e-ISSN: 2349-9745
p-ISSN: 2393-8161

Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

Asynchronous VLSI Design: An overview

Jayesh Diwan[1], Rutul Patel[2], Kavita Jain[3]
[1,2]EEE Department, Indus University
[3]EC Department, MBICT Engineering college

Abstract — An asynchronous circuit, or self-timed circuit, is a sequential digital logic circuit which is not enabled by a global clock signal. Instead of that, they often use signals that indicate completion of operations and instructions, specified by handshaking or simple data transfer protocols. This type is contrasted with a synchronous logic circuit where changes to the output signal values are triggered by clock signal. Most of the digital devices today are using synchronous logic circuits. But future Nano CMOS VLSI Technology will not be compatible with synchronous designs as Variations in physical parameters affect timing. These increased timing variations will reduce robustness and performance. However asynchronous circuit designs have the potential to be faster, less power consumption, lowering electromagnetic interference, with modular designs in large systems (SoCs).

Keywords— Asynchronous Design, GALS - Globally Asynchronous Locally Synchronous, Isochronic Forks, QDI - Quasi Delay Insensitive Logic, Handshake Protocol

I. INTRODUCTION

Today’s digital design methodologies are dominated by the synchronous style, where execution of functions in a machine is kept in lock-step by a central timing generator. This has not always been the case. In the early days of digital design a variety of design styles flourished. One of the dominant research areas during this time was in a particular style called asynchronous design. Asynchronous circuits are sequential circuits that do not require any central timing to coordinate their internal operations. During the 1950's and 60's, many computers and systems were built using this type of circuit. However, during the 70's, the interest in asynchronous design started to decline and had all but disappeared in the early 80's. The reason for this was the rapidly growing complexity of digital systems. Synchronous circuits offered simplicity in their discrete and deterministic behavior. Designers only had to make sure that the clock period was large enough for the system to reach a stable state before the next clock tick. Asynchronous circuits, on the other hand, required very detailed examination to ensure a proper behavior, a task that became too hard as system complexity increased. However, with the advance of modern technology, system complexity and the demand for higher performance has revealed several inherent problems with the synchronous design style. Some of the more notable problems are clock skew due to high frequency operation, power dissipation due to clock distribution and high speed interfacing with the environment. Asynchronous circuits do not suffer from these problems and have therefore lately received renewed attention from researchers and designers.

II. FUNCTIONAL ASYNCHRONOUS DESIGNS

Twenty-six years ago, in December 1988, one research group at Caltech, California Institute of Technology submitted the world’s first asynchronous (“Clockless”) microprocessor design “Caltech Asynchronous Microprocessor (CAM)”, a 16-bit reduced-instruction-set computing (RISC) processor with 23,000 transistors for fabrication to MOSIS [1]. The design test was completed in February, 1989. The chips were found fully functional on first silicon.

It was followed by the first “Amulet” (a family of asynchronous clones of the ARM processor) from the University of Manchester in 1993, the TITAC, an 8-bit microprocessor from the Tokyo Institute of Technology in 1994, and the Amulet2e and TITAC-2 in 1997 [2]. The TITAC-2 is a 32-bit microprocessor. Also in 1997, the Caltech group designed the MiniMIPS [3], an asynchronous version of the 32-bit MIPS R3000 microprocessor. With 2 million transistors and a performance close to four times that of a clocked version in the same technology for the first prototype, the MiniMIPS is the fastest complete asynchronous processor ever fabricated. A group at Grenoble ported the Caltech MiniMIPS building blocks to standard cells to use in a 16-bit RISC. Other asynchronous chip experiments include the design of a Fast Divider at Stanford in 1991 and an Instruction-Length Decoder for the Pentium 4 by a research group at Intel in 1999. Few Low-power asynchronous microcontrollers also have been designed at Philips, Caltech, and Cornell [2]. Lutonium, an asynchronous 8051 microcontroller, is designed by Caltech in the year 2003. Specifically it's having energy efficiency of 0.5 nJ per instruction at 200 MIPS (in 0.18-micron CMOS at 1.8 V). It uses a new theory of energy complexity and new synthesis tools.

The First ever Licensable Clock less 32-bit RISC Processor Core ARM996HS was designed by Handshake Solutions in 2006. ARM996HS is having ARMv5TE architecture; Five-stage integer pipeline and high robustness converge. It was followed by HT80C51, 2nd generation Microcontroller from Handshake Solutions in 2007. It provides better soft error protection and extremely low power consumption (only 0.1 nano joules per instruction).

GA144 is multi-core Asynchronous processor from GreenArray Inc. by Charles H. Moore in 2010. Total 144 independent F18A computers can run up to 96 billion operations per second.

In 2013, A 8 bite, 8x single channel Interleaved redundant asynchronous SAR ADC was designed in 32 nm CMOS technology by IBM Research - Zurich: Rueschlikon, Switzerland. It is having higher sampling speed of 1.2GS/s at 1V, redundant capacitive DACs (CDAC), multi-bit decisions per step and interleaved designs. The ADC achieves 39.3dB SNDR and 34fJ/conversion-step with a core chip area of 0.0015mm^2.

III. ASYNCHRONOUS DESIGN CONCEPT

The basic idea behind asynchronous design is that a digital circuit is asynchronous when no clock is used to implement sequencing. Such circuits are also called “clockless”. The various asynchronous approaches differ in their use of delay assumptions made to implement sequencing. A circuit is called delay-insensitive (DI) when its correct operation is independent of any assumption on delays in operators and wires except that the delays are positive and finite [2]. In 1990, it was proved that in a model where all delays are exposed - the building blocks are elementary logic gates with a single Boolean output - the class of completely delay-insensitive circuits is very limited [3]. Most circuits of interest to the digital designer fall outside this class. But it can be proved that a single delay assumption of certain forks connecting the output of a gate to the inputs of several other gates is sufficient to implement a Turing machine, and therefore that whole class of Turing computable function [4]. Those forks are called isochronic forks.

Asynchronous circuits with this delay assumption of isochronic forks are called Quasi-Delay-Insensitive (QDI) circuits. QDI can be used as the basis for asynchronous logic. All other forms of the technology can be viewed as a transformation from a QDI approach to other by adding some delay assumptions. An asynchronous circuit in which all forks are assumed isochronic corresponds to what has been called a speed independent circuit, it is a circuit in which the delays of interconnects are negligible compared to the delays in the gates. This concept of speed independent circuit was introduced by Muller [5]. In synchronous design, information propagates with each clock edge in a steady lock step fashion, whether it is needed or not. In asynchronous design, the information propagates through the circuit only when and where it is required.

Asynchronous circuits communicate via handshakes protocol. A handshake consists of a series of signal events sent back and forth between the communicating processes. We can divide the communicating elements into a sender and a receiver. The sender is that element who initiates the handshake sequence. If the sender wants the receiver to perform a certain task, it will make a request to the receiver. When the receiver has finished executing the task it makes an acknowledgement to the sender that the task has been completed. By this Handshake Communication, the sequencing of actions is handled in asynchronous circuits.

There are two type of handshake signaling - Two Phase Signaling or Transition Signaling Protocol and Four Phase Signaling or Level Signaling Protocol.

SoCs are complex and distributed systems where a large number of parallel components communicate with each other and synchronize their activities by message exchange. Synchronous (clocked) logic brings a simple solution to the concurrency issues by partially ordering transitions with respect to a succession of clock signals so as to order conflicting read/write actions.

While in the absence of a global time reference, asynchronous logic has to deal with concurrency by the methods and notations of computing concurrent process. Communicating Hardware Processes (CHP) is one of the high-level languages used in SoC design for distributed computing [6].

The systematic design of SoC is a process of successive refinements taking the design from a high level description to a transistor level netlist. Those three levels of representation CHP, HSE (Handshake Expansion), PRS (Production Rule Set) are the three main stages of the refinement process.

IV. ASYNCHRONOUS DESIGN ADVANTAGES

A variety of advantages have been demonstrated by asynchronous circuit design [7], including both Quasi Delay Insensitive (QDI) circuits and less pure forms of asynchronous circuitry which use timing constraints for higher performance and lower power and area:

- Robust handling of Metastability of arbiters.
- Higher performance functional units, which provide average-case completion rather than worst-case completion.
- Lower power consumption because no transistor ever transit unless it’s required to perform useful computation. Also, clock drivers are not required which can significantly reduce power consumption.
- Flexible pipelines, which achieve high performance while gracefully handling Variable input and output rates and mismatched pipeline stage delays
- Highly Modular and easier global timing issues
- No clock skew at all
- No need for wide spread clock distribution network.
- Semiconductor devices are becoming susceptible to Soft errors caused by Alpha Particles, Cosmic Rays and other radiation sources as they shrink to nano-scale. Quasi delay insensitive (QDI) asynchronous circuits have a strong potential against soft error [8].

V. PROBLEMS WITH ASYNCHRONOUS DESIGNS

Though asynchronous circuits do not suffer from many of the problems found in synchronous circuits, they do have some problems of their own. Most important of them are:

Hazards: Since asynchronous circuits completely rely on events on wires to communicate and sequence their order of operation, they are susceptible to glitches and hazards. Therefore special care must be taken while synthesis to eliminate the possibility of Hazards. However, the resulting circuit is still sensitive to glitches caused by ground bounces, noise etc.

Handshake Latency: Due to their way of communicating via handshakes signaling, asynchronous circuits have a handshake overhead that reduces timely performance. This penalty can be reduced by placing communicating elements close to each other during placement and routing.

Various Design Methodologies: There exist a wide variety of asynchronous design approached and methodologies. Unfortunately, this results in inconsistent specification and implementation styles, making it difficult to make fair comparisons between two systems.

Immature Synthesis Methodologies and CAD tools: For asynchronous design to be accepted as a viable solution by synchronous designers and industry there is need for mature synthesis methodologies and tools. Unfortunately many proposed methods are still in their early stages of research and have not yet been demonstrated on large industrial designs.

Despite these problems asynchronous design style is a viable complement to synchronous methodology. It is especially useful for applications requiring low latency operations and applications that can take advantage of average case delay instead of worst case delay. Low power applications are also an area where asynchronous circuits have an advantage. Many designs have been effectively implemented as asynchronous circuits, yielding better performance or power efficiency than their synchronous solutions.

VI. CONCLUSION

Asynchronous circuits have certain characteristics that differ significantly from those of synchronous circuits and, it is possible to exploit these characteristics to design circuits with very interesting performance parameters in terms of lower power consumption, better performance, smaller electromagnetic emissions (EMI), etc. All these advantages make them perfect for applications such as computer networks, mobile phones, smart cards and embedded medical devices.

Asynchronous design is not yet a well-established and widely-accepted design methodology. There is a tremendous amount of research opportunities for creating new Asynchronous design methodology, efficient testing & verification style and for CAD tool development too. All this will gradually lead to integrate "islands" of clockless logic into future generations of Asynchronous VLSI Designs.

REFERENCES

[1] Martin, A. J. “25 Years Ago: The First Asynchronous Microprocessor” at Department of Computer Science, California Institute of Technology.

[2] Nystrom Mika, Martin, A. J. “Asynchronous Techniques for System-on-Chip Design”. Proceedings of the IEEE, Vol. 0018-9219/$20.00 _2006 IEEE 94, No. 6, June 2006

[3] Alain J. Martin, Mika Nyström, and Catherine G. Wong. “Three Generations of Asynchronous Microprocessors”. California Institute of Technology. Published by the IEEE CS and the IEEE CASS November–December 2003.

[4] Manohar, Rajit, Martin, A. J. “Quasi delay insensitive circuits are Turing complete. Department of Computer Science, California Institute of Technology November 17,1995.

[5] Muller, D. E. and Bartky, W. S.. “A theory of asynchronous circuits”. Proceedings of International Symposium on Theory of Switching, 1959, pp. 204–243.

[6] C. A. R. Hoare, “Communicating sequential processes”. Commun. ACM, vol. 21

[7] Babak Rahbaran and Andreas Steininger, “Is Asynchronous Logic More Robust Than Synchronous Logic?”, IEEE Transactions on Dependable and Secure Computing, Vol. 6, No. 4, October-December 2009

[8] Jayanthi, D., Rajaram, M. “The Design of High Performance Asynchronous Pipelines with Quasi Delay Insensitive”. International Journal of Computer Applications (0975 – 8887) Volume 52– No.16, August 2012.