

ASYNCHRONOUS INTEGRATED CIRCUITS DESIGN

Many modern integrated circuits are sequenced based on globally distributed periodic timing signals called clocks. This method of sequencing, synchronous, is prevalent and has contributed to the remarkable advancements in the semiconductor industry in form of chip density and speed in the last decades. For the trend to continue as proposed in Moore's law, the number of transistors on a chip doubles about every two years, there are increasing requirements for enormous circuit complexity and transistor downscaling.

As the industry pursues these factors, many problems associated with switching delay, complexity management and clock distribution have placed limitation on the performance of synchronous system with an acceptable level of reliability. Consequently, the synchronous system design is challenged on foreseeable progress in device technology.

These concerns and other factors have caused resurgence in interest in the design of asynchronous or self-timed circuits that achieve sequencing without global clocks. Instead, synchronization among circuit elements is achieved through local handshakes based on generation and detection of request and acknowledgement signals.

Some notable advantages of asynchronous circuits over their synchronous counterparts are presented below:

- ♦ Average case performance. Synchronous circuits have to wait until all possible computations have completed before producing the results, thereby yielding the worst-case performance. