A Low-Power Low-Voltage Digital Bus Interface for MCM-Based Microsystems

J. H. Correia, E. Cretu, M. Bartek and R.F. Wolffenbuttel

Delft University of Technology
Dept. of Electrical Engineering
Lab. for Electronic Instrumentation
Mekelweg 4, 2628 CD Delft, The Netherlands
Tel.: +31 15 2781602, Fax: +31 15 2785755
e-mail: higino@ei.et.tudelft.nl

Abstract

This paper describes a digital local bus interface, which is designed for use in a multi-chip-composed microsystem. The chip area using a CMOS 1.6µm n-well technology is 1mm^2. Power consumption at 5V@100kHz is less than 500µW and for 5V@4MHz less than 2mW due to a smart power management of all functional blocks. The bus interface is able to transmit a digital code, bitstream, analog voltage, frequency, duty-cycle and also provides calibration facilities, service request and interrupt request for the smart sensors or microactuators.

Introduction

Present microsystems require a sophisticated local bus to address sensors, microactuators, controllers. Available bus protocols lack the flexibility that is needed to deal with a multi-sensor system on the die level. Suitable buses like: I^{2}C (having too complex protocol) and the basic Integrated Smart Sensor bus (ISS-bus) [1] don't provide calibration, test facilities and also no interrupt mechanisms. In this paper a microinstrumentation system is presented that contains an active silicon platform including all infrastructural functions, such as power/thermal management of the system, test facilities and an on-system sensor bus protocol that allows flexible and low-power data handling using interrupt[2].

The microsystem, as shown schematically in Fig.1, includes all features of a complex measurement system on the smallest possible material carrier; a silicon chip. The microsystem is composed of an universal platform which is to be populated with the required sensors, microactuators and a microcontroller using MCM (Multi Chip Module) techniques.

The bus interface should be versatile enough to ensure efficient communication between all sensors and systems on the platform, however, should be simple enough to be on-chip merged within platform [3]. As an additional feature the bus interface should be able to handle both digital and semi-digital signals, such as pulse width and frequency modulated pulse series. Moreover, self-test should be implemented over the bus by using analog excitation signals and simultaneous semi-digital or digital readout.

The upgraded version of the ISS bus presented here is based on a single controller to coordinate the activity on the bus and includes: a maskable interrupt mechanism, calibration facilities, small size, low-power consumption, which makes it very suitable for implementation on microsystems.

Implementation of an interface bus in a complex structure, like a microsystem requires techniques to reduce the power consumption. As the density and size of chips and systems continues increase, the difficulty of providing adequate cooling either adds significant cost to the system or limits the amount of functionality that can be provided on the microsystem [4]. Assuming that short-circuit current, glitching, and leakage can be kept in bounds with design care, the dominant power source in CMOS is the dynamic power consumption.

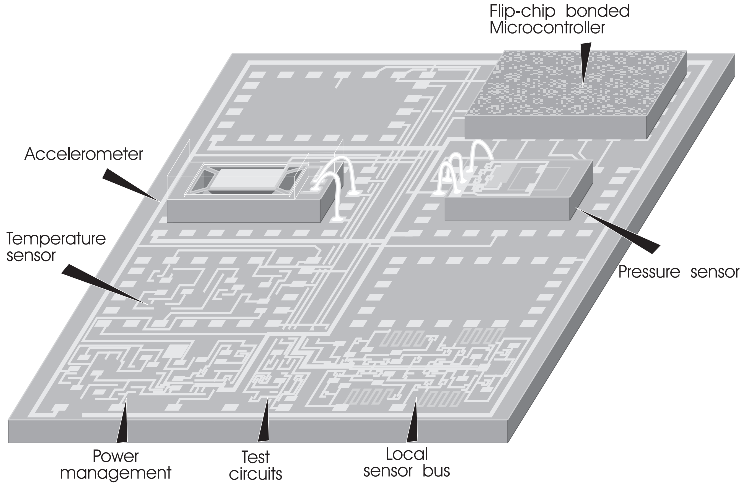

Fig.1: The microsystem.

