

# Power Optimization in Microprocessors based on Parallel Data Transfer Schemes

Karan Singh<sup>1</sup>

Department of Electronics and Communication Engineering  
Mangalmai Institute of Engineering & Technology  
Greater Noida, India

Amit Kumar Mourya<sup>2</sup>

Department of Electronics and Communication Engineering  
Mangalmai Institute of Engineering & Technology  
Greater Noida, India

Dhananjay Singh<sup>3</sup>

Department of Electronics and Communication Engineering  
Mangalmai Institute of Engineering & Technology  
Greater Noida, India

**Abstract**— Managing power is a very broad topic, and spans software and hardware. It is important to understand the role of power management in a successful microprocessor, microcontroller and IoT deployment, and how to manage power efficiently for remote devices and long-lived devices. Our main concentration here is on microprocessor power management according to data flow. A microprocessor is an electronic chip which has computing and decision making capability. It is a central processing unit on a single integrated circuit chip containing millions of very small components including transistors, resistors, and diodes that work together. It is very important to take into account how the millions of very small components will behave and how this behavior affects energy consumption. So, early optimization of low power is very essential and important. In this paper we optimize power saving approach on the basis of data transfer schemes. The power optimization approach is based on the various parallel data transfer schemes. The Result of power optimization method shows that the approach is effective in microprocessor and microcontroller.

**Keywords**—Data Transfer Schemes, IoT, Low Power, Microprocessor, Microcontroller, Parallel Data Transfer.

## I. INTRODUCTION

Data transfer or transfer is any information that is transferred from one location to another through some communication method. There will be several IO and memory devices are connected to transfer data between memory and mp, No much problem for transfer data communication between microprocessor and memory, since same technology is used in the microprocessor and memory and speed of both is compatible.

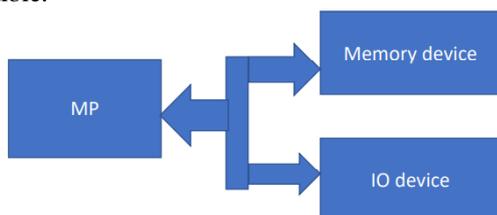


Fig. 1. Data Transfer in Microprocessors

Now if the speed of microprocessor or memory or input output devices is mismatch then Data transfer between the mp and IO devices or memory is not so easy.

## II. DATA TRANSFER SCHEMES

To handle such cases we need various Modes of Data Transfer schemes to optimize power because working time of ALU section will also be responsible of power consumption.

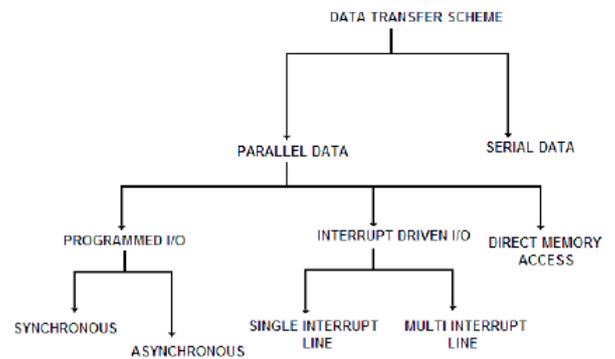


Fig. 2. Data Transfer Schemes

### A. Programmed I/O Data Transfer

Programmed I/O Data Transfer scheme of microprocessor is a simple parallel data transfer scheme. This method of data transfer is generally used in the simple microprocessor systems. It is obvious that where speed is unimportant. This method uses instructions to get the data into or out of the microprocessor. Programmed I/O Data Transfer scheme of microprocessor can be work on synchronous or asynchronous mode. The data transfer can be synchronous or asynchronous it completely depends upon the type and the speed of the I/O devices.

#### a) Synchronous Data Transfer

Synchronous type of data transfer can be used when the speed of the I/O devices matches with the speed of the microprocessor. So for synchronization established between I/O device and microprocessor we need common clock pulse. This common clock pulse synchronizes the microprocessor and the I/O devices. Synchronous type of data transfer scheme because of the matching of the speed, the microprocessor does not have to wait for the availability of the data. The microprocessor immediately sends data for the transfer as soon as the microprocessor issues a signal.

*b) Asynchronous Data Transfer*

The asynchronous data transfer method is used when the speed of the I/O devices is slower than the speed of the microprocessor. Because of the mismatch of the speed, the internal timing of the I/O device is independent from the microprocessor. That is why two units are said to be asynchronous to each other. The asynchronous data transfer is normally implemented using ‘handshaking’ mode. Now question is what is handshaking mode? In the handshaking mode some signals are exchanged between the I/O device and microprocessor before the data transfer takes place. By this handshaking the microprocessor has to check the status to the input/output device. Now if the device is ready for the data transfer or not. First step of microprocessor is initiates the I/O device to get ready. Then status of the I/O device is continuously checked by the microprocessor. This process remain continues until the I/O device becomes ready and after that microprocessor sends instructions to transfer the data.

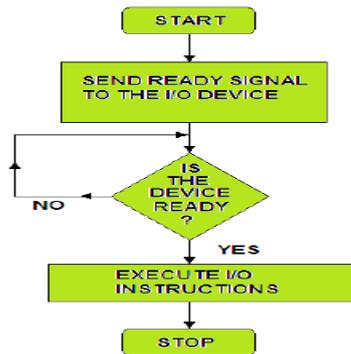


Fig. 3. Asynchronous Data Transfer Flow Diagram

Now form this bellow figure, the microprocessor sends a ready signal to I/O device. When the device is ready to accept the data, the I/O device sends an ‘ACK’ (Acknowledge) signal to microprocessor. By sending ACK, it indicates that the I/O device has acknowledged the ‘Ready’ signal. Now finally it is ready for the transfer of data.

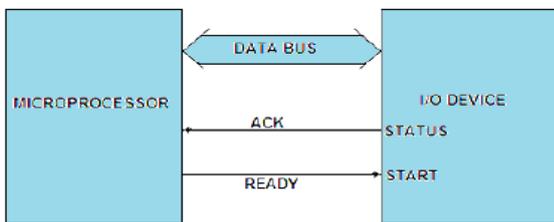


Fig. 4. Asynchronous Transfer

**B. Interrupt Driven Data Transfer**

As we saw that in the programmed I/O data transfer method, microprocessor is busy all the time in checking for the availability of data from the slower I/O devices and it also busy in checking if I/O device is ready for the data transfer or not. In other words in this data transfer scheme, some of the microprocessor time is wasted in waiting while an I/O device is getting ready. To overcome this problem interrupt driven I/O data transfer introduced.

The interrupt driven I/O data transfer method is very efficient because no microprocessor time is wasted in waiting for an I/O device to be ready. In this interrupt driven I/O data

transfer method the I/O device informs the microprocessor for the data transfer whenever the I/O device is ready. This is achieved by interrupting the microprocessor. As we know that the interrupt is hardware facilities provided on the microprocessor.

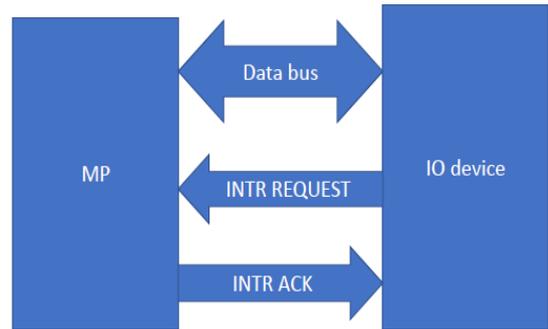


Fig. 5. Interrupt Driven Data Transfer in Microprocessors

Now come to the working process of interrupt driven I/O data transfer. So the beginning the microprocessor initiates data transfer by requesting the I/O device ‘to get ready’ and then continue executing its original program rather wasting its time by checking the status of I/O device. Whenever the device is ready to accept or supply data, it informs the processor through a control signal. This control signal known as interrupt (INTR) signal. In response to this interrupt signal, the microprocessor sends back an interrupt acknowledge signal to the I/O device. By sending acknowledgement it indicates that it received the request. It then suspends its job after executing the current instruction. It saves the contents and status of program counter to stack and jumps to the subroutine program.

This subroutine program is called Interrupt Service Subroutine (ISS) program. The ISS saves the processor status into stack; and after executing the instruction for the data transfer, it restores the processor status and then returns to main program.

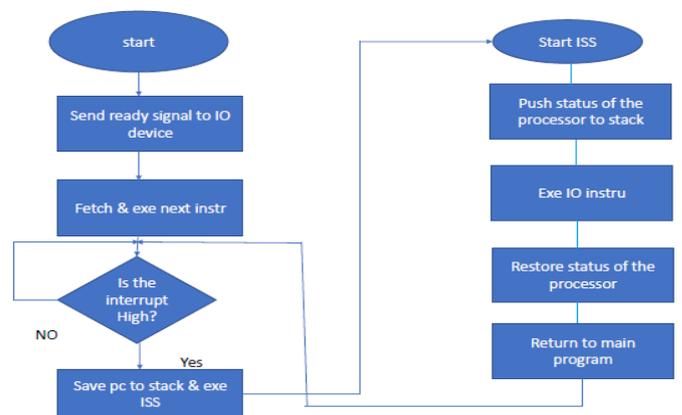


Fig. 6. Interrupt Service Subroutine flow diagram

As already discussed, several input/output devices may be connected to microprocessor using Interrupt Driven Data Transfer Scheme. Following interrupt request configuration may arise while interfacing the I/O devices to microprocessor.

a) Single Interrupt System

When only one interrupt line is available with the microprocessor and several I/O devices are to be connected, then the method is known as Single Interrupt System.

b) Multi Interrupt System

When the microprocessor has several interrupt terminals and one I/O device is to be connected to each interrupt terminal, then it is known as multi interrupt system. In this scheme, the number of I/O devices to be connected to the interrupt lines should be equal to or less than the number of interrupt terminals. In this way one device is connected to each level of interrupt. So when a device interrupts the microprocessor, it immediately knows which device has interrupted. Such an interrupt scheme is known as vectored interrupt.

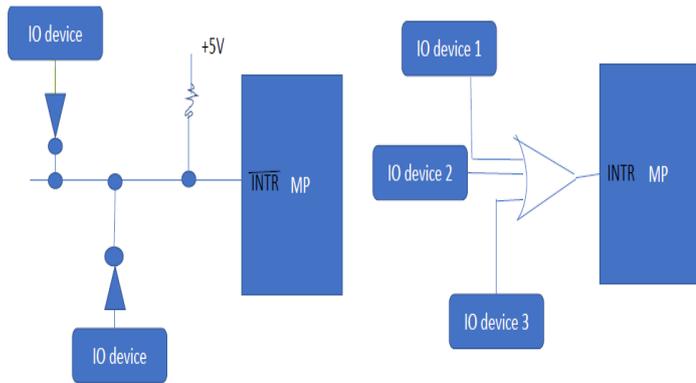


Fig. 7. Multiple Interrupt System

C. Direct Memory Access (DMA) Data Transfer

As we discussed earlier that in programmed I/O or interrupt driven I/O methods of data transfer between the I/O devices and external memory is via the accumulator. Now think for bulk data transfer from I/O devices to memory or vice-versa, these two methods discussed above are time consuming and quite uneconomical even though the speed of I/O devices matches with the speed of microprocessor because in those methods the data is first transferred to accumulator and then to concerned device.

To overcome those problem direct memory access data transfer method is introduced. The Direct Memory Access (DMA) data transfer method is used for bulk data transfer from I/O devices to microprocessor or vice-versa. In this method I/O devices are allowed to transfer the data directly to the external memory without being routed through accumulator. For this reason the microprocessor relinquishes the control over the data bus and address bus, so that these can be used for transfer of data between the devices.

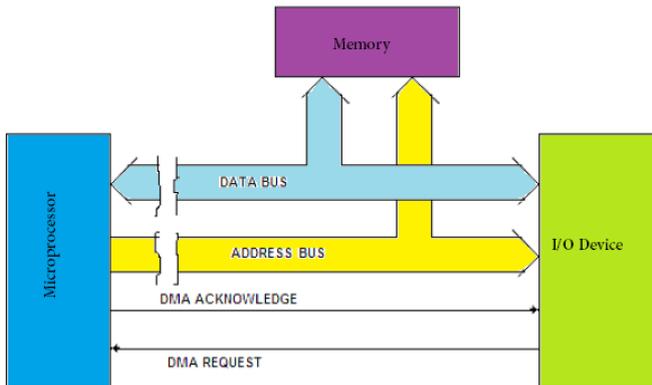


Fig. 8. Direct Memory Access Operation

III. POWER OPTIMIZATION

Power consumption is an important parameter for digital CMOS circuit. Microprocessors consume large amount of power because it include a large number of very small components and different data transfer schemes. So, it has become a challenge to reduce the power dissipation of the circuit. We can improve the performance, speed of the circuit with the help of reduction in power. So power optimization of microprocessor using data transfer is necessary.

There several power optimization techniques can be applied to minimize dynamic power consumption such as

- System Level
- Parameter Scaling
- I/O Standard
- Architectural
- Logic Level Power Minimization

IV. RESULTS & CONCLUSION

The simulation results for the microprocessors are shown in Fig. 9, which verified the successfully by different data transfer schemes for different architectures. The ISIM simulator is used for the verification of the results. Our experimental results show that significant amount of energy savings (up to 58%) can be achieved in microprocessors during data transfers with no or minimal performance penalty if the correct parameter combination is used.

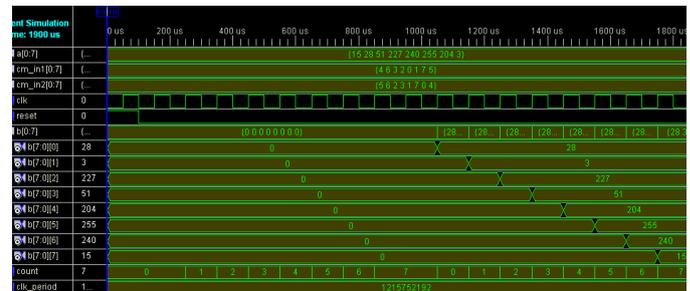


Fig 9. Simulation results

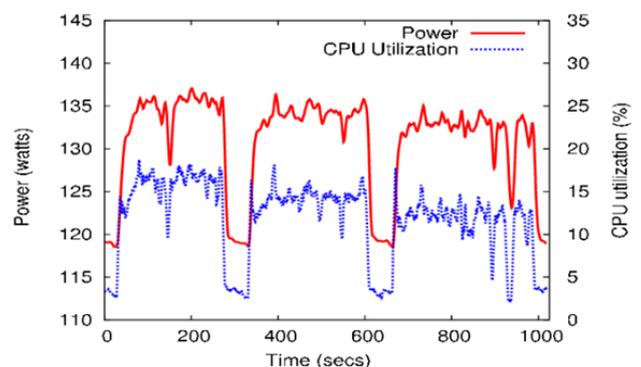


Fig 10. Power Consumption versus CPU utilization during data transfer

All the data transfer schemes have been displayed in behavioral style utilizing VHDL. The DMA engineering is the best design regarding execution, adaptability and force utilization among all the data transfer schemes.

V. REFERENCES

[1] R. Basmadjian, H. De Meer, Evaluating and modeling power consumption of multi-core processors, in: Future Energy Systems:

- Where Energy, Computing and Communication Meet (e- Energy), 2012 Third International Conference on, 2012, pp. 1-10.
- [2] A. Family, 10h server and workstation processor power and thermal data sheet.
- [3] E. Pinheiro, R. Bianchini, E. V. Carrera, T. Heath, Dynamic cluster reconfiguration for power and performance, in: Compilers and operating systems for low power, Springer, 2003, pp 75-93.
- [4] M. Gupta, S. Singh, Energy conservation with low power modes in ethernet lan environments, in: IEEE INFOCOM (MiniSymposium) 2007.
- [5] D. Economou, S. Rivoire, C. Kozyrakis, P. Ranganathan, Full-system power analysis and modeling for server environments, in: Proc. of Workshop on Modeling, Benchmarking, and Simulation, 2006.
- [6] D. Brooks, V. Tiwari, M. Martonosi, Wattch: a framework for architectural-level power analysis and optimizations, in: Proceedings of the 27th annual international symposium on Computer architecture, ISCA '00, ACM, New York, NY, USA, 2000, pp. 83-94.
- [7] P. Mahadevan, P. Sharma, S. Banerjee, P. Ranganathan, A power benchmarking framework for network devices, in: In Proceedings of IFIP Networking, May 2009.
- [8] T. Do, S. Rawshdeh, W. Shi, ptop: A process-level power profiling tool, in: Proc. of HotPower 2009, 2009.