sensor node. A similar procedure is followed to read data from a sensor node.

Operations involving the sensor nodes are carried out using two basic functions: write data and read data. The write data function transfers data from the bracelet processor and to the sensor nodes. Likewise, the read data function transfers data from a sensor node to the bracelet processor. More descriptive variations on these basic functions include:

> Read proximity value
> Read sensor type
> Read sensor status
> Read sensor ID
> Read sensor setup (for verification)
> Write setup parameters (global and local)
> Write trigger (global and local)

It is important to note that the same data may be passed to all nodes on a bracelet simultaneously (a global write).

5.2.2. Data Reduction

The bracelet processor is responsible for one type of data reduction. Proximity values accumulated from the sensor nodes are compared to threshold values downloaded from the host, and flags are set whenever a proximity value exceeds its threshold. These flags are grouped and readied for transmission upon request by the host.

5.2.3. Types of Operations

Several types of operations are available, and these include setup, calibration, and thresholding. Each operation requires the use of several primitive functions. Additional operations are possible by using other sets of primitives. Note that these operations require cooperation between host and bracelet software.

5.2.3.1. Setup

For a setup operation, the host must send data through the bracelet processor to a single node or to all nodes on a bracelet. At the bracelet processor level, this is a write setup operation. In the case of the capacitive sensor, the A/D converter must be triggered (write trigger), and then the data must be read (read proximity). These steps are repeated several times to complete the setup procedure. For other types of sensors, the setup procedure may be more or less complicated.

5.2.3.2. Calibration

In the calibration operation, several proximity readings are taken and the results are stored by the host. To carry out this operation, the host must trigger a proximity measurement (write trigger) and then read out the data (read proximity). This procedure is repeated as needed to develop a calibration curve for each sensor.

5.2.3.3. Thresholding

Thresholding is the normal mode of operation for the obstacle avoidance system. In this operation all sensor nodes on a bracelet are triggered simultaneously (write trigger global) and then the bracelet processor collects data sequentially from each of its nodes (read proximity). This data is compared to the thresholds stored in the bracelet processor's memory and the results of the comparison are formatted and held ready for transmission to the host.

Thresholding the data at the bracelet processor level results in two benefits. First, because less data must be transmitted to the host, the sensor update rate is faster for a given serial link baud rate. Second, some of the data reduction is off-loaded from the host to the bracelet processor, which reduces the hosts' computing burden.

29

5.2.3.4. Other Operations

Operations such as a scan of all sensor proximity readings are possible. Because this operation simply involves triggering proximity measurements and then reading out the data, it is actually much the same as the calibration operation. The scan could involve all sensors or only a programmed (at the host level) subset of interest.

6. REFERENCES

J. L NOVAK and J. T. FEDDEMA, "A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance," Proceedings of 1992 IEEE Int. Conference on Robotics and Automation, Nice, France, May 1992.

INTERNAL DISTRIBUTION

1. G. A. Armstrong	26. D. W. McDonald
2-11. S. M. Babcock	27-29. S. A. Meacham
12. J. B. Berry	30. M. W. Noakes
13. H. R. Brashear	31. K. E. Plummer
14. C. L. Britton, Jr.	32. B. S. Richardson
15. B. L. Burks	33. S. L. Schrock
16. P. L. Butler	34. J. O. Steigler
17. M. N. Ericson	35. A. L. Wintenberg
18. M. J. Haire	36. H. R. Yook
19. D. C. Haley	37. ORNL Central Research Library
20-21. W. R. Hamel	38. ORNL Document Reference Section
22. J. N. Herndon	39. ORNL Laboratory Records
23. R. L. Kress	40. ORNL Laboratory Records RC
24. A. P. Malinauskas	41. ORNL Patent Section
25. R. C. Mann	

EXTERNAL DISTRIBUTION

42. Clinton Bastin, Manager, LMR Reprocessing Projects, Division of Fuels and Reprocessing, Office of Facilities, Fuel Cycle, and Test Programs, NE-471, Department of Energy, Washington, DC 20545

43. D. W. Bennett, Pacific Northwest Laboratories, P.O. Box 999, Richland, WA 99352

44. L. F. Blankner, Fusion and Nuclear Technology Branch, Energy Programs Division, Department of Energy, P.O. Box 2008, Oak Ridge, TN 37831-6269

45. S. A. Couture, Lawrence Livermore National Laboratory, P.O. Box 808, L-437, Livermore, CA 64551

46-50. R. W. Harrigan, Sandia National Laboratories, P.O. Box 5800, Division 1414, Albuquerque, NM 87185

51. D. L. Herman, Fernald Management Company of Ohio, P.O. Box 398704, Cincinnati, OH 45239-8704

52. R. M. Hollen, Los Alamos National Laboratory, P.O. Box 1663, MS-J-580, Los Alamos, NM 87545

53. C. R. Ward, Westinghouse Savannah River Company, Building 773-A, D-1145, Aiken, SC 29808

54. B. Wilding, Westinghouse Idaho Nuclear Company, P.O. Box 4000, Idaho Falls, ID 83403-5104

55. A. P. Williams, Westinghouse Hanford Company, P.O. Box 1970, LO-18, Richland, WA 99352
56. L. W. Yarbrough, Department of Energy, 12800 Middlebrook Road, MS EM-55, Trevion II, Washington, DC 20874
57. Office of Assistant Manager for Energy Research and Development, DOE Oak Ridge Field Office, P.O. Box 2008, Oak Ridge, TN 37831-6269
58-59. Office of Scientific and Technical Information, DOE Oak Ridge Field Office, P.O. Box 62, Oak Ridge, TN 37831

END

DATE
FILMED
6/22/93