Functional circuit design

Apart from simplicity, this bus interface has two convenient features, which makes it very suitable for a microsystem. Firstly, analog data can be transferred over the bus. Data generated by a sensor with limited signal processing capability usually comes in this form, so this requirement is necessary, but not present in the usual standard interfaces, which are designated to interconnect only digital subsystems. Secondly, the use of the Manchester encoding scheme for transmission of the data at the logical level adds further flexibility. In such a scheme, the clock is embedded into the data allowing four logical level instead of having two (see Fig.2).

The physical structure consists of two wires, a data line and a clock line, both open drain driven. The data line allows half-duplex communication between the modules connected to the bus. In order to increase flexibility, a second data line was added, to be used only in case of duplex transmission (e. g. in the case of an on-line sensor calibration or testing procedure).

In the case of the embedded systems, particularly for instrumentation systems, sensor modules should also be able to announce the controller when data is available, or more generally, when some particular event has happened. In the realised interface this flexibility is obtained by adding an interrupt request and a service request protocol.

An interrupt request message, if it is not disabled by the configuration used for that particular module, could be sent over the bus in any moment, even if the controller was in the middle of another conversation. A service request instead is allowed only if the bus is in the idle state. Except for the request messages, all the other messages are initiated by the controller.

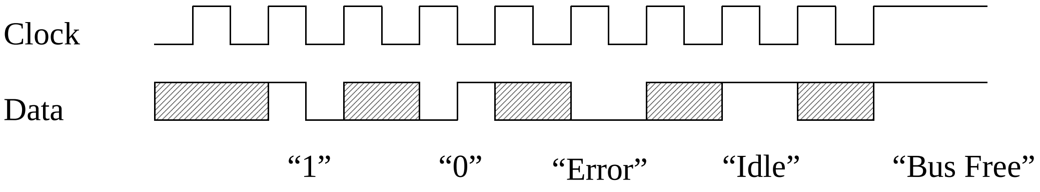

Fig.2: Manchester encoding scheme.

The length of the frame is variable, depending on its significance. The structure of the bus interface is shown schematically in Fig.3. These functions have been realised in a 1.6 µm CMOS process. A photograph of the chip (1.5x0.7 mm^2) is shown in Fig.4.

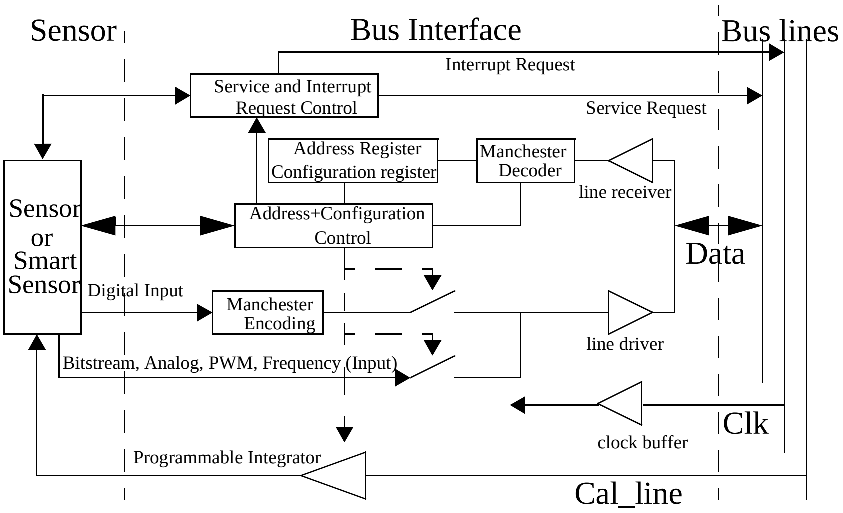

Fig.3: The block diagram of the bus interface.

Smart power management

The interface bus is composed with some synchronous blocks. Especially ones that are pipelined, have a significant portion of the total power dissipated by the clock (responsible for 50% of the total power dissipated) [5]. The power wasted due to the clocking of blocks, which are idle for a significant period of time in normal or standby modes, must be avoided. To manage the power consumption two modes are implemented:

- Selective shut-down of different blocks based on the level of activity required to run a particular application. Different blocks of the chip may be idle for a certain period of time when running different applications (this happens with the service and interrupt request blocks).
- An idle mode for saving of the power dissipation in the standby mode. Waking-up is initiated only at the start of frame to enable verification of the address.

