2011 International Conference on Information and Network Technology
IPCSIT vol.4 (2011) © (2011) IACSIT Press, Singapore

Synchronization In Digital Systems

Ranjani .M. Narasimhamurthy

Lecturer, Dr. Ambedkar Institute of Technology,
Department of Telecommunication Engineering, Bangalore.

Abstract. In digital system design, the methods of synchronous & asynchronous designing have their own merits and de-merits. Designers adopt the appropriate method based on the requirements like power efficiency, modularity, performance, clock skew, concurrency, metastability, testability, design tools support, automatic adaptation to physical properties and many more. The "need for testability" is the driving factor to select the method of synchronous design. This helps to achieve comparatively better testability than the design based on asynchronous design. The synchronous design method was applied for a bus control unit to control heater switches from both the serial and I/O ports. The coding may be performed using VHDL and synchronous testing be achieved using the Modelsim and synthesis using the Synopsys tools.

1. Introduction

Synchronization involves determining or enforcing and ordering of events on signals. 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 the system increases, or as the speed of operation increases, or issue of testability arises then synchronization plays an important role in the system design. Digital abstraction depends on all signals in a system having a valid logic state. Hence digital abstraction depends on reliable synchronization of external events. Synchronization involves imposing or recognizing an ordering of events on signal lines. An asynchronous signal, which may change at any time, must be synchronized with a clock before it can be used as an input to synchronous logic [1].

The first section details a few points on the disadvantages of asynchronous design & second section outlines some uses of synchronization. The third section details on the mechanism of synchronization. The fourth section gives the details on synchronization and the fifth section concludes with the process of synchronization as applied to the heater switching unit followed by acknowledgement and references.

2. Obstacles in Asynchronous Design

- **Testability**: Although one can argue that power consumption, performance, EMI, of asynchronous signals are very real, so are the threats. The most frequently cited argument against asynchronous design is the difficulty of production testing asynchronous designs. Much less is known about the testability of such circuits, and design margin characterization poses a significant problem. Production test equipment is heavily committed to using a clock to regulate the application of stimuli, and to time the sampling of responses from the device under test. Timing driven test equipment is particularly not suitable for testing time driven asynchronous circuits. In addition, asynchronous circuits tend to require logically redundant gates to remove hazards, and these are un-testable by normal methods. Economically testability is an absolute requirement for volume production.
- **Design Tool Support**: Design tool support is another area of difficulty. Modern CAD tools are aimed at clocked circuits, and though it is clearly possible to design a clockless circuit using them, they could be lot more helpful, if the tools were targeted for clockless designs. For example most synthesis

+ Author: Ranjani.M.N. Tel.: +91-9844041877; fax: +91-80-23217789
E-mail address: ranjanimn@rediffmail.com

tools will automatically remove logically redundant gates that were carefully inserted to remove hazards [2]

- **Design Practice**: The biggest obstacle to the acceptance of asynchronous design techniques assuming their merits can be demonstrated irrefutably is the aversion of the design community. Engineers have been trained to avoid asynchronous operation, though it was used in the distant past. This causes that the designers will be reluctant to suggestions to revert back to asynchronous design methodologies that were rejected a few decades ago.

3. Uses of Synchronization

Operations in a digital system can either proceed concurrently, or they must obey a precedence relationship. If two operations obey precedence then the role of synchronization is to ensure that operations follow the correct order. Synchronization is thus a critical part of digital system design.

There are three common uses of synchronization

- Arbiting between asynchronous requests
- Sampling asynchronous signals with a clock
- Transmitting synchronous signals between two clock domains

The most common approach to synchronization is to distribute a clock signal to all modules of the system. With the scaling of feature size in VLSI design, clock speeds are increasing rapidly, but increases in the complexity tend to prevent significant reductions in chip size. As a consequence of this scaling, clock speeds in digital system design are increasing in relation to propagation delays. However because of the wide difference between system physical sizes in relation to clock speeds, the design styles of the communities have developed almost independently. [1]. Also the impact of accessibility on testing leads to the observation that asynchronous design or those with unconstrained timing signals are much more difficult to test than synchronous designs that have easily accessible clock generation and distribution circuits. [4].

