

Asynchronous Digital Design in Nanometer CMOS Microelectronics Education

Ndubuisi Ekekwe¹, Chinyeaka Ekenedu²

¹ Dept. of Electrical & Computer Engineering, Johns Hopkins University, 3400 N. Charles Street, Baltimore MD 21218, USA; nekekwe1@jhu.edu

² Dept. of Electrical Engineering, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden; ccek05@bth.se

ABSTRACT

The invention of complementary metal oxide semiconductor (CMOS) technology has revolutionized the modern industry. But as it scales into the nanometer regime; it faces numerous challenges on performance and reliability owing to increased interconnect noise, power dissipation etc that partly emanate from its synchronous sequencing. Though possessing inherent drawbacks, asynchronous or self timed circuit offers the prospect of overcoming some of these challenges as it uses locally generated handshakes for synchronization. This design paradigm is presently offered in few schools with limited awareness among undergraduate students. Towards new design innovation in computing as well as managing the limiting challenges of synchronous nanometer CMOS, asynchronous digital design should form key part of undergraduate microelectronic education.

1. INTRODUCTION

The invention of CMOS integrated circuit is a major milestone in the history of modern industry and commerce. It has transformed medicine, aerospace, information and communication technologies among others as a result of its low cost, performance and seamless integration. Over the decades, CMOS systems have been developed and synchronously sequenced using globally distributed periodic timing signals called *clocks*. To sustain Moore's law, the number of transistors in a CMOS chip doubles about every two years, circuit complexity and transistor downscaling have increased (see Figure 1) [1]. These factors have resulted to reliability and performance problems in forms of signal integrity and power dissipation thereby challenging the future of CMOS technologies [2].

Asynchronous or self timed circuits sequenced using locally generated handshakes (*request* and *acknowledgement* signals), without clocks, for synchronization enjoy some advantages over the synchronous counterparts (Figures 2 & 3) [3, 4]:

- It consumes lesser power since signal transitions occur only in areas involved in current computation
- Average case performance as system could sense when computation has ended
- Design flexibility and cost reduction, with higher level logic separated from lower timing design
- Elimination of clock skew since there is no clock to distribute
- Separation of timing from functional correctness thereby enabling insensitivity to delay variance in layout, fabrication and operating environments

On the other hand, asynchronous circuits have the following drawbacks compared to synchronous counterparts:

- Larger in size for similar functional designs with potential to higher cost.

- As operations are pre-ordered, its design is difficult
- Very complex testing procedure
- Lack of robust CAD tools

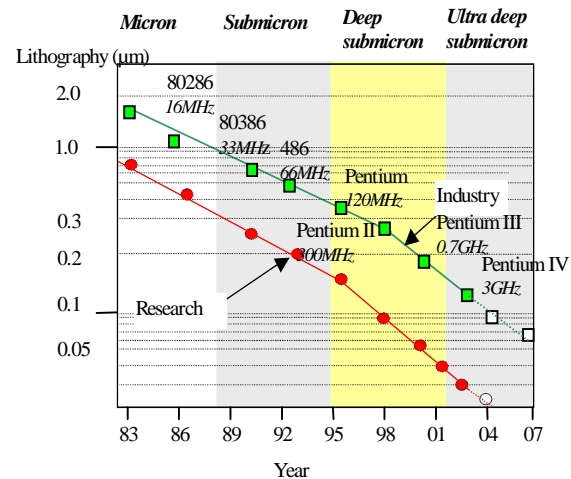


Figure 1 Evolution of lithography [1]

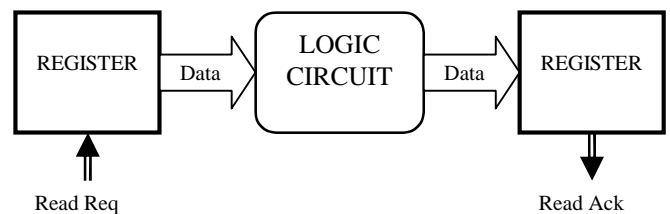


Figure 2 Asynchronous data transfer

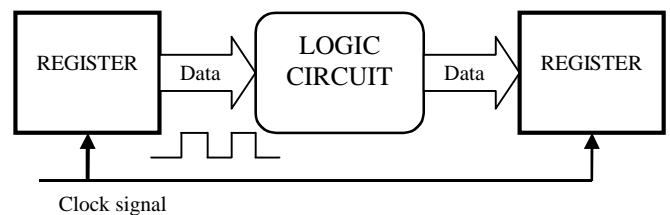


Figure 3 Synchronous data transfer

Despite these shortcomings, asynchronous circuits offer a bold architectural innovation to develop CMOS systems with good performance and reliability in the nanometer regime where synchronous counterparts experience massive setbacks. Considering this potential and its poor awareness among students, it is very vital that schools adopt and integrate it into their microelectronics curricula.

2. ASYNCHRONOUS DIGITAL DESIGN FLOW

Asynchronous design is different from synchronous design since no clock is used. Nonetheless, it follows the

established CMOS design flow, which involves in order: *system specification, system design, circuit design, layout, verification, fabrication and testing* though with major differences in concepts [3]. Based on A.J.Martin's *communicating hardware processes* (CHP) [3, 4], the major design stages radically different from synchronous development systems include:

- *CHP*- used to provide constructs that provide the specification of the circuit to be synthesized.
- *Process decomposition*- breaks the CHP program into multiple parts.
- *Handshaking expansion*- this represents each communication action with operations on Boolean variables. For data transfer accuracy, the two ends of the channel have to obey some given *protocol*.
- *Production rule expansion*- provides simplified description of the CMOS transistor behavior producing the *production rule set (PRS)*, which is the canonical representation of the digital circuit
- *CMOS implementation*. This is the hardware realization based on the PRS.

To illustrate the design concepts, an asynchronous buffer is developed. Beginning from the CHP given by $*[L?x;R!x]$, the schematic (Figure 4) and layout (Figure 5) where L, R and x are the sender, receiver and local variable respectively are developed. The design uses a *quasi delay insensitive two-phase handshaking protocol* with r_o, r_i and l_i and l_o as decomposed variables and a state variable (k). The *bubbled reshuffled* and *CMOS implementable PRS* obtained after many transformations and used to realize the circuit of Figure 4 is presented below [3]. This circuit though complex can withstand variations in temperature and voltage than the synchronous counterpart. It is variation tolerant by design.

$$\begin{aligned}
 l_i \wedge k \wedge r_o &\rightarrow l_o \downarrow \\
 \neg l_o &\rightarrow l_o \uparrow \\
 l_o \wedge l_i &\rightarrow k \downarrow \\
 l_i &\rightarrow l_i \downarrow \\
 \neg l_i &\rightarrow l_i \uparrow \\
 r_i &\rightarrow r_i \downarrow \\
 l_o &\rightarrow l_o \downarrow \\
 \neg l_i \wedge \neg k &\rightarrow l_o \uparrow
 \end{aligned}$$

$$\begin{aligned}
 \neg r_i \wedge \neg k \wedge \neg l_o &\rightarrow \neg r_o \uparrow \\
 \neg r_o &\rightarrow r_o \uparrow \\
 \neg r_o \wedge \neg r_i &\rightarrow k \uparrow \\
 r_o &\rightarrow r_o \downarrow \\
 \neg r_i &\rightarrow r_i \uparrow \\
 r_i \wedge k &\rightarrow \neg r_o \downarrow \\
 \neg \neg r_o &\rightarrow r_o \uparrow \\
 \neg r_o &\rightarrow r_o \downarrow
 \end{aligned}$$

3. COURSE CONTENTS

The following could form contents of any asynchronous microelectronics education.

- Introduction

- Communicating Hardware Processes
- Synchronization
- Production rules
- Handshaking expansions
- Synthesis
- Reshuffling/Process decomposition
- Tools/CADS
- Control/data separation
- Arbitration/Interfaces
- Performance, optimization and timing Analysis
- Pipeline Dynamics & event-rule systems
- Project

4. CONCLUSIONS

As the quest to develop faster and dependable computing continues, asynchronous system would play key roles. Despite its present low commercial impacts and awareness among undergraduate students, it has promises in areas of reliability in both adverse environments and nanometer CMOS regime [5] and should be integrated into digital microelectronics curricula in schools.

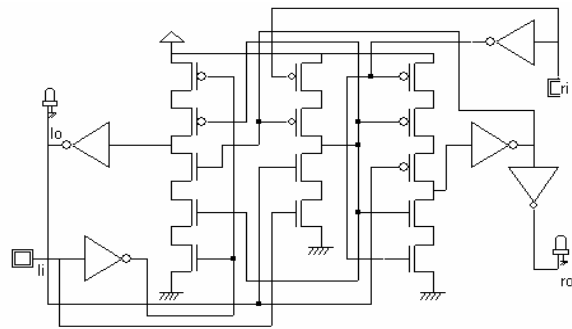


Figure 4 Asynchronous CMOS buffer

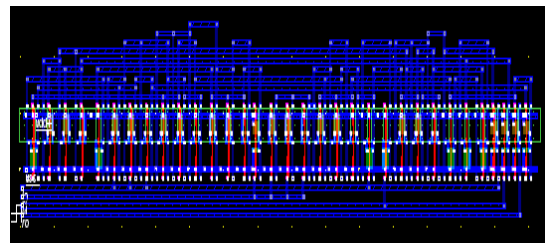


Figure 5 Asynchronous CMOS buffer layout

REFERENCES

- [1] E. Sicard, D. Bendhia, Basics of CMOS Design, Tata McGraw-Hill, New Delhi, 2005
- [2] International Technology Roadmap for Semiconductor (ITRS) 2006, [online] <http://www.itrs.net/>
- [3] N. Ekeke, Asynchronous Integrated Circuit Design, IEEE Potentials, vol.25, pp 36, 2006
- [4] A. Martin, "A program transformation approach to asynchronous VLSI design", in *Deductive Program Design*, NY, Springer 1996
- [5] N. Ekeke et al, "Power dissipation and possible control techniques in the ultra-deep submicron technologies", *Microelectronics Journal*, Vol. 37, Issue 9, Sept 2006, Pp 851-860