

Distributed Direct Memory Access Arbitration

Vesselina Kirilova Zaharinova-Papazova

Faculty of Electronics and Electronic Technology
Technical University-Sofia

Abstract

This paper relates to the successful experience in implementing distributed Direct Memory Access (DMA) arbitration. It presents a Multiprocessor System for control and data acquisition in Power engineering. The system includes a Central Processor Unit (CPU) and 8 peripheral blocks. The peripherals communicate with the CPU through Static Random Access Memory. This RAM is located at a parallel bus common to all system blocks. Every block includes its local arbitrating device, which determines whether the bus is available to it or not. This paper discusses in details the implementation of distributed bus arbitration. It summarizes the advantages and disadvantages of the designed system.

The paper mentions some problems referring to the reliability of the implemented system. The main possibilities to solve them are also described. In the paper are marked some difficulties we came across through the design.

I. INTRODUCTION

The main goal was to implement a reliable microprocessor system with high performance. The CPU is the brain of the computer system and performs most of the high level computation functions. Multiprocessor system was designed to ensure high performance. The main question was "How to implement the data transfer between microprocessors?". There are several possibilities. Two of them have been discussed. The best one is to use Dual-Port RAM's located on each peripheral block. This way ensures the highest time performance, because it allows the main CPU and Peripheral microprocessor to have access to the memory at one and the same time, except when writing on the same address simultaneously. In this case a special arbitrating device is not needed. The main disadvantage is the high cost. For word data transfer 16 Dual-Port RAMs has to be used. So I decided to replace them with a simple SRAM, common to all blocks of the system. One of the merits of the designed Multiprocessor system is the organization of the bus access.

II. Block Diagram of the designed Multiprocessor System

The designed system is based on Micro-controller Units (MCU) of Motorola Company. Its block diagram is presented in the *Figure.1*. This is a reliable multiprocessor system with high performance intended for data acquisition and control in Power engineering. It includes a Central Processor Unit (CPU) and 8 peripherals blocks.