Experimental results

A typical service request sequence transmitted serially over the data bus has been recorded and is shown in Fig.5. When clock and data lines are idle, the sensor demands a service request by lowering the data line, and in reply the master issues a request for the address sensor.

The chip has a power consumption of 500µW (5V@100kHz) and 2mW (5V@4MHz). A network with three bus interfaces was implemented and the request block was successfully tested.

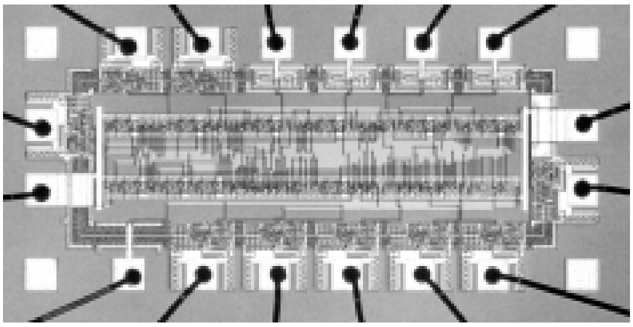

Fig.4: A microphotograph of the bus interface.

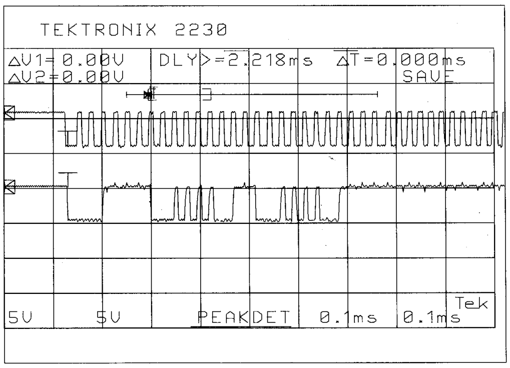

Fig.5:A plot of service request proceeding.

Conclusions

The features of the microsytem bus interface are: simplicity of structure; only two communication wires are used in the minimum configuration; reliable data transfer by using the Manchester encoding with error detection schemes; flexibility of signal type, as synchronous and asynchronous transmission of digital data is possible in combination with semi-digital signals, such as bitstreams, or even analog signals; flexibility of signal handling based on a maskable interrupt mechanism; sensor self-test capability over the bus using separate directional data lines; to be used as a separate die in microsystems and also suitable for on-chip integration with sensors; smart power management in order to reduce power consumption [6].

The features of the bus claimed above have been demonstrated, which makes it suitable for implementation in a practical microsystem. A future version, with the change of 5V supply to a 1.3V supply giving a 90% reduction in power dissipation, will be soon available. However, delay times increase. Pipelining and parallelism techniques will be required in order to reduce the critical paths, so that the supply voltage could be decreased, while maintaining constant throughput[7].

Acknowledgments

This work is supported in part by STW (project DEL 55.3733), TUDelft and JNICT- Portugal (Program Praxis XXI-BD/5181/95).

References

[1] J.H.Huijsing, F. Riedijk, G. v. Horn, Developments in integrated smart sensors, Proc. Transducers ‘93, pp.320-326.

[2] A. Mason, N. Yazdi, K. Najafi, K. D. Wise, A low-power wireless microinstrumentation system for environmental monitoring, Proc. Transducers ‘95, pp. 107-110.

[3] R.F.Wolffenbuttel, Silicon sensors and circuits: on-chip compatibility, Chapman-Hall-1996.

[4] J. Rabaey, Digital integrated circuits: a design perspective, Prentice-Hall, 1996.

[5] A. Chandrakasan, S. Cheng, R. Brodersen, Low-power CMOS digital design, IEEE Journal of Solid-State Circuits, vol. 27, No. 24, April 1992.

[6] U. Ko, P.T. Balsara and W. Lee, A self-timed method to minimize spurious transistions in low power CMOS circuits,IEEE Symposium on Low Power Electronics, Tech. Dig., pp. 62-63, October 1994.

[7] A. Chandrakasan, R. Brodersen, Low-Power Digital CMOS Design, Kluwer, 1995