Heterogeneous embedded system with "microcontroller–CPLD" based shared memory interface for sensor applications

ARTICLE *in* SENSOR REVIEW · NOVEMBER 2005

Impact Factor: 0.74 · DOI: 10.1108/02602280510620132

6 AUTHORS, INCLUDING:

Jivan Parab

Goa University

17 PUBLICATIONS **17** CITATIONS

SEE PROFILE

Gourish M Naik

Goa University

30 PUBLICATIONS **58** CITATIONS

SEE PROFILE

Heterogeneous embedded system with 'Microcontroller-CPLD' based shared memory interface for sensor applications

Jivan Shrikrishna Parab, Rupesh Sadanand Paliekar Porob, Kottanal Roy Francis Joseph, Kunal Vishwanath Naik, Rajanish K. Kamat# and Gourish M. Naik
Department of Electronics, Goa University, GOA – 403 206
#: Department of Electronics, Shivaji University, Kolhapur – 426 004

Brief biographical note:

Dr. Rajanish K. Kamat is Reader in Electronics at Department of Electtronics Shivaji University, Kolhapur.
Dr. Gourish M. Naik is Head of Instrumentation Department, Goa University, Goa. Rest are the research students at Department of Electronics, Goa University, Goa.

STRUCTURED ABSTRACT:

Title: Heterogeneous embedded system with 'Microcontroller-CPLD' based shared memory interface for sensor applications

Authors: J. S. Parab, R. S. Paliekar Porob, K. Roy F. Joseph, K. V. Naik, R. K. Kamat# and G. M. Naik
Department of Electronics, Goa University, GOA – 403 206
#: Department of Electronics, Shivaji University, Kolhapur – 426 004

Purpose: Aims to design a heterogeneous embedded system with CPLD and microcontroller as coprocessors sharing a memory module.

Design/methodology/approach:
The system receives external analog input signal, which is applied to the PIC 16F73 Microcontroller. Upon converting the data in to digital format using the on-chip ADC, the PIC stores the digitized version in the SRAM (HCM6264) chip. SRAM HCM 6264 has been used as a shared memory model, of which both the PIC and CPLD can access all the locations. Once the PIC passes controls to the CPLD, the further processing is carried out by the CPLD without any intervention of the PIC. This is a true example of co-processing of the architecturally diversified computing modules from completely different vendors with totally different programming suits.

Findings:

The board has been tested with IC temperature sensors and also found to be useful for sensor array applications involving three types of processing viz. analog (through instrumentation amplifier), real time digital (through microcontroller) and customized reconfigurable digital (with the CPLD).

Practical implications:

The system has several potential applications in avionics, military and robotic embedded systems, which have inherent real-time constraints that need to be supported by the underlying hardware and driver programs.

Originality/value:

Discusses the rare and unique combination of diversified processing core to build an embedded system.

Abstract:

The paper describes design of a heterogeneous embedded system with CPLD and microcontroller as coprocessors sharing a memory module. The system comprises of a microcontroller with built in ADC catering to the requirements of sensor interfacing. Device drivers have been developed to sample the analog output of the sensors and are subsequently stored in the shared RAM for further processing by the CPLD module. The CPLD has been configured as a customized digital signal processor with the required utilities for data processing. The processed data by the CPLD is further converted back to analog domain in order to actuate the final control elements. The developed interface is a standalone embedded system, which is programmed through the centronix and RS-232 interface using PC. The board has been tested with IC temperature sensors and also found to be useful for sensor array applications. The system exhibits three types of processing viz. analog (through instrumentation amplifier), real time digital (through microcontroller) and customized reconfigurable digital (with the CPLD).

1. Introduction:

Many real time sensor system applications are dominated with the requirement of digital processing of signals, which faces typical challenges of retaining the overall real time nature of the system. Digital signal processing in the above-mentioned systems has traditionally been done using enhanced microprocessors or specialized DSP processors. While the high volume of these generic product provides a low cost solution, the performance falls seriously short for many applications requiring a real time response. The application domains, which typically face the above-mentioned bottlenecks, are military embedded systems, automotive control systems and biomedical systems. All the above mentioned systems are in need of diversified performance requirements such as flexibility, customizability, reliability, easy upgradation etc. Moreover there is a tradeoff between the computational intensive processing Vs real time performance. Designers face considerable challenges in interfacing the multiple sensors or sensor arrays (which are the common attributes of these systems) with the limited I/O port lines of today's microcontrollers. The other important design issues are low power operation, at an affordable price with rapid prototyping to reduce the time to market. In order to satisfy the design requirements the sensor fraternity is now relying on embedded systems with heterogeneous multicomputing. This is an advanced parallel processing technique that significantly reduces the cost of embedded computing by allowing more than one kind of processor type that is just suitable for the intended task.