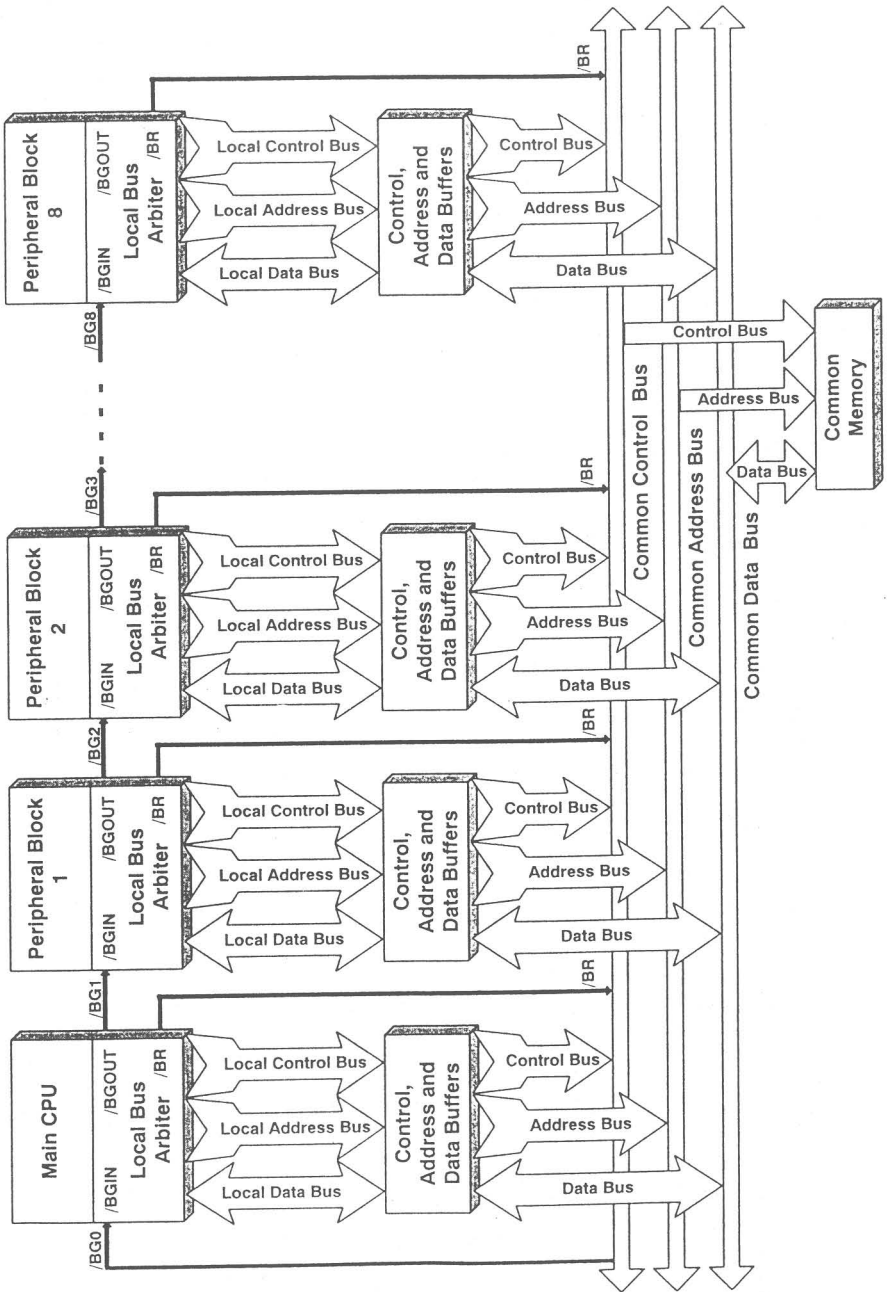


Figure.1

Bus mastering refers to the ability of devices connected to the common memory (common bus) to perform DMA or Direct Memory Access without the use of the main CPU. This is usually implemented in one of the two ways: a) Third Party software control or b) in hardware in the device level controller, such as EIDE Controller or a Sound Card or Video Card. Bus mastering devices allow the peripherals to access main memory without using the main CPU to do a memory fetch and relieves a lot of unnecessary pressure from the CPU to do other things. The designed multiprocessor system fulfills this principle. To improve time performance the main CPU is unloaded of trivial tasks such as bus mastering DMA and data acquisition. In this case intelligent peripheral blocks collect data from sensors, make some calculations and supply the results to the CPU via common memory. Thus the CPU can perform all important computations without losing time for communication with the peripherals.

As I mentioned above the peripherals communicate with the CPU through Static Random Access Memory (SRAM). This RAM is located at a parallel bus common to all system blocks. The first difficulty I came across was during the implementation of bus arbitration. One possibility is to use DMA controller, but there is a limitation in the number of DMA channels. For that reason I designed a dislocated arbitrating system. This means that every block (The main CPU and the peripherals) includes its own arbitrating device, which determines whether the bus is available to it or not. The common memory is divided in 8 sectors. Unlike the main CPU, which has full access to the memory, peripherals have access only to one sector corresponding to its purpose. The maximal size of the memory is 16kB.

III. Distributed Direct Memory Access Arbitration

As it is mentioned above every block in our system includes its local arbitrating device. It is implemented by using reprogrammable MACH devices. This makes the system very flexible. We could use different number of peripheral blocks according to our needs. An insignificant change in MACH equations has to be done in this case. You could see the block diagram of Bus arbitration in *Figure 2*.

In contrast to the serial bus the parallel is more limited in length because of the high cost of the multiconductor cable and connectors and for electrical reasons. The signal does not travel at the exact speed in all conductors, so the data rate must be reduced for longer cable runs. Some parallel busses are implemented by using cables and connector (for example, SCSI). However, the rest of this description will discuss parallel bus that is implemented on the motherboard of the designed device.

