Srijan Chatterjee, V.V.Subrahmanya Kumar Bhajana / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 3, May-Jun 2013, pp.543-547

Design and Implementation of 8 Bit Asynchronous Microcontroller

Srijan Chatterjee*, V.V.Subrahmanya Kumar Bhajana**
*(M.Tech. VLSI Design and Embedded System, School of Electronics Engineering. KIIT University. Orissa. INDIA)
** (Associate Professor, School of Electronics Engineering. KIIT University. Orissa. INDIA)

ABSTRACT

The synchronous microcontroller is a common processor found in many embedded system. By using asynchronous design techniques, the performance of the microcontroller is increased. Through the simulation and existing synchronous design tools in the asynchronous design flow, a "clock enabling technique" is simulated and then implemented. The asynchronous architecture includes an ALU, a decoder, a RAM, a register and a clocking element. This paper mainly deals with design of 8 bit asynchronous microcontroller simulation in Verilog using Xilinx.

Keywords – ALU, Asynchronous, Clock enabling technique, clock skew, Decoder, Event dependant, Embedded system, Global clock, Handshaking technique, Master-slave, RAM, Register, Synchronous, Time dependant, Verilog, VHDL, Xilinx.

1. INTRODUCTION

In synchronous design, the clock period of global clock must be keep large enough than the worst case performance of the slowest operation. But in case if high frequency operation for the gradually increasing of complexity and the size of the circuit, the propagation time of the clock in different paths are different, it may cause the circuit to malfunction.

Main drawbacks of the synchronous design are power consumption and clock skew. The main advantages of the asynchronous design are lower power consumption, no clock skew, better technology migration and less global timing issues.

1.1 Asynchronous Design flow with synchronous tools

Even though complete asynchronous design solution do not exist, one can make use of existing synchronous design tools in asynchronous design flow. In this flow, each design is partitioned into blocks, and each block is controlled by clock signal and a clock controlling signal. Simulation tool Xilinx is used to simulate the functionality of the asynchronous design. Once the design has been functionally simulated, it must now be implemented. While an entire asynchronous circuit can not be designed in synchronous tool, individual logic blocks can. During the design of the asynchronous microcontroller chip, VHDL (Very high speed integrated circuit Hardware Description Language) or Verilog is used to generate certain blocks.

1.2 Deference between Synchronous, Asynchronous and Master – Slave Design

The main characteristics of the synchronous design is, it is completely "Time Dependant". That means, suppose at t = 2 second we need the output so, all blocks of circuit will generate the output at t = 2 second. As in Fig.1.

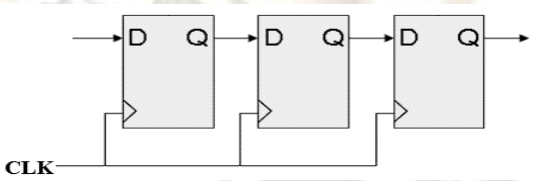

Fig.1: Synchronous Design

This types of design is comparatively fast than other design. But, since the clock frequency is acting on each blocks during the whole operation time, the power consumption is more. On the other hand, the clock period of the global clock must be keep large enough than the worst case performance of the slowest operation. Otherwise, we face the problem regarding "clock skew", shown in Fig.2.

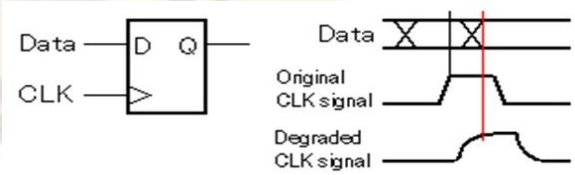

Fig.2: Clock Skew

On the other hand, the fundamental characteristic of Asynchronous design is, it is fully event dependant. That means, the output of a block depends on the output of previous block. And when a particular block performs, the other blocks are paused. So, it do not misuse the clock. As in Fig.3.