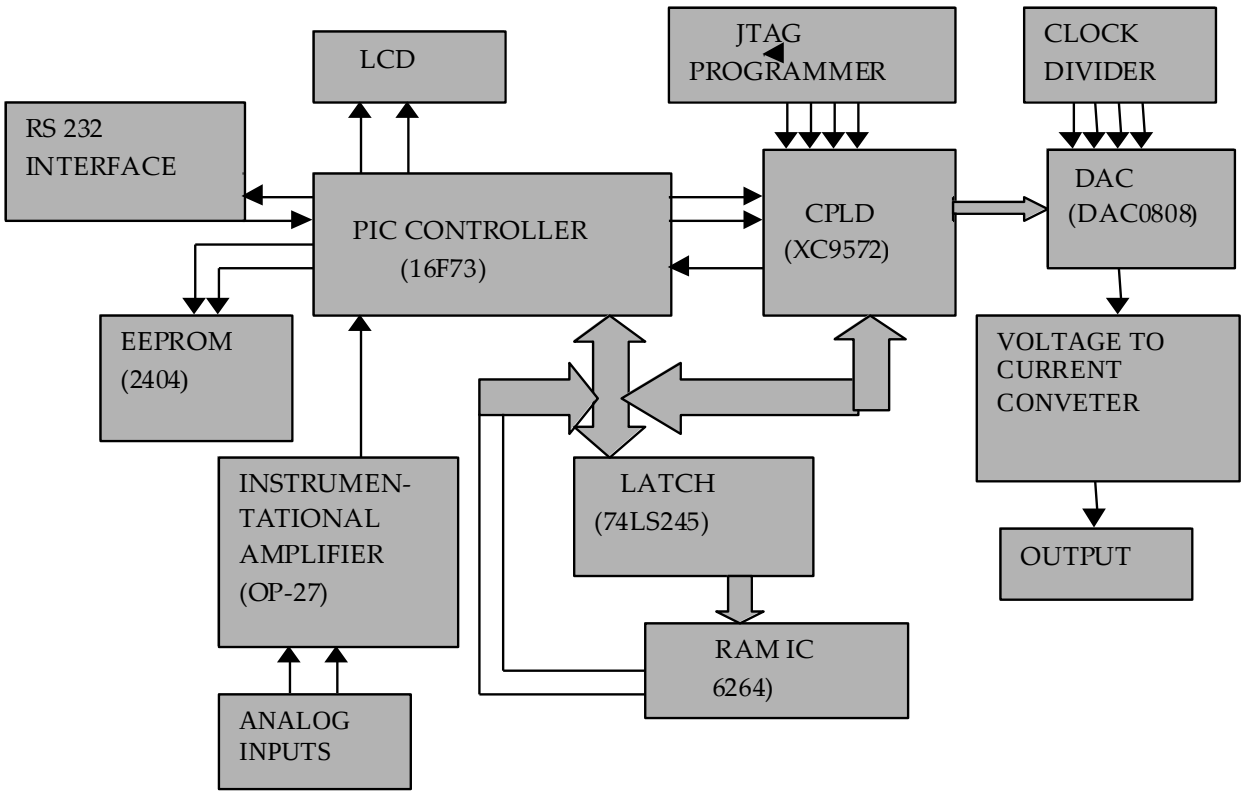
The present communication reports design of a heterogeneous embedded system with low cost, high speed and reliable DSP architecture with real time I/O control. The system has two major blocks i.e. a PIC microcontroller and Xilinx CPLD. Both the blocks work as tightly coupled coprocessors with a shard memory model. The task partitioning goes as follows: PIC is responsible for data acquisition with digitization and subsequent storage in the memory module accessible by both the co-processors. The CPLD takes care of processing of the stored data and outputs the same to the DAC module.

2. Proposed Scheme:

The proposed signal conditioning board is a heterogeneous embedded system having architecturally diversified computing resources like Microcontroller and the CPLD. The former has been widely used in the industrial world as a real time controller while the latter has been seen to restrict

its applications to customized reconfigurable digital modules. PIC 16F73 has been chosen as the microcontroller owing to its onboard ADC, which is readily used for digitization of the sensor outputs. The CPLD sub module of the system is implemented using XILINX CPLD XC 9572 owing to its freely available software (exclusively for students) for programming. The block schematic of the system is shown in figure 1.

Figure 1: System Block Diagram



The system receives external analog input signal, which is applied to the PIC 16F73 Microcontroller. It features an in-built Analog to Digital Converter, which has been programmed to convert the incoming analog signals from sensors into digital domain for storage and further processing. Upon converting the data in to digital format the PIC stores the digitized version in the SRAM (HCM6264) chip. SRAM HCM 6264 has been used as a shared memory model, of which both the PIC and CPLD can access all the locations. Once the PIC passes controls to the CPLD, the further processing is carried out by the CPLD without any intervention of the PIC. This is a true example of co-processing of the architecturally diversified computing modules from completely different vendors with totally different programming suits.