As it is known the parallel bus is the collection of electrical connections, connectors and voltages, timing, and functionality defined for communication between blocks. The system has a unique bus interface Also it is very important what kind of devices are connected to the bus. The bus environment has to be homogeneous. A CMOS RAM is used , so the buffers and MACH's are CMOS compatible.

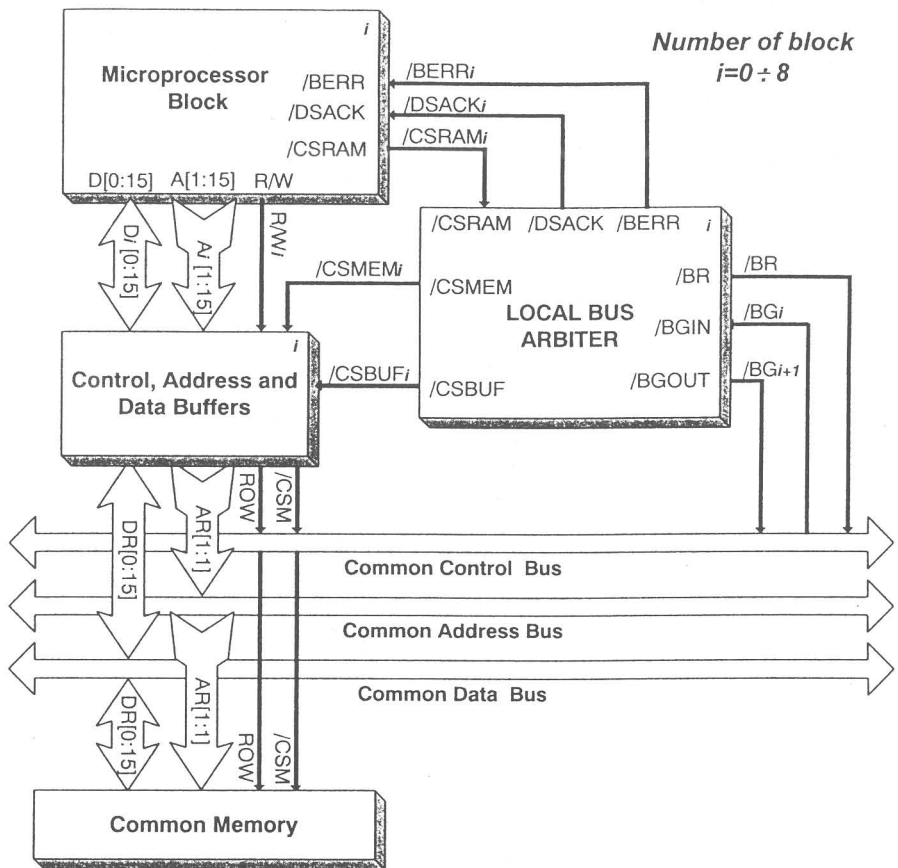


Figure.2

Bus arbitration protocols determine when a device can become bus master. Bus arbitration requests are recognized during normal processing. When two or more devices attempt to become bus master at the same time, the one having the highest priority obtains the bus first. The priority of the device depends on the place of it on the bus. The Main CPU has the highest priority and Peripheral block 8 – the lowest. The common memory is located on the main CPU board. As you could see in *Figure.3* only word access to the memory is allowed.

The protocol sequence on the bus is:

- The devices that need to obtain the bus must assert chip select line dedicated to the Common RAM ($/CSRAM$). $/BG_{i+1} = /BG_i$

- The arbiter determines whether there is someone active on the bus ($/BR=0$). If the bus is obtained from other devices the arbiter waits until it is released ($/BR=1$). The local arbiter asserts $/BR$ in case of free bus. $/BG_{i,i+1}=/BG_i$
- In response to $/BR$ the bus grant signal ($/BG_i$) is asserted to indicate that the bus is available. $20ns(t1)$ later, the local arbiter checks if this signal is still active. In case of $/BG_i=0$, $/BG_{i,i+1}$ is set to inactive state ($/BG_{i,i+1}=1$) Otherwise it waits for a stable $/BG_i$ and $/BG_{i,i+1}=/BG_i$
- The local arbiter waits for $80ns$ before performing the data transfer. This time depends on the number of blocks on the bus. It is because the signal $/BG$ does not travel at exact speed. There are delays in the chain of the local arbiters.
- $20ns$ later the device assumes mastership and enables buffers of the bus.
- Perform data transfer and asserts chip select signal of the memory data strobe acknowledge signal ($/DSACK$).
- The bus is released: $/BR=$ "high-impedance" and $/BG_{i,i+1}=/BG_i$.

It takes minimum $250ns$ to accomplish one DMA access cycle.

If this sequence is not finished in the next $16ms$ bus error signal is asserted ($/BERR$). In *Table.1* you could find some time characteristics referring to the timing diagram in the *Figure.3*.

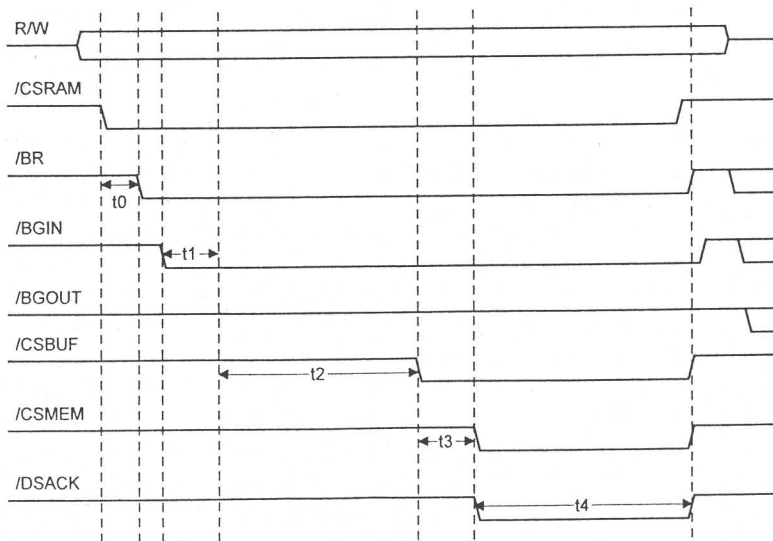


Figure.3

Table.1

Symbol	Characteristic	Time
t0	Min	5ns
t1	Nominal	20ns
t2	Nominal	80ns
t3	Nominal	20ns
t4	Nominal	50ns

For better software synchronization between microprocessor blocks there is a common interrupt request signal (INT) on the bus. INT is generated at every 1ms. Thus the possibility two or more bus requests to occur at the same time decreases.

IV. SUMMARY

A balance between high performance, reliability and low cost has been the main goal during the design of the described system.

- ✓ Multiprocessor system is designed to ensure high performance. For data transfer between different processors a common memory has been used. The access to this memory is through parallel bus.
- ✓ High reliability is achieved with very careful design of each microprocessor boards and the motherboard. Care should be taken to make the antennas as ineffective as possible, i.e., make the areas enclosed by the outward and return lines as small as possible.
- ✓ Also it is very important what kind of buffer, memory and MACH devices have been chosen. The bus environment has to be homogeneous. In some cases a termination of the bus is necessary to avoid line reflections.
- ✓ Number of blocks communicating with the common memory is restricted because the parallel bus length is limited for electrical reasons.
- ✓ The designed system is very flexible because it could be changed easily to supply all contemporary needs. If a change in arbitration sequence is necessary, it takes minutes to reprogram the MACH devices.
- ✓ There is a interrupt request signal on the bus which has been used to synchronize the software. In this way it is less probable two or more bus requests to occur simultaneously.
- ✓ The cost is reasonable.

References

- [1] Texas Instruments, "Design Considerations for Logic Products. Application Book", Texas Instruments Deutschland GmbH, 1998
- [2] Timkov I., Markov S, Aleksandrova Z., Aleksandrov A., "Microprocessor systems for control", Technika, 1986
- [3] Avreski D., "Microprocessor systems with increased reliability", Technika, 1984