3.1. Abstractions in Synchronization

Abstractions are often applied in a hierarchical fashion, where each layer of abstraction relies on the essential features of abstraction levels below and hides unessential details from the higher level

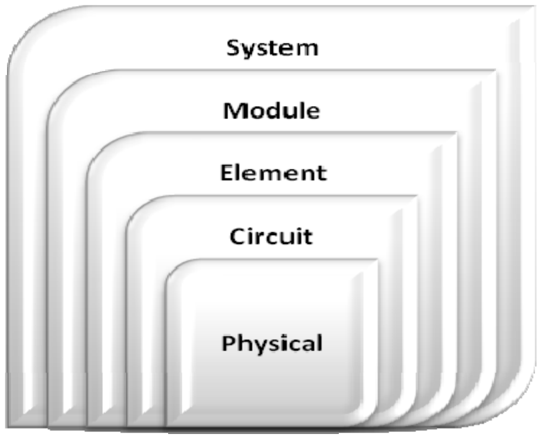


Fig. 1: Abstraction applied to digital system design

At the base of the design we have the physical representation where we have semiconductor materials, interconnect metallization. Above this is the circuit abstraction, where we have transistors, interconnections. Next is the element like flip flops and gates. Here we describe elements in terms that deals with their operational characteristics.(timing diagrams, Boolean functions etc.,). Next is module where we group elements to form more complex entities like memories, register files, arithmetic logic units etc.,

While abstractions are very useful and in fact absolutely necessary, they should be applied with care. The essence of an abstraction is that we are ignoring some details of the underlying behaviour, which we hope are irrelevant to the operation. It should always be verified that in fact the characteristics being ignored are in fact irrelevant and with considerable margin, or else the final system may turn out to be inoperative or unreliable. This is especially true of synchronization, which is one of the most frequent causes of unreliable operation of the system. [1].

3.1.1. Synchronism

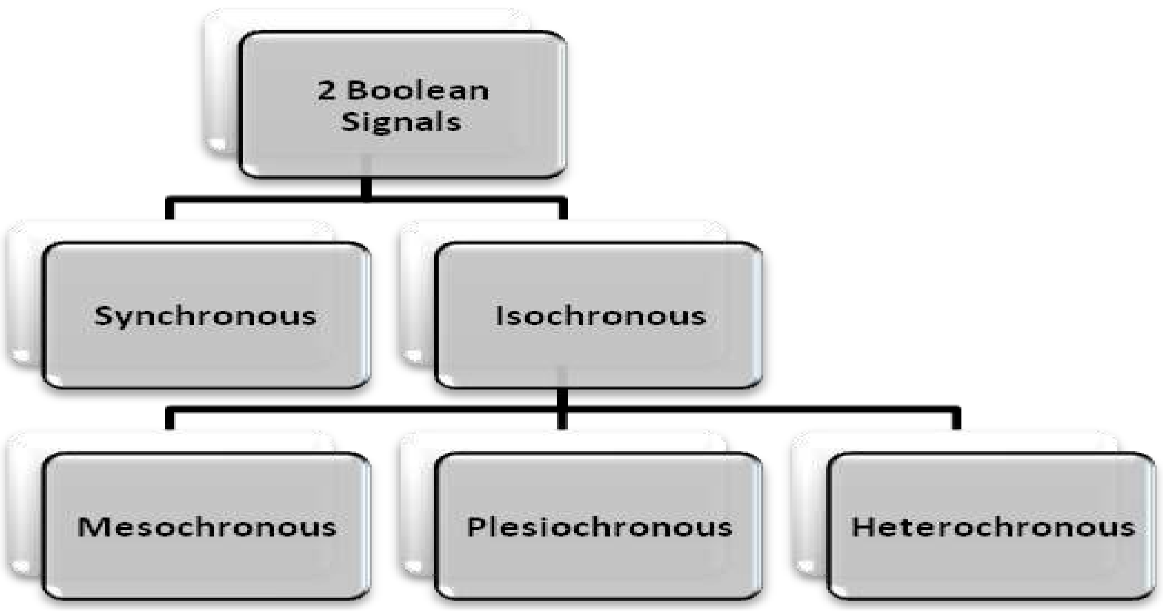


Fig. 2: Taxonomy of synchronization possibilities

Given two data signals if both are isochronous, frequency offsets are the same, and the instantaneous phase difference is zero, then they are said to be synchronous. If two isochronous signals have exactly the same average frequency f + Df, then they are called mesochronous. Ex: two signals generated from the same clock, but suffering intermediate interconnect delays relative to one another are mesochronous. Two signals that have average frequencies nominally same but not exactly same are plesio (near) chronous. Two signals that have average frequencies that are nominally different average frequencies are called heterochronous.

4. Synchronization

Role of synchronization is to coordinate the operations of the digital system. Here each element 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.

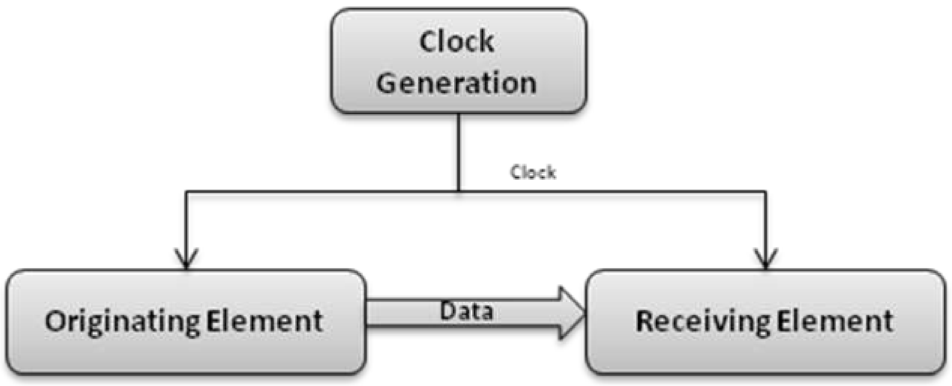


Fig. 3: Synchronous Interconnection

In Fig. 3, a common clock is used to synchronize computational elements. Now we will review the synchronous interconnection which is used as the basis for the synchronous design of the heater switching module

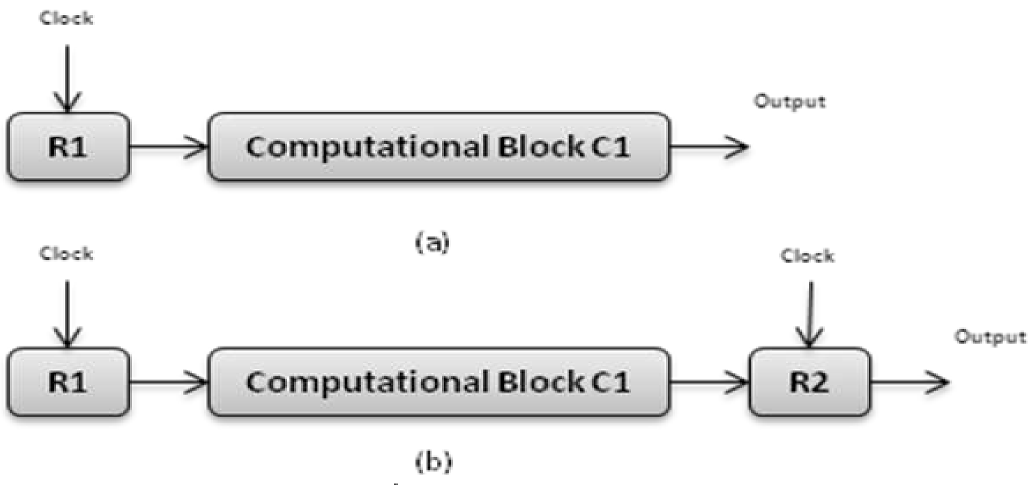


Fig. 4: Principle of synchronous interconnection

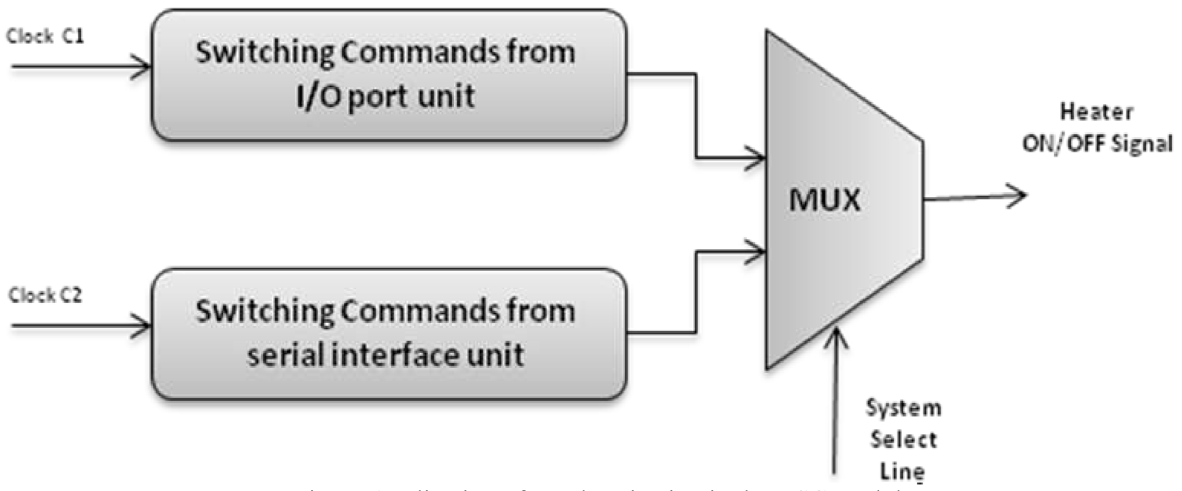
The fundamental principle of synchronous design is as shown in the Fig. 4. In Fig. 4(a) a computational block C1 is connected to a synchronizing register R1 at its input. This register is clocked using the positive transition of a periodic clock signal. The assumption is that output signal of register changes synchronously with the positive transition of the clock. The computational block performs the same computation repeatedly on new input signals being applied at each clock transition. The purpose of R1 is to control the time at which the computational block begins to perform its work, in order to synchronize it to other computational clocks in the system. This is an edge triggered model.

In Fig. 4(b) the synchronous interconnect isolates the system behavior from these realities of the circuit implementation by setting the clock period T so that there is a certainty period during which the output signal is guaranteed to be correct and then samples the output signal again. This interconnect is called synchronous because the proper phase of clock to use at R2 must be known in advance. With synchronous interconnect the undesired behavior is hidden behind R2. (Output of R2).We can then abstract the operation of the computational block as viewed from the output of R2 as an element that completes computation at the positive transition of the second clock and this makes it to synchronize the block with the other since the uncertainty period has been eliminated.

5. Conclusion

In the switching unit we can bring about the same analogy as in the Fig. 5. We have computational blocks (a) I/O port units & (b) Serial Interface Unit. The synchronous interconnect in each block isolates the system behavior from the throughput and latency of the circuit implementation by setting a clock period T=0.0125ms, so that there is a certainty period during which the heater ON/OFF commands are found to be present due to either of the blocks.

Each computational block is associated with a set of registers for latching heater commands. One set of these commands are selected by the Multiplexer (another computational block) at the known certainty period. Hence the system is synchronized with respect to system clock.


Clock C1
Switching Commands from I/O port unit
Clock C2
Switching Commands from serial interface unit
MUX
Heater ON/OFF Signal
System Select Line
Fig. 5: Application of synchronization in the HSC module

Once the uncertainty period is eliminated the occurrence of events is known and the testability of such circuit becomes easier [1]. Therefore we can conclude that the method of synchronous designing is thus achieved by designing synchronizer circuits as per the requirements for the need of synchronization also in future complex designs will remain in synchronous design method as it supports a consistent approach. [3].

6. Acknowledgements

I thank Dr Jambunath Baligar, Asst. Prof, Dept of Electronics & Communication, Dr. AIT, Bangalore, for his timely guidance and support.

7. References

[1] David G. Messerschmitt, "*Synchronization in digital system design*", in IEEE journal on selected areas in communication. Vol. 8. No. 8. October 1990.

[2] Report on "*Asynchronous design*" ASYNC2000.

[3] ACid-WG trip report on ISSCC97 panel discussion.

[4] Miron Abramovici, Melvin A. Breuer, Arthur .D. Friedman, Chapter 9, "*Digital System Testing and Testable Design*", Computer Science Press.

[5] William J. Dally and John W. Poulton "*Digital systems engineering*", Cambridge University Press- 1998.

[6] Ran Ginosar, "*Fourteen Ways To Fool Your Synchronizer*", ASYNC 2003.