# Design of a Programmable Bus for Microprocessor-Based Systems

**Dr. Kasim M. Al-Aubidy, Muthana A.K. Attyah**

Faculty of Engineering, Philadelphia University,
Sweileh P. O. Box 1101, Amman JORDAN,
Tel: 962-2-6734444, Fax: 962-2-6734440
*e-mail: kasimal@joinnet.com.jo*

**Abstract:. A programmable bus system has been developed to solve several problems in the design and implementation of microprocessor-based systems. This bus system consists of two parts; a standard I/O bus add-on card, and the bus system motherboard. The proposed bus provides four slots, each slot has independent interrupts, decoding signals, and a programmable oscillator with a wide frequency range. Such a bus system is suitable for many applications with normal requirements.**

**Keywords:** ***PC buses, bus systems, microprocessor interfacing, Shared memory, microprocessor-based systems.***

## 1. INTRODUCTION

The outside environment parameters to be measured and processed by a computer are in different physical types, and each type has its own speed and amplitude of change. This introduces a wide range of different configurations of data acquisition and distribution systems. Different computer interfaces are therefore required to achieve the correct sampling rate, data word size, number interrupts, number of I/O ports, power supply, and clock speed [1-3]. Consequently, it would be difficult to design a single interface on a personal computer (PC) card that satisfies the different needs of microprocessor-based system developers. The solution is to have a general bus on which different data acquisition/ distribution cards can be connected.

A computer bus is simply a set of lines over which information flow between two or more devices. Devices on the bus can send to, and receive information from other devices. Most modern PCs have at least four buses, these are: the processor and memory bus, the cache bus, the local bus, and the standard I/O bus. The local I/O bus is a high-speed bus used for connecting performance critical peripherals to the microprocessor system. The two most common local I/O buses are the VESA local bus (VLB) and the peripheral component interconnect bus (PCI). The standard I/O bus is used for slower peripherals, and also for compatibility with older devices. In most modern PCs, the standard I/O bus is the industry standard architecture (ISA)[4]. Newer PCs actually use an additional port for graphics communications only. This port is called the accelerated graphics port (AGP).

A number of buses have been designed and applied to satisfy different needs and applications, each bus has its own data width, speed, bandwidth, protocol, signal levels and number of pins. Table I lists the main characteristics of the common I/O buses on PCs today[4].

Table I
Characteristics of common I/O buses

| Bus Type | Width (bits) | Speed (MHz) | Bandwidth (MBytes/sec) |
|---|---|---|---|
| ISA | 8 | 8.3 | 7.9 |
| ISA | 16 | 8.3 | 15.9 |
| EISA | 32 | 8.3 | 31.8 |
| VLB,PCI | 32 | 33 | 127.2 |
| PCI2.1 | 64 | 66 | 508.6 |
| AGP | 32 | 66 | 254.3 |
| AGP*4 | 32 | 66*4 | 254.3*4 |

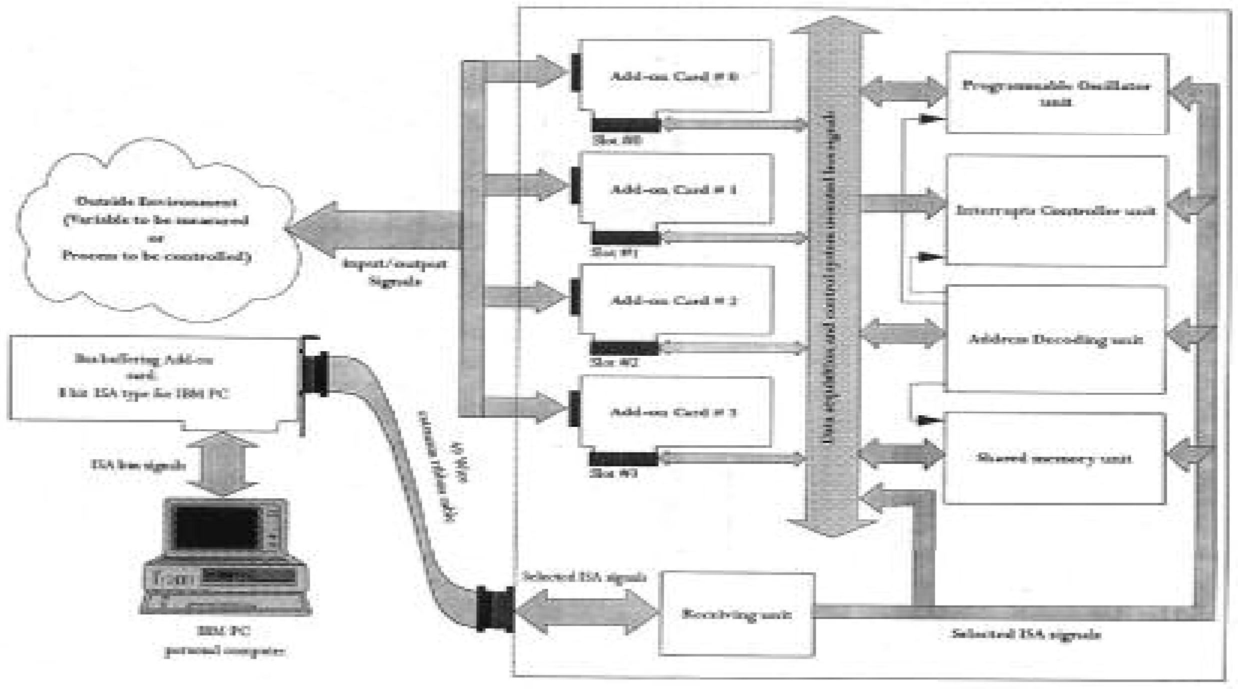


**Fig.1:** General block diagram of the bus system.

Many applications require knowledge of the bus systems located within the PC. These systems often require custom interfaces attached to one of the buses on the main board[5,6]. From an I/O stand-point, much more I/O bandwidth is needed to handle the large amount of information that must be moved between the processor, memory, video and storage disks. The most common bus in the PC world used for normal I/O interfacing is the ISA. Also, there are still many devices for which the ISA's speed is more than efficient. The ISA provides reasonable throughput for low bandwidth devices. Many expansion cards, even modern ones, are still only 8-bit cards [4]. Therefore, the ISA bus has been used in the present work as a bridge between the proposed bus system and the host PC. This system will provide the hardware/software developers an easy start-up point that will save both time and effort in developing the required microprocessor-based system.

## 2. HARDWARE DESIGN

The bus system hardware consists of two main parts, the first part is an ISA add-on card for a PC that extends ISA signals. The second part is the bus motherboard.

### 2.1 The PC Bus Extended Card:

The ISA bus extended card is a PC add-on card used to redrive the selected ISA signals and to extend them through a ribbon cable. The bi-directional and unidirectional extended signals are buffered using bus transceivers. The total extended signals get out from the add-on card via a 49-pin connector, as illustrated in Fig.1.

### 2.2 The Bus Motherboard:

In this part the signals are buffered and distributed by the receiving unit to the address-decoding unit, interrupt controller unit, programmable oscillator unit, and the shared memory unit. Signals collected from these units together with the ISA signals of the receiving unit are combined to form the new bus that extends across the four bus slots, as illustrated in Fig.1.

- *Receiving Unit:*

The function of this unit is to buffer the signals received through the ribbon cable. The used buffers have TTL hysteresis that removes noise added to the signals during the process of data exchange between the PC and the bus motherboard. Also, the buffer will give the signals enough strength to cope with the loads added from the other units and the bus add-on cards.

- *Address Decoding Unit:*

In any add-on card, whatever its function, there must be an I/O address by which the card can communicate with the PC. A decoding unit is required to detect the selected address lines and generates the selected signal to activate the I/O device at the proper time. The decoding unit is used to provide ready decoded I/O addresses that can be used in addition to the I/O control lines or memory control lines to form I/O ports or memory locations. It provides four signals to interrupt unit, four signals to the programmable oscillator, three signals to the shared memory unit, and four signals for each bus slot, as shown in Fig.2.

The address-decoding unit consists of two 16-bit decoders. The enable line of the decoding unit is controlled by a 4-bit comparator, which compares the status of DIP switches with the address lines (A5-A8). This allows selecting the base address of the 32 select lines by setting the DIP switches.

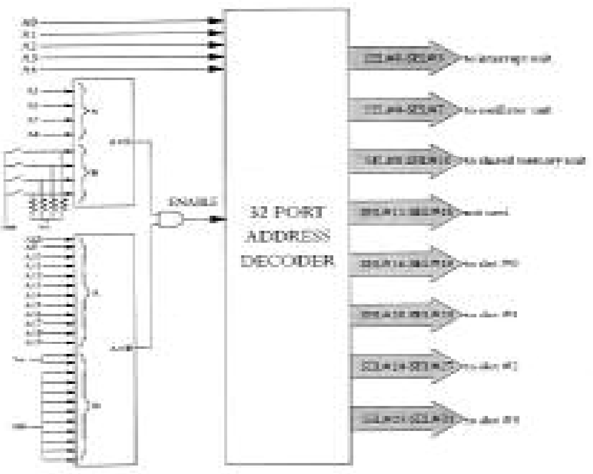


**Fig. 2:** I/O address decoder block diagram.

- *Interrupt Controller:*

The programmable interrupt controller (8259A) chip is cascaded to the main interrupt controller of the PC by taking one of the ISA interrupt lines. The interrupts IRQ3, IRQ4, and IRQ7 are chosen to be attached to the demultiplexer, as shown in Fig.3. The interrupt controller manages eight levels or requests and can be expanded up to 64 levels. It

is programmed by the system's software as an I/O peripheral.

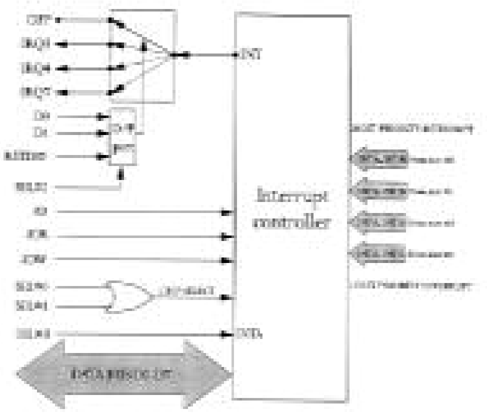
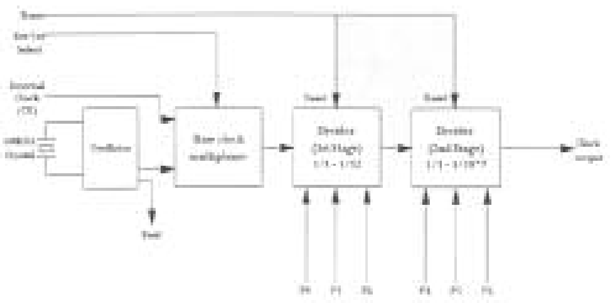


**Fig. 3:** Interrupt unit block diagram.

- *Programmable Oscillator:*

The bus system oscillator can be programmed to provide up to 57 different frequencies. Four programmable oscillator chips are used, one for each slot. This means that each slot has an independent wide range of frequencies from which a frequency can be selected by software. The oscillator chip (PXO-600) generates 57 different frequencies from a quartz crystal (600kHz), which represents the base clock. Also, it is possible to supply an external clock with additional 57 frequencies, as shown in Fig.4. The first stage frequency divider provides dividing ratios of 1/1 to 1/12 as selected by the programming pins p1, p2, and p3. The second stage provides dividing ratios up to 1/10^7 as selected by the programming


Oscillator
Base clock multiplexer
Divider (1st Stage) 1/1 - 1/12
Divider (2nd Stage) 1/1 - 1/10^7

pins p4, p5, and p6.

**Fig.4:** Simplified block diagram of the PX0-600 oscillator chip.

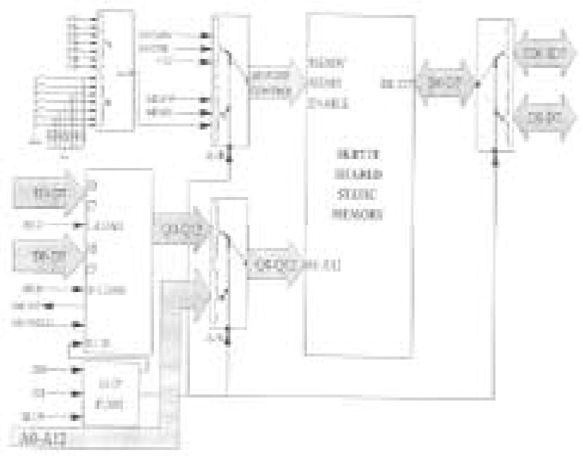


**Fig. 5:** Shared memory block diagram.

- *Shared Memory Unit:*

A secondary memory buffer is required to store the acquired data without disturbing the main processor. When this memory is filled, the data can then be transferred to the main memory in one burst under the microprocessor control.

## 3. SHARED MEMORY DESIGN

The shared memory consists of an 8k byte static RAM, 13-bit up/down counter, a two-channel multiplexer, address decoder, an output port and bus transceivers, as illustrated in Fig.5. The counter has the capability to count up or down as instructed by a control line (U/-D) generated through an output port. The counter output lines (Q0-Q12) are acting as a secondary address bus that controls the memory when it is handled by the add-on cards. The address bus of the shared memory is connected to the multiplexer output. The select line of the multiplexer is generated by the microprocessor through an output port. It either selects the counter output (Q0-Q12) or the microprocessor address bus (A0-A12). The address decoder, which is an n-bit comparator, compares the microprocessor address lines (A13-An) with the status of the DIP switches.

The memory data bus is connected to the outputs of two bus transceivers, only one of them can be enabled at a time. The first transceiver input is connected to the slots data bus (SD0-SD7) that can be used to access the shared memory from the add-on cards. The second transceiver input is connected to the microprocessor data bus (D0-D7).

## 4. SOFTWARE DESIGN

The software design part of the bus system is a collection of initialization program segments, hits, and procedures that can be used as a base for writing specific application program without wasting time or efforts.
The first step in writing the application programs is to know the I/O map of a particular PC, the second step is to select a free I/O address range which can be used by the bus system. Table II gives details of each port address and its usage in the bus system. Then, it is necessary to define the base I/O address as global variable accessible by all the application programs that might use the bus system.
The additional 8259A controller offers two interrupts for each bus slots, and for each interrupt there is a service routine. All of the eight service routines are called from the service routine of the cascading interrupt, which is IRQ3, IRQ4, or IRQ7 as selected from port BasePort+2.
The programmable oscillator in each slot is programmed through an output port. The addresses of these ports are given in Table II.
The shared memory is programmed to be used by the four slots. In this case only one slot can access the shared memory at a time. A software flag is used to have this indication, as illustrated in Fig.6.

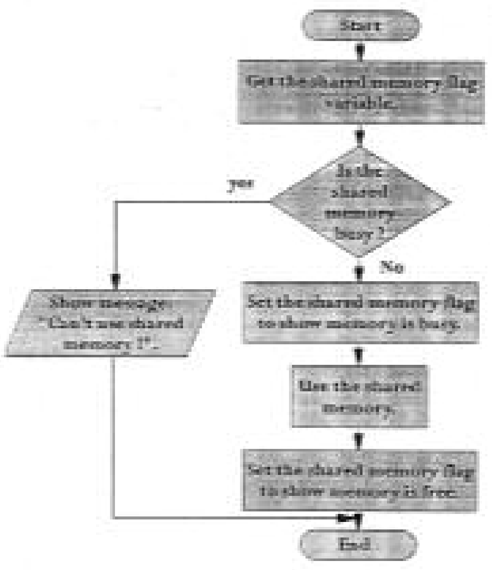

Start
Get the shared memory flag variable.
Is the shared memory busy ?
yes
No
Show message: "Can't use shared memory !!".
Set the shared memory flag to show memory is busy.
Use the shared memory.
Set the shared memory flag to show memory is free.
End

**Fig. 6**: Shared memory flag setting procedure.

The initialization software part is defined to be written as a program that runs from the DOS prompt, or as a device driver that can be added to the CONFIG.SYS file.

Table II
I/O Ports usage in the bus system.

| UNIT | PORT FUNCTION | ADDRESS |
|---|---|---|
| Interrupt Controller | Port #0 | BasePort+0 |
| | Port #1 | PasePort+1 |
| | Select ISA Interrupt | PasePort+2 |
| | INT ACK Port | PasePort+3 |
| Programm Oscillator | Oscillator of Slot #0 | PasePort+4 |
| | Oscillator of Slot #1 | PasePort+5 |
| | Oscillator of Slot #2 | PasePort+6 |
| | Oscillator of Slot #3 | PasePort+7 |
| Shared Memory | Setting L Address Count | PasePort+8 |
| | Setting H Address Count | PasePort+9 |
| | Master/Count Direction | PasePort+10 |
| Not Used | For Future Use | PasePort+11.. |
| Slot #0 | Ports A, B, C & D | PasePort+16.. |
| Slot #1 | Ports A, B, C & D | PasePort+20.. |
| Slot #2 | Ports A, B, C & D | PasePort+24 |
| Slot #3 | Ports A, B, C & D | PasePort+28 |

Table III
Pin descriptions for the proposed bus.

| SYMBOL | I/O | DESCRIPTION |
|---|---|---|
| RSTDRV | O | To reset or initialize system logic upon power up or during a low line voltage outage. |
| OSC | O | Clock o/p of the programmable oscillator. |
| CK | I | To feed the external base clock. |
| CKSEL | I | To select internal or external base clock. |
| A0-An | O | Microprocessor address bus. |
| D0-D7 | I/O | Microprocessor data bus. |
| SD0-SD7 | I/O | Shared memory data bus. |
| ALE | O | Address latch enable. |
| -I/O CHRDY | I | Normally high, is pulled low by a memory or I/O device to lengthen I/O or memory cycles. |
| INTA, INTB | I | Interrupt request A, B. Used to signal the processor that an I/O device requires attention. |
| -PORTA,... , -PORTB | O | Ready decoded I/O addresses used with memory or I/O control lines. |
| -IOR | O | I/O read command. |
| -IOW | O | I/O write command. |
| -MEMR | O | Memory read command. |
| -MEMW | O | Memory write command. |
| -SMEMR | I | Shared memory read command. |
| -SMEMW | I | Shared memory write command. |
| SMEMF | O | Shared memory full flag. |
| -SMCLK | I | Shared memory address increment command. |
| DRQ1 | I | DMA Request 1 for DMA services. |
| -DAK0, -DAK1 | O | DMA Acknowledge 0 & 1 used to acknowledge DMA requests. |
| AEN | O | Address enable used to allow DMA transfers to take place. |
| T/C | O | Used to provide a pulse when the terminal count for any DMA channel is reached. |
| 5,12,GND, -5, -12 | O | Voltages available to the Add-on cards. |

## 5. CONCLUSIONS

- A programmable bus system prototype based on the ISA bus has been designed, built and tested experimentally. It consists of two stages of buffering circuits between add-on cards and the main PC motherboard. The bus system provides four slots, each one equipped with two interrupt lines, and an independent programmable oscillator that can be programmed over a wide range of frequencies.
- Table III outlines an overview of the label and function description of each pin in the bus system.
- Any add-on card that requires four or less I/O ports to be connected to the PC will not need any decoding circuits.
- An 8k bytes of static RAM is available for the slots. It can be used as a temporary storage for data acquisition or distribution without using the original data/address buses of the PC.
- Installing the proposed bus system on a PC will occupy only one ISA slot, which is still available on newer PCs.
- The current bus system performance is fair enough for applications with normal speed requirements. However, for applications that demand high data traffic, a wider data bus, and larger shared memory are required. Also, a local microprocessor can be used in the bus motherboard to implement the basic tasks without disturbing the main processor.

## REFERENCES

[1]. **R.L. Krutz**, "Interfacing techniques in digital design with emphasis on microcomputers", John Wiley & Sons, 1988.

[2]. **B. B. Brey**, "Microprocessors and peripherals: hardware, software, interfacing & applications", Macmillan publishing co, 1998.

[3]. **M. A. Havell, P.S. Tang & S. Fitch**, "A modular and innovative software package for multichannel data acquisition" IEEE Trans. on instru. & measurements Vol.37, No.4, December 1988.

[4]. **http:// www.pcguide.com**

[5]. **K. M. Al-Aubidy, N. S. Abdullah & M. A. Al-Taee**, " Design & implementation of a PC-based events recorder", Engineering & technology magazine , Vol.13, No.6, pp.(119-132), June 1994.

[6]. **F. Mora, A. Sebastia, H. Muller, C. Fernandes & Y. Ermoline**, "Design of a high-performance PCI interface for an SCI network", Computing & control eng. journal, Vol.9, No.6, pp.(275-282), December 1998.