The co-processing is initiated by the PIC 16F73 microcontroller, which is responsible for data acquisition, digitization and storage of the same into the memory. The control is then passed on to the XILINX CPLD XC9572, which in turn becomes the master processor. It then independently generates all the control and timing signals for acquiring the data from memory and processes the same with the built in algorithms and further passes it on to the DAC for analog conversion. DAC 0808 has been interfaced to the CPLD for the conversion purpose. The analog sensor output is processed by using an instrumentation amplifier based on OP-27 to scale it in the range of the built in ADC on the PIC 16F73 microcontroller. With the analytical marriage of the architecturally diversified processors the maximum benefits of the analog and digital processing is achieved by dividing the labor to the appropriate blocks which would give better justice to the real time system performance in addition to the computational intensive operation. The analog signals processing is achieved by using OP-27 based instrumentation amplifier. The interfacing of the peripherals like LCD and digitization is done by the PIC 16F73 microcontroller as shown in the circuit diagram given as figure 2.

As shown in the block diagram, CPLD is programmed by using the JTAG interface. This JTAG four wire interface has been derived by modifying the Centronix interface of the PC. The CPLD shown in the diagram is programmed through the JTAG interface by using the freely downloadable webpack provided by Xilinx. Using a separate programmer, the diagram of which is given separately, programs the PIC microcontroller. The EPROM 24C04 is I2C compatible, which functions as a buffer for transmission and remote telemetry to other systems. The instrumentation amplifier at the input of the PIC microcontroller scales the sensor output in the range of the built in ADC. The

system has a provision to upload the data to the PC if required through the USART by using the RS-232 interface. The CPLD and microcontroller acts as coprocessors to read the sensor data from the shared SRAM based on IC 6264. The PIC microcontroller also drives a serial LCD to display the status of the system. The analog reconstruction of the processed data is achieved by using DAC 0808 followed by a current to voltage converter. The clock divider provided on the board facilitates the system to operate at different speeds depending on the requirement.

3. System Specifications:

PIC Specifications:

	Typical	Board values
Operating speed	0-20 MHz	3-12MHz
Data Memory (RAM)	368 x 8	---------
Synchronous Serial Port (SSP)	AVAILABLE	NOT USED
SPI	AVAILABLE	NOT USED
I2C	AVAILABLE	USED
USART	AVAILABLE	USED
Power Consumption	<2 mA typical @ 5V, 4 MHz 20 mA typical @ 3V, 32 kHz < 1 mA typical standby current	
Operating voltage range	2.0V to 5.5V	5V
Built-in ADC Channels	5	2

ADC Specifications:

A/D conversion clock	MIN: 1.6 μs	
Resolution PIC16F73	8 bit	
Total absolute error	< ±1 LSb	
Integral linearity error	< ±1 LSb	
Differential linearity error	< ±1 LSb	
Full scale error	< ±1 LSb	
Offset error	< ±1 LSb	
Reference voltage	2.5 --5.5 V	
Analog input voltage	MIN:VSS - 0.3	MAX:VREF + 0.3 V
Recommended impedance of Analog voltage source	10.0 kΩ	
Ref input current	±5 μA	During Acquisition.
	500 μA	During A/D Conversion
Acquisition time	Min: 5μS	

CPLD Specifications

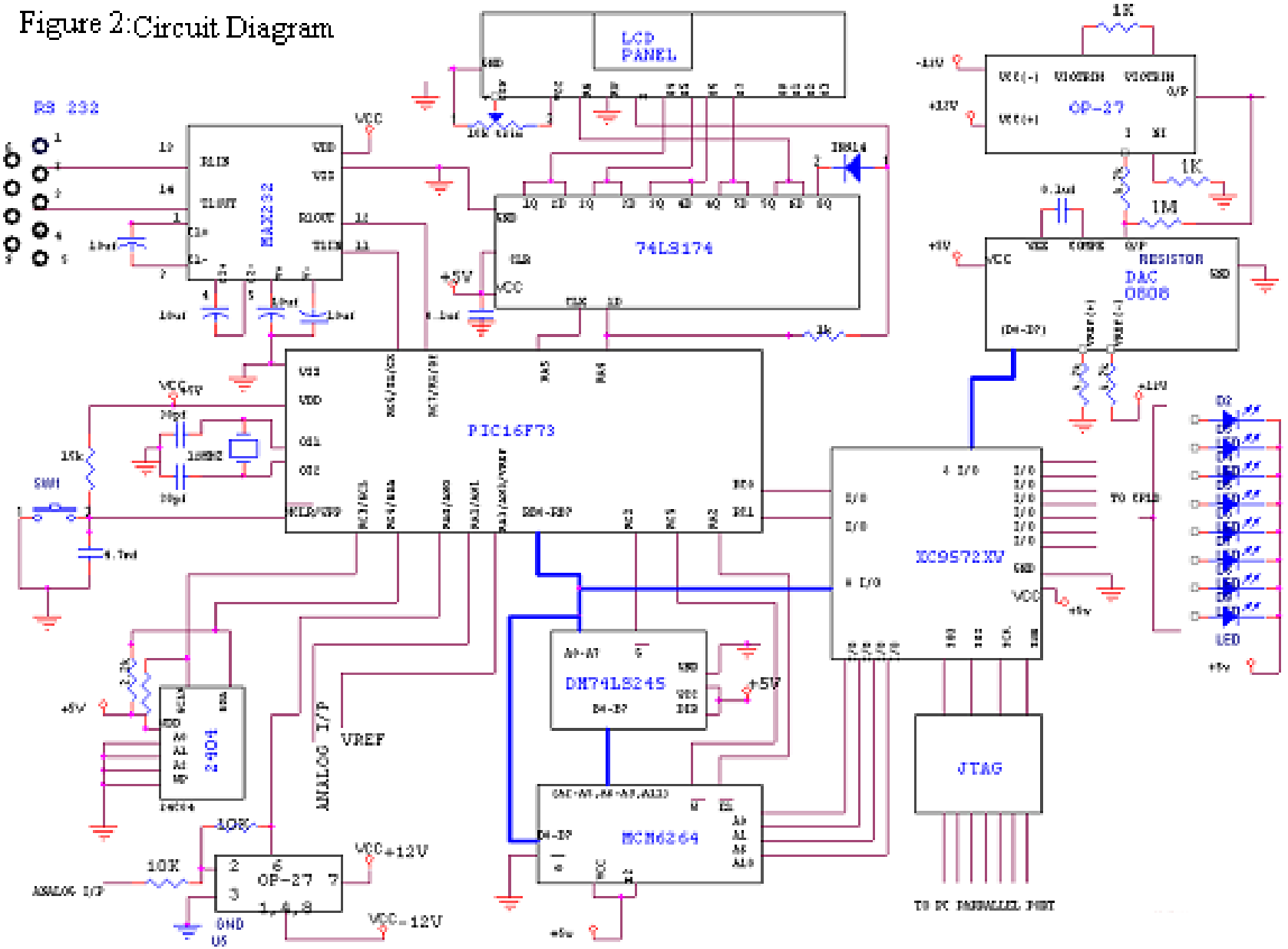
7.5 ns pin-to-pin logic delays on all pins
72 macrocells with 1,600 usable gates
Extensive IEEE Std 1149.1 boundary-scan (JTAG) support

RAM Specifications
Fully Static — No Clock or Timing Strobes Necessary
Fast Access Times: 12, 15, 20, 25, and 35 ns
Low Power Operation: 110 – 150 mA Maximum AC
Fully TTL Compatible — Three State Output

DAC Specifications

Full scale output current Settling time:	150 ns
MAX Power Dissipation:	33 mW with ±5V supplies.
Relative accuracy:	± 0.19% error maximum
Full scale current match:	± 1 LSB typical
Fast settling time:	150 ns typical
Low power consumption:	33 mW @ ±5V

Figure 2:Circuit Diagram



4. Interfacing and Co-processing details:

The Central processor of the interface board is PIC16F73. In the prototyping phase, we have used only two analog inputs viz. RA0/AN0 (PIN 2) and RA1/AN1 (PIN 3). A provision for external reference is provided at RA3/Vref (PIN 5). It needs to be left unconnected when internal reference is used. The PIC runs at a frequency of 12 MHz. The built in ADC has a conversion speed of not less than 1.6μs; however it basically depends on the internal clock frequency. The actual sampling rate is however, very less of the order of 1 μs since the PIC has to store each sample received by it instantly in the SRAM. Because of the restriction placed by the number of pin on the PIC we decided to provide the column address from the CPLD. In the sensor matrix based applications, where there is a possibility of running short of CPLD gates a provision of manual column selection is kept. The 8K SRAM 6264, shown in the diagram may apparently seem to be very large. However, it was used so as to store a larger number of samples acquired by the PIC. The SARM is configured in always-enabled mode for simplifying interfacing complexity. The SRAM also has multiplexed address and data lines, which are shared by both PIC as well as the CPLD. Care is taken in the program for PIC and the CPLD, to allow wired ANDing of the common lines. This also means that the data can go from the PIC to CPLD, or from PIC to SRAM and vice versa. A latch is used to de-multiplex the address and data signals. The PIC and CPLD share even this line. Since the column is fixed, only one column can be used i.e. only 256 locations could be used at a time. The PIC takes in data and after conversion stores in the SRAM. There are two control lines running from the PIC to CPLD, which specifies the action of the CPLD. (Refer the state machine of CPLD for the details). Moreover, there is a reset line to the CPLD for resetting purpose. In addition to these lines the busy line serves as an input to the PIC and forces the processor to monitors the status of the CPLD. When the PIC is occupied with processes not related to the CPLD, it forces the CPLD in the idle mode (which is also the default mode of the CPLD). In this mode the CPLD pins automatically become tri-stated to avoid bus contentions. The basic operation of CPLD in the system is for processing the signal, and hence the PIC passes the control to the CPLD only when it has stored some data in the SRAM that needs to be processed in the CPLD.

The state diagrams shown in figures 3 and 4 reflect all these implementations in the CPLD. On receiving the control from PIC the CPLD pulls its busy line low indicating that it is under the control of the main processor i.e. PIC. In this state the CPLD follow the instruction given by the master i.e. PIC. After providing the necessary information to the CPLD the PIC puts the CPLD in

a self-operating mode. The 'necessary information' provided is regarding the column in which data has to be stored and / or the last location accessed by the PIC to store data. Looking at the limited gates on the CPLD, the above mentioned aspects are implemented in a minimum scale. In the self-operating state the CPLD places logic '1' on the busy line indicating that it is now out of the control of the PIC. The only signal to which the CPLD will respond in this mode is the reset. Reset in any state will put the CPLD in the idle state. Once the CPLD is configured as a Digital Signal Processor, it automatically generates the address starting from 0 until the last location. Concurrently, the other sections generate the ALE to de-multiplex the address and data signals. The read logic acquires the data and places it in the processing block. As soon as the data is processed the same is provided at the output. The processing block works in parallel with the other blocks in the CPLD.

The output provided by CPLD drives the 8-bit DAC 0808 in order to actuate the final control elements like solid state relay. An instrumentation amplifier based on OP-27 is provided at the input to scale the sensor signal. The additional on board resources are a 2-wire LCD (for displaying the status), I2C bus (for communication to other boards), RS 232 interface (for uploading the readings to PC), JTAG interface and clock divider. The 2-wire LCD requires only 2 line of the controller namely data and clock .In our board PIN 6 of the microcontroller is assigned as data and PIN 7 as clock. The serial LCD module consists of 6-bit shift register, which converts serial data from controller into parallel. This parallel data drives the LCD. The PIC processor has built in support for I2C protocol. The actual pins supporting this protocol are SDA (pin RC4) and SCL (RC3). A serial EEPROM is interfaced through these pins on the PIC. PIC features an in-built UART which is used to interface the system to a PC through a MAX 232. The MAX 232 makes the signal provided by PIC compatible with the RS232 standard. The programming of the CPLD is done through a JTAG interface the details of which is covered in standard references [1,2].

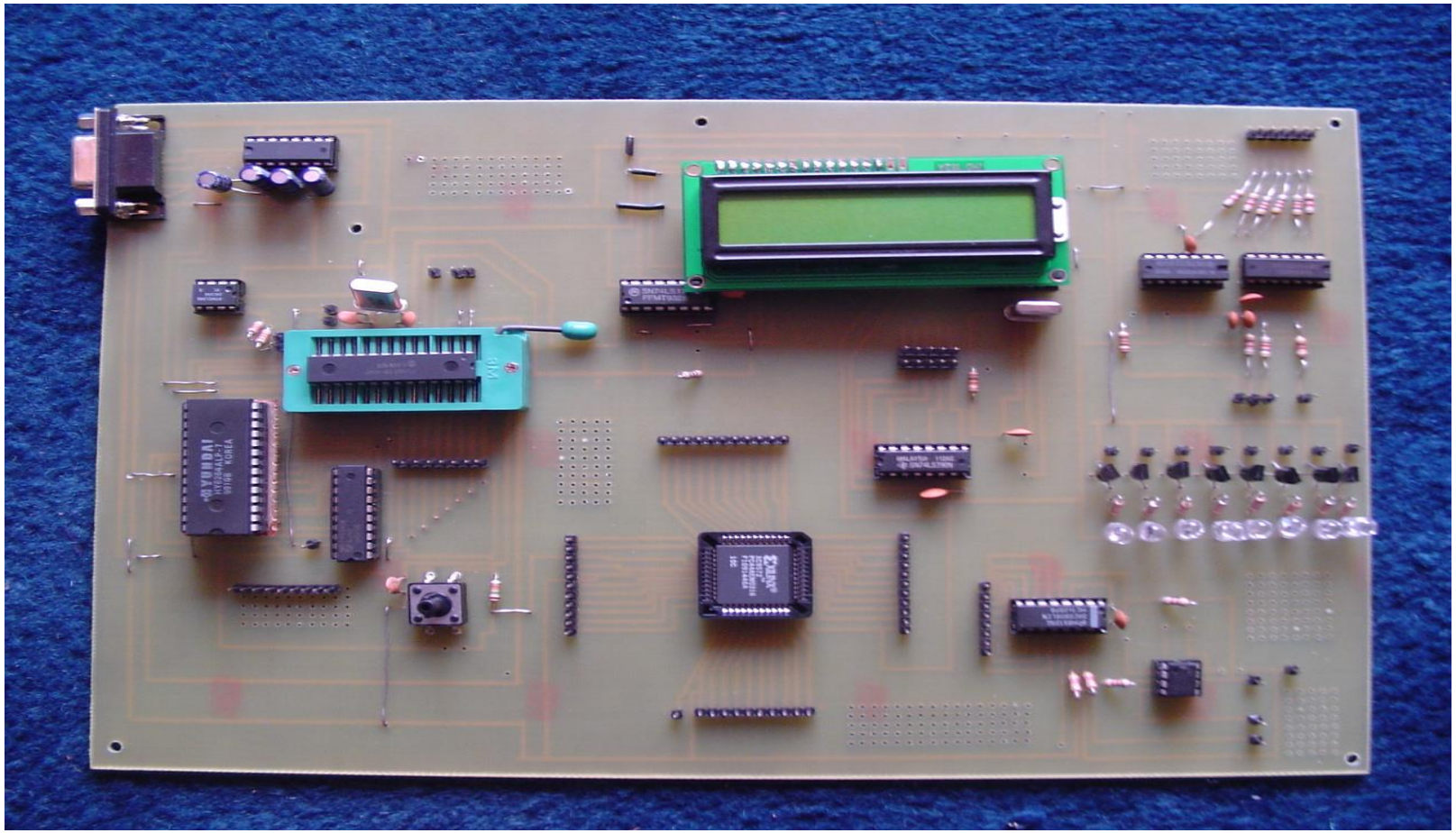
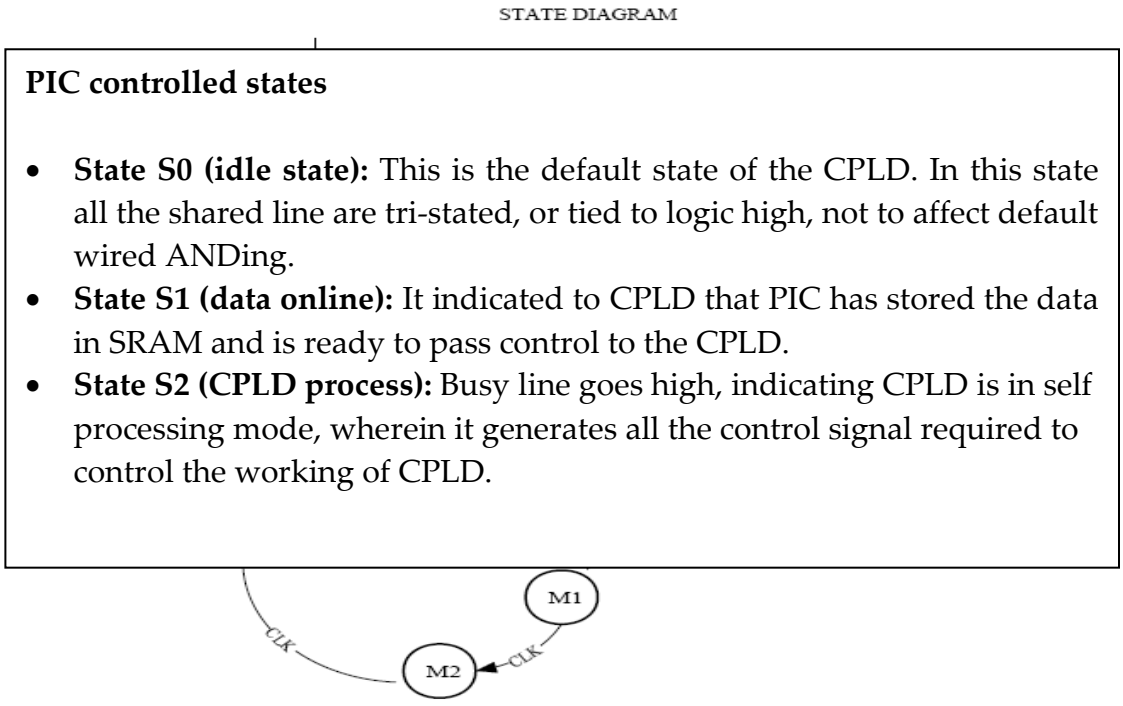


Figure 3: A snapshot of the system

In order to facilitate the choice of different operating speeds a clock divider is implemented on the board. This can provide 16 MHz, 8 MHz and so on. The clock divider is implemented using a decade ripple counter.

5. State space diagram of the system:



Self processing states of CPLD:

- **M0**: put address on the address data line
- **M1**: latch ALE
- **M2 (DSP)**: reads data on address data line and places in the DSP processing block.
- **M3**: increment program counter and compare for last location. If last location reached put busy high and go to M0; else go to M0.

The programming logic is expressed in the flowcharts shown in figure 4.

6. Utilization of on-board resources:

PIC16F73	AVAILABLE	USED
Operating Frequency	DC - 20 MHz	12 MHz
Interrupts	11	1
I/O Ports Ports	A, B, C	ALL
Serial Communications	USART	USED
Parallel Communications		
8-bit Analog-to-Digital	5	3

.

XILINX XC9572	AVAILABLE	USED
Operating Frequency	125MHz	16
Macro cells	72	ALL
Usable gates	1600	ALL
User I/O	33	ALL

7. Conclusion:

The paper addresses the problem of designing heterogeneous multiprocessor embedded systems for sensor applications. It focuses on the basic steps of a proposed design flow with particular emphasis on the partitioning activity and the related software driver development.

The signal conditioner developed features real time control maintaining the requirements of high speed DSP processing by getting rid of the redundant blocks in the digital signal processor. Maximum throughout is achieved with microcontroller and CPLD in a co-processing mode for digital processing, while instrumentation amplifier for analog signal conditioning. Sensor fraternity and

industrial community has expressed the necessity of such kind of signal conditioning boards since a long time

The system has several potential applications in avionics, military and robotic embedded systems, which have inherent real-time constraints that need to be supported by the underlying hardware and driver programs.

References:

1. *Xilinx*

   URL:http://www.xilinx.com/prs_rls/software/0250webpackiyeah42i.html

2. *WebPACK 6.1i*

   URL: www.xilinx.com/ise/webpack6

3. *I²C*

   URL: http://www.pond.ie/techinfo/uf876b/

4. *Two-WIRE LCD INTERFACE*

   URL: http://www.rentron.com/myke1.htm

5. *RS232 Interfacing*

   URL:http://www.scopus.com/scopus/results/results.url?sort=plff&src=s&sid=4puuHZqAox5FoCSJ62qpK0q%3A250&sot=aut&sdt=a&sl=31&s=AUTHOR-NAME: Costa+De+Paula%2C+L.%29

   AUTHORS: Costa De Paula, L., Fernandes, J.M.

   SOURCE TITLE:ISA TECH/EXPO Technology Update Conference Proceedings 413 I , 695-700 DATED: 2001

6. *RS232 Pin out Details*

   URL: www. LX200 RS232 Port Connections.htm

7. *PIC AND RS232 INTERFACING*

   URL:www.electronicengineering.ch/microchip/projects/rs232/rs_test/rs_test.html

8. *I²C MASTER SLAVE*

URL:http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1999&ty=&dty=&section=&NextRow=&ssUserText=i2c+bus+&DesignDocSelect=

9. PIC COMPILER

URL: http://www.ccsinfo.com/download.shtml

10. *Web Resources on heterogeneous embedded systems:*

URL: http://www.cotsjournalonline.com/home/article.php?id=100233

URL: http://www.rtcmagazine.com/home/article.php?id=100055

URL: http://www.isis.vanderbilt.edu/projects/acs/Papers/HPEC99_Abstract.pdf**.**

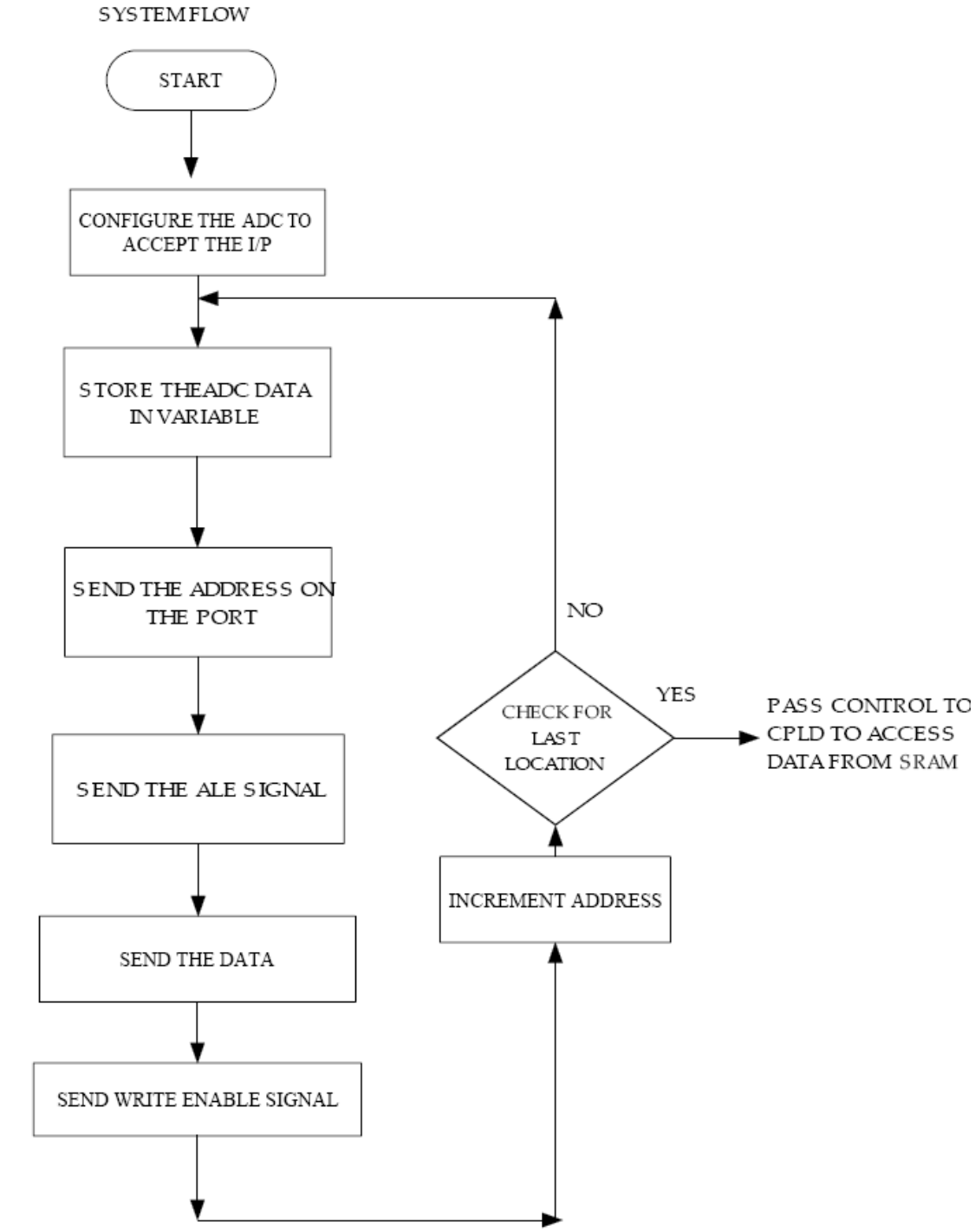
URL: http://www.mc.com/literature/literature_files/edn-7-18-94- at.pdf**.**

11. HETEROGENEOUS MULTICOMPUTING FORCOST-EFFECTIVE EMBEDDED SYSTEM

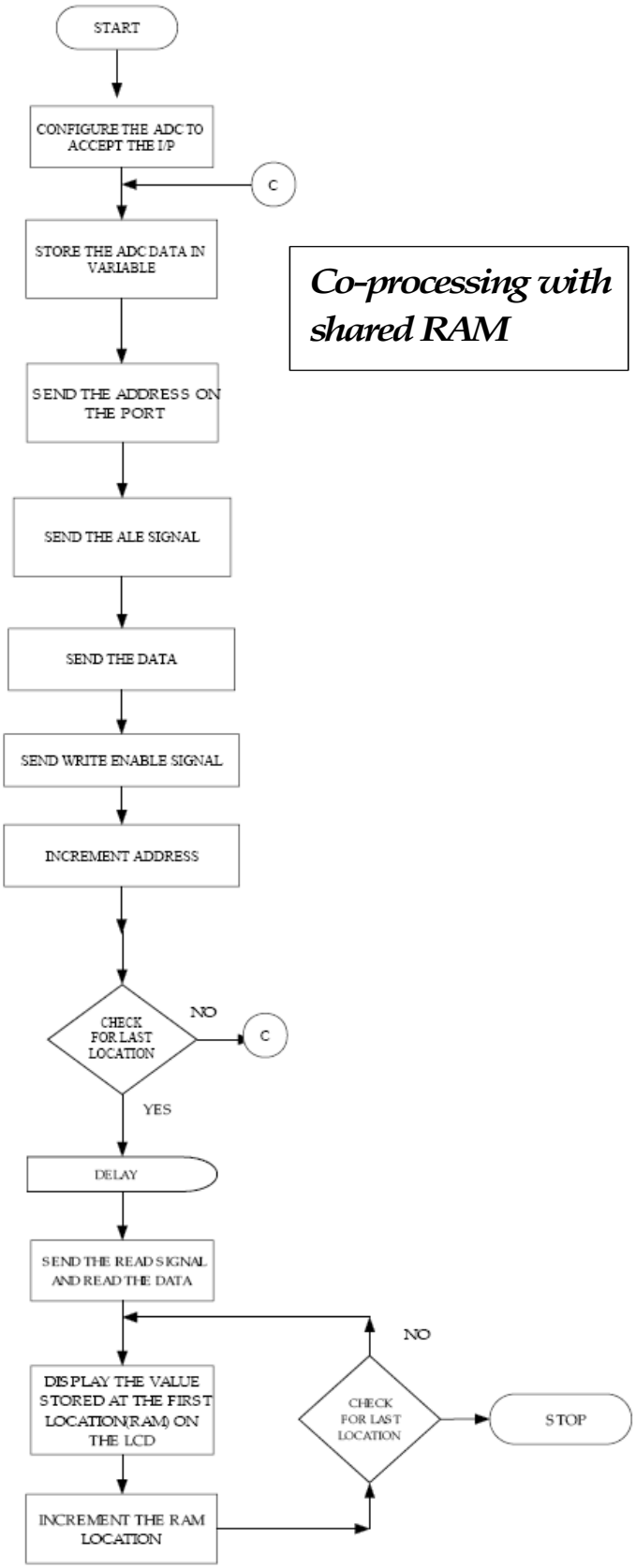
URL: http://portal.acm.org/citation.cfm?id=502217.502223

URL: http://www-old.ece.rice.edu/~brogioli/research_interests_page.htm

System Flowchart:



FLOWCHART TO WRITE AND READ FROM SRAM



Co-processing with shared RAM