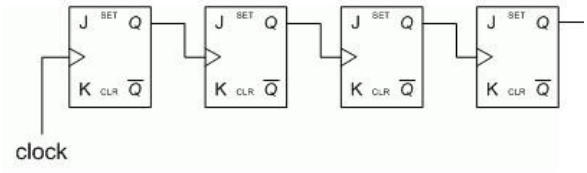

Fig.3: Asynchronous Design

Last but not the least, in the case of master – slave design, the master circuit performs at a particular edge or level of the clock and the slave circuit performs at the other edge or level of the clock. As in Fig.4.

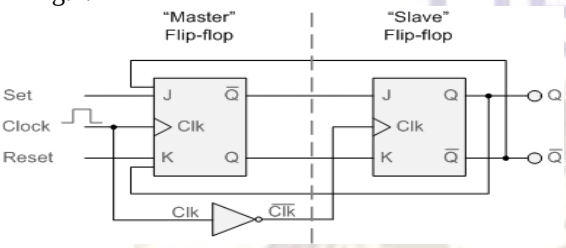

Fig.4: Master – Slave Model

2. ASYNCHRONOUS MECHANISM

Previously introduced asynchronous design are based on "Handshaking Technique". In that case, the power consumption was reduced than the synchronous model. But the design of the Handshaking model is very complex. In this paper a new clock controlling technique is introduced which is more easier to write codes and implement.

2.1 Clock Controlling Technique

In the asynchronous implementation of the microcontroller, the synchronization between any two blocks is implemented through a "clock enabling signal", which will be controlled by the previous block. When the operation of a block will be done, a high signal will be generated. The signal will be the clock enabling signal for the next block, then the clock will act on the next block. Clock controlling technique block diagram depicted in fig.5.

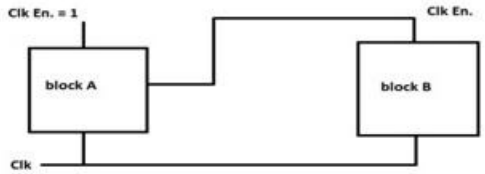

Fig.5: Clock Controlling Technique

2.2 Block Diagram of basic blocks of Microcontroller

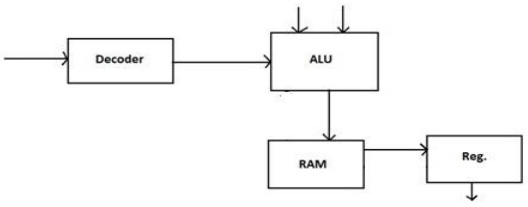

Fig.6: Block diagram of basic blocks of a microcontroller

A block diagram of basic blocks of microcontroller is illustrated in fig.6. The ALU gets instructions from decoder and two operand externally. Then it processes and send the result to RAM and store the result to register via RAM.

2.3 Proposed Architecture

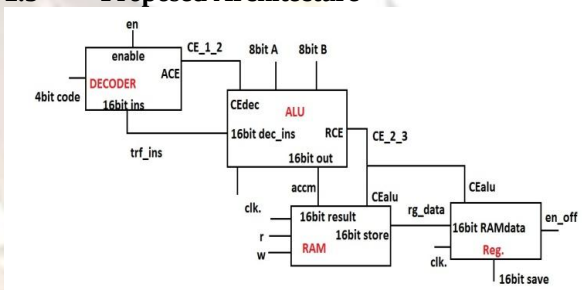

Fig.7: Proposed Architecture of Microcontroller

Where,
trf_ins: transfer instructions from decoder to ALU.
CE_1_2: Clock enable signal from stage 1 to stage 2.
CEdec (same as CE_1_2): Clock enable by decoder.
CE_2_3: Clock enable signal from stage 2 to stage 3.
CEalu (same as CE_2_3): Clock enable by ALU.

2.4 Working Principle

Fig.7 shows that the proposed architecture of asynchronous microcontroller. This microcontroller can perform sixteen arithmetic and logical operations viz. add, subtraction, multiplication, division, compare, shift right, shift left, increment, decrement, AND, OR, NOT, NAND, NOR, XOR, XNOR. These will be selected by a 4 to 16 decoder.

This decoder has one output named "ACE (ALU Clock Enable)". It will be high when any one out of the sixteen operations will be selected. This signal will enable the clock for ALU, then ALU will perform.

ALU can performs above sixteen operations on two 8bit operands. ALU gets sixteen operation bits from decoder and two 8bit numbers which are user defined.

On the basis of sixteen operation bits ALU generates 16bit result. After completing any

operations among the above sixteen it generates one high signal "RCE (RAM & register Clock Enable)" to enable clock for RAM and register together as before decoder did for the ALU.

Whenever the register become clock enable the "en_off" bit will be zero, it will disable the decoder. So, the clock enable output of the decoder will be also zero. As a result, ALU will be deactivated. When the RAM is clock enabled, the result from ALU, enters to the RAM and when the "w (write)" bit of the RAM become high, the result is stored to the register.

For this part, we have to first input two 8bit numbers then have to select 4bit decoder input for a particular operation then have to make enable the decoder and at last have to make high the write bit of the RAM to store the result.

3. SIMULATION RESULTS

The simulation results are performed using Xilinx. The schematic of proposed architecture is shown in Fig.8. The simulations are performed for the operations add, subtraction, multiplication, division, compare, shift right, AND and OR. The obtained results are depicted in Fig.9 to Fig.16.

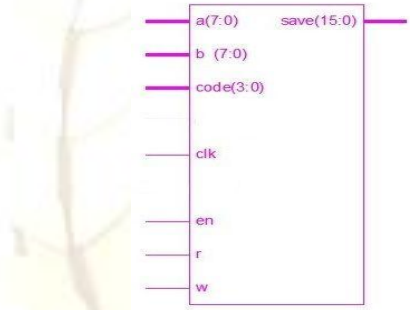

Fig.8: Schematic diagram of the microcontroller

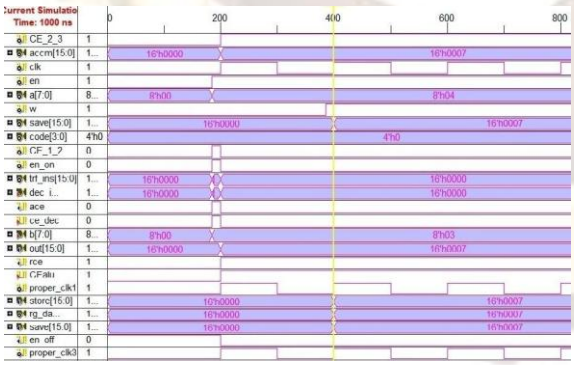

Fig.9: Simulated result of ADD operation

This is the simulation result for ADD (4+3) operation. Whenever enable is ON and decoder gets 4bit data then 16bit data and clock enable signal for ALU are generated. After that ALU gets two 8bit data and generates result and clock enable signal for next stage.

When the 'w' is high, the result transferred to register and saved. And the 'en_off' bit will go to zero. As a result decoder is OFF.

In this is way total operation is completed.

As follows:

Fig.10: Simulated result of SUB (4-3) operation

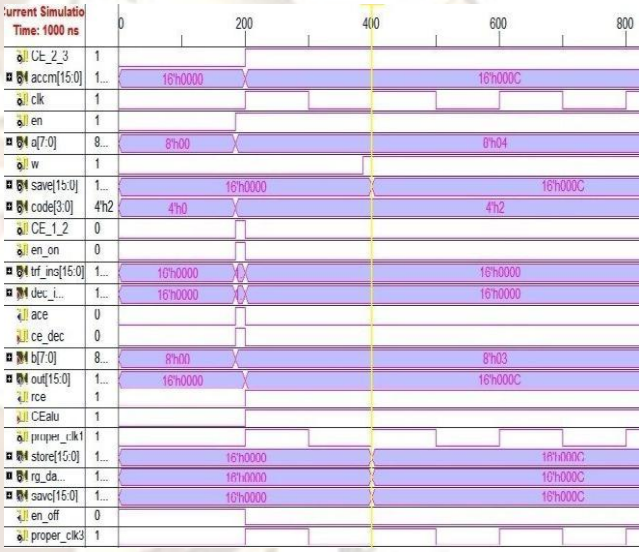

Fig.11: Simulated result of MUL (4*3) operation

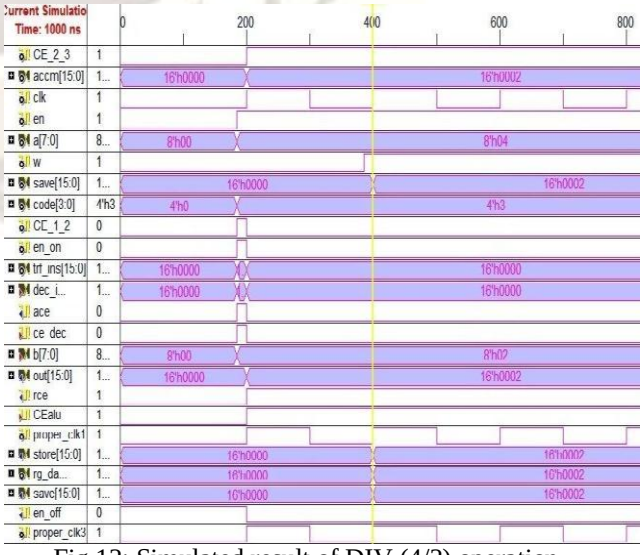

Fig.12: Simulated result of DIV (4/2) operation

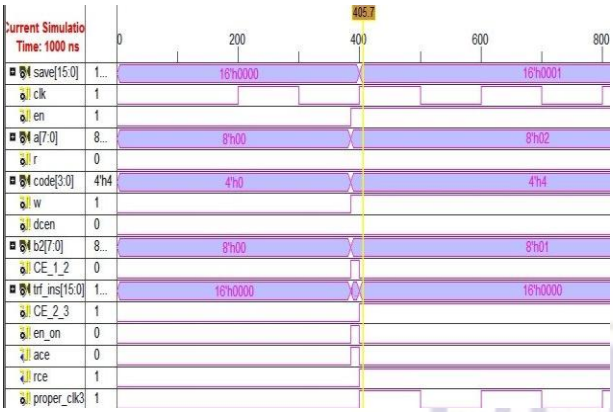

Fig.13: Simulated result of COMP between 2 and 1

This is the simulation result of comparison between 2 and 1. If first input 'a' is larger than second input 'b' then the output will be 1, otherwise 0.

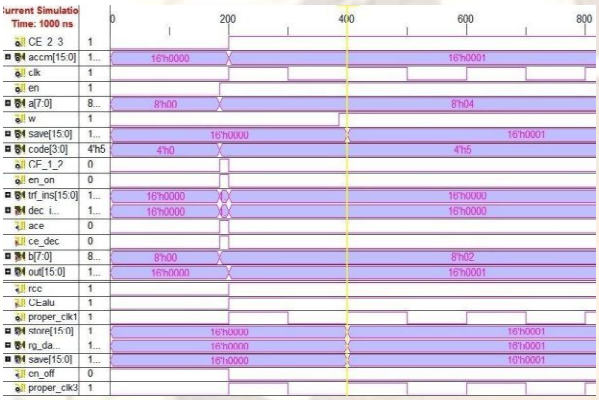

Fig.14: Simulated result of Right Shift (4 shifted by 2bit)

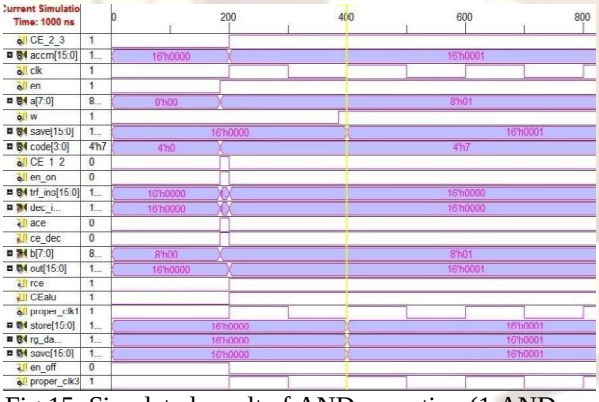

Fig.15: Simulated result of AND operation (1 AND 1)

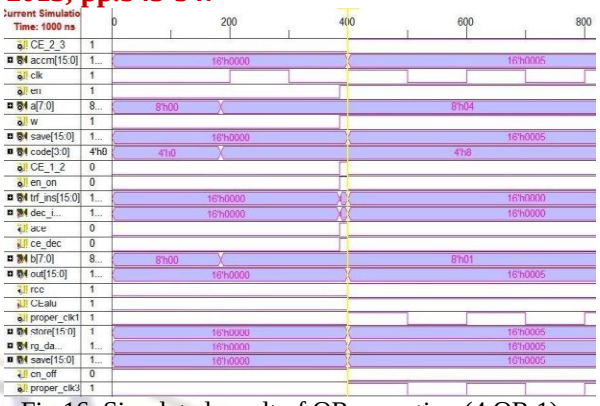

Fig.16: Simulated result of OR operation (4 OR 1)

4. CONCLUSION

This paper is mainly deals with the simulation and its implementation of the asynchronous design of an 8bit microcontroller with clock controlling technique. Using which technique in a circuit block we can make activated the working block only and the other blocks are paused.

The above technique is very much useful in low power applications viz. Bio medical instruments, mobile phones etc.

5. ACKNOWLEDGEMENT

First of all I would like to thank my guide B.V.V.Subrahmanya Kumar for guiding me throughout the research part of the thesis. I am really privileged to have opportunity to work with him.

I will also thank Prof. S.K.Mandal (Associate Dean, School of Electronics Engineering, KIIT University, Bhubaneswar, Orissa) for supporting me throughout my research work. I am also thankful to Assistant Professor Mr. Ankit Shivare for all his guidance and pleasant work everyday in the lab. I am also thankful to all the staffs and employees of School of Electronics Engineering of KIIT University.

I am really also grateful to my friends Pratik Ganguly and Abhinab Anand for all their helping hands during my days of research.

Finally, last but not the least I pay all my sincere gratitude towards my loving mother and caring father for supporting throughout my sorrow and best days of my life.

REFERENCES

[1] Raj Kamal, *Embedded systems* (The McGraw Hill Companies)

[2] Muhammad Ali Mzidi, Janice Gillispie Mazidi, *The 8051 microcontroller and embedded sysytems*

[3] Peter Marwedel, *Embedded system design* (Springer)

[4] Kok – Leong Chang, Bah – Hwee Gwee, Centre for Integrated Circuits and Systems, Nayang Technological University, Singapore, *A low – energy low – voltage asynchronous 8051 microcontroller core*, 2006 IEEE.

[5] Hans van Gageldonk, Eindhoven University of Technology, The Netherlands, Kees van Berkel, Ad Peeters, Philips Research Laboratories, The Netherlands, Daniel Baumann, Daniel Gloor, Gerhard Stegmann, Philips Semiconductors, Zurich, Switzerland, *An asynchronous low – power 80C51 microcontroller*.

[6] Kok – Leong Chang and Bah – Hwee Gwee, Centre for Integrated Circuits and Systems, Nayang Technological University, Singapore, Yuanjin Zheng, Integrated Circuits and Systems Laboratory, Institute of Microelectronics, Singapore, *A semi – custom memory design for an asynchronous 8051 microcontroller*, 2008 IEEE.

[7] YU Ying, ZHOU Lei, MIN Hao, ASIC & System State – Key Lab, Fudan University, Shanghai, China, *Design and vlsi implementation of an asynchronous low power microcontroller*.

[8] Jamin M.C.Tse and Daniel P.K.Lun, Department of Electronics Engineering, The Hong Kong polytechnique University, Hong Kong, *Asynmpu: a fully asynchronous cics micro – processor*, 1997 IEEE International Symposium on Circuits and Systems, June 9 – 12, 1997.

[9] Hans Jacobson, Department of Computer Engineering, Lulea University of Technology, *Asynchronous circuit design*, May 1996.