iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

www.ijraset.com Volume 3 Issue IV, April 2015
IC Value: 13.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Synchronization in Digital System Design

Jyoti Chandel[1], Neha Gupta[2], Jyoti Yadav[3]

Abstract- In digital system design, synchronization ensures that operations occur in the logically correct order, and is a critical factor in ensuring the correct and reliable system operation. As the physical size of a system increases, or as the speed of operation increases, synchronization plays an increasingly dominant role in the system design. The number of techniques has been developed by digital communication to deal with synchronization on a global and even cosmic scale; and as the clock speeds of chip, board, and room-sized digital systems increase, they may benefit from similar techniques. The digital communication communities and digital system have evolved synchronization techniques independently, choosing different terminology and different techniques . In this paper. we present a unified framework and terminology for synchronization design in digital systems, borrowing techniques and terminologies from both digital system and digital communication design disciplines. We then compare the throughput of synchronous and asynchronous interconnects.

I. INTRODUCTION

In digital systems operations can either proceed concurrently, or they must obey a precedence relationship. If two operations follow a precedence, then the role of synchronization is to ensure that the operations follow in the right order. Synchronization is thus a critical part of digital system design. The most common approach to synchronization is to distribute a clock signal to all modules of the system. With the scaling of feature-sizes in VLSI design, clock speeds are increasing rapidly, but increases in complexity tend to prevent significant reductions in chip size. As a consequence of this scaling, clock speeds in digital system designs are increasing in relation to propagation delays. This is causing increasing problems with the traditional synchronous design methodologies, certainly at the system and board levels, and increasing even within high performance chips . This problem will be accentuated with the more common application of optics to system interconnection. In this paper. we place the synchronization problem and design approaches in digital communication and digital system design in a common framework, and then examine opportunities for cross-fertilization between the two fields. In attaining the unification of design methodologies that we attempt in this paper. the first difficulty we face is the inconsistencies. and even contradictions, between terms as used in the two fields.

A. Abstraction In Synchronization

A basic approach in system design is to define abstrac*tions* that enable the designer to ignore unnecessary details and focus on the important features of the design. While every system is ultimately dependent on underlying physical laws, it is clear that if we relied on the solution of Maxwell's equations at every phase of the design, systems could never get very complex. Abstractions are often applied in a hierarchical fashion, where each layer of abstraction relies on the essential features of the abstraction level below, and hides unessential details from the higher level.

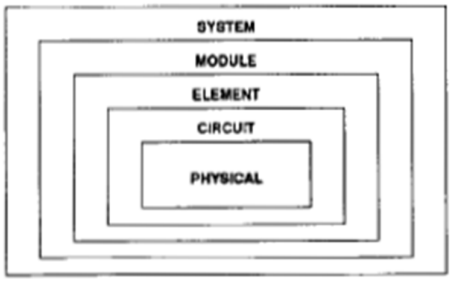


Fig1: An example of some abstractions applied to digital system design.

Some Basic Synchronization Abstractions-

1) Boolean Signals: A Boolean signal (voltage or current) is assumed to represent, at cach time, one of two possible levels. At the physical level, this signal is generated by saturating circuits and bistable memory elements. There are a couple of underlying behaviours that are deliberately ignored in this abstraction: finite rise-time and the metastable behaviour of memory elements.

The Boolean abstraction is most valid when the rise time is very much shorter than the interval between transitions, and metastability is avoided or carefully controlled. But the designer must be sure that these effects are negligible, or unreliable system operation will result.

2) Signal Transitions: For purposes of synchronization we are often less concerned with the signal level than with the times at which the signal changes state. In digital systems this time of change would be called an edge or transition of the signal. The notion of the transition time ignores rise-time effects. In fact, the transition time is subject to interpretation. Transition times are a useful abstraction for the case where the rise times are very short in relation to the interval between transitions, with the result that the variation in the transition time is negligibly small over the set of all possible definitions of the transition time. Rise-time is governed by underlying physical phenomena, such as transmission line dispersion, and can be decreased by using wider bandwidth drivers or intermediate repeaters. As system clock rates increase, however. for a given interconnect style the behaviour ignored by this abstraction inevitably becomes essential.

3) Phase and Frequency: For a Boolean signal, we can define a phase and frequency of the signal as the phase and frequency of the associated clock. It is convenient to describe mathematically a clock signal with uniformly spaced transitions as
x([) = p ((j + 4) modulo I) ,
where p (t) is a 50%′ duty cycle pulse

$$p(t) = \begin{cases} 1. & 0 \le t < 0.5 \\ 0. & 0.5 \le t \le 1. \end{cases}$$

f is the nominal frequency, and ϕ is the phase. As ϕ varies over the range $0 \le \phi < 1$, the transitions are shifted in time over one cycle. The phase is thus expressed as the fraction of a cycle. When we have two Boolean signals, the relative phase can be expressed as the phase difference $(\phi_1 - \phi_2)$ between their respective associated clocks.

4) Synchronism: Having carefully defined some terms, we can now define some terminology related to the *synchronization* of two signals. A taxonomy of synchronization possibilities is indicated. As previously mentioned, a Boolean signal can be either isochronous or anisochronous. Given two data signals, if
both are isochronous, the frequency offsets are the same, and the instantaneous phase difference is zero.

$$\Delta\phi(t) = 0$$

then they are said to be synchronous (from the Greek “syn” for “together”). Common examples would be a Boolean signal that is synchronous with its associated clock (by definition), or two signals slaved to the same clock at their point of generation. Any two signals that are not synchronous are *asyn*chronous (or “not together”). Some people would relax the definition of synchronous signals to allow a nonzero phase difference that is constant and known.

B. Additional Timing Abstractions in Digital Systems

This section has covered some general considerations in the modeling of Boolean signals from a synchronization perspective. In this subsection we describe a couple of additional abstractions that are sometimes applied in digital system design, and represent simplifications due to the relatively small physical size of such systems.

1) Equipotential Region: It is often assumed at the level of the element abstraction that the signal is identical at all points along a given wire. The largest region for which this is true is called (by Seitz 161) the equipotential region. Like our other models, this is never strictly valid, but is useful if the real time it takes to equalize the potential along a wire is small in relation to other aspects of the signals, such as the associated clock period or rise-time. The equipotential region is a useful concept in digital system design because of the relatively small size of such systems.

2) Ordering of Signals: In the design of digital systems, it is often true that one Boolean signal is slaved to another, so that at the point of generation the one signal transitions can always be guaranteed to precede the other. Conversely, the correct operation of circuits is often dependent on the correct ordering of signal transitions, and quantitative measures such as the minimum time between transitions. One of the main reasons for defining the equipotential region is that if a given pair of signals obey an ordering condition at one point in a system, then that ordering will be guaranteed anywhere within the equipotential region.

III. SYNCHRONIZATION

The role of synchronization is to coordinate the operation of a digital system. In section III-A and B**,** we review two traditional approaches to synchronization in digital system design: synchronous and anisochronous interconnection. In section III-C, we briefly describe how synchronization is accomplished in digital communication systems.

A. Synchronous Interconnection

As shown in Figure, each element (or perhaps module) is provided a clock, as well as one or more signals that were generated with transitions slaved to the clock. The common clock controls the order of operations, ensuring correct and reliable operation of the system.

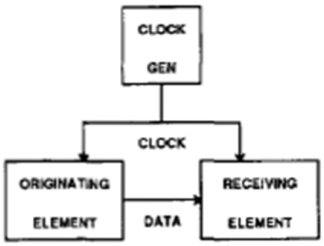


Figure:- Synchronous interconnection in which a common clock is used to synchronize computational elements.

B. Anisochronous Interconnect

The synchronous interconnect approach uses isochronous signals throughout the system, since all signals are slaved to an isochronous clock. A popular alternative to synchronous interconnect has been to abandon the isochronous assumption, and further abandon the use of a global clock signal altogether. Rather, the elements of the system are chosen to fall within an equipotential region, and the interconnection between elements is designed to operate in a delay-insensitive manner: that is, operate reliably regardless of what the delays are. This is accomplished by having each element of the system generate a completion signal which has a transition coincident with the settling time of that element. The completion signal is a sort of locally generated clock, and is used to synchronize the different elements. Since the completion signal depends on the settling time, which can be data-dependent, the resulting signals are anisochronous.

C. Synchronization in Digital Communication

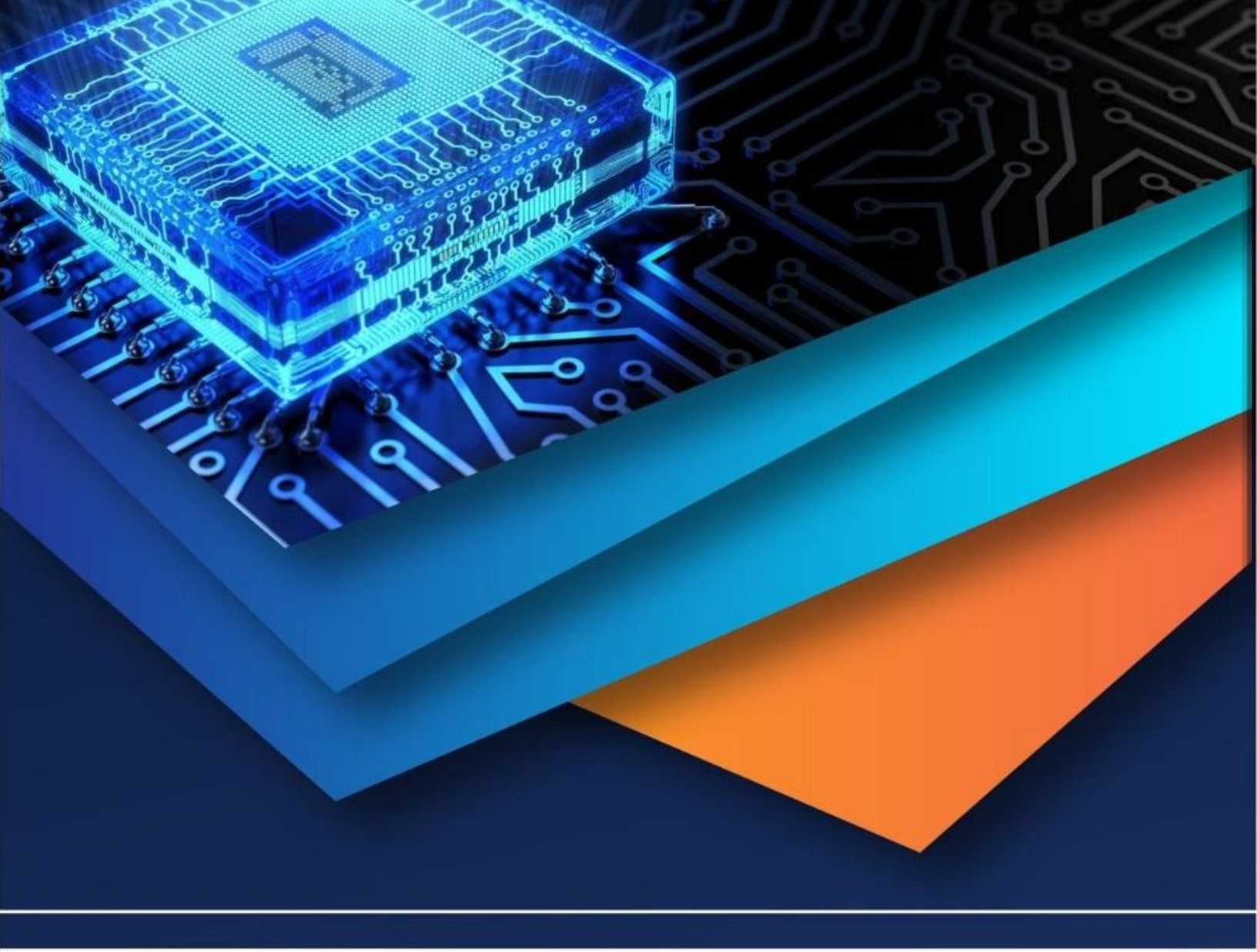
In digital communication, the interconnect delays are very large, so that alternative synchronization techniques are required . These approaches are all isochronous, implying that the signals are all slaved to clocks, but differ as to whether a common clock distributed toe ach node of the network is used (mesochronous) or independent clocks are used at the nodes (plesiochronous and heterochronous). They also share a common disadvantage relative to synchronous and anisochronous interconnect-the inevitability of metastable behaviour. Thus, they all have to be designed carefully with metastability in mind, keeping the probability of that condition at an acceptable level.

IV. CONCLUSION

In this paper we have attempted to place the comparison of digital system synchronization on a firm theoretical foundation, and compare the fundamental limitations of the synchronous, anisochronous, and mesochronous approaches. A firm conclusion is that interconnect delays place a fundamcntalimitation on the communication throughput for anisochronous interconnect (equal to the reciprocal of two or four delays), this limitation does not exist for mesochronous interconnect. Further, anisochronous requires extra interconnect
wires and completion signal generation.

REFERENCES

[1] M. Hardmian, L. A. Homak, T. E. Little, S. K. Tewkshury, and P.Franzon, “Fundamental interconnection issues,” AT&T Tech. J., vel. 66, p. 134. July 1987.
[2] G. Messerschmitt, “Digital communication in VLSI design,“ In Proc. 23rd Asilomar Conf. Signals, Syst. Comput.. Oct. 1989.
[3] De Micheli, and M. Flynn. “Designing high-performance digital circuits using wave pipelining,” lEEE ICCAD-X9 Dig. Tech. Papers, Nov. 1989.
[4] P. Bassett, “A high-speed asynchronous communication technique for MOS VLSI systems.” Mass. Inst. Technol., Dec. 1985.
[5] M. Hatamlan and G. L Cash, “Parallel bit-level pipelined VLSI designs for high-speed signal processing,” Proc. IEEE. vol. 75, p. 1192, Sept. 1987.
[6] Mead and L. Conway, Introduction to VLSI Systems. Reading, MA: Addison-Wesley, 1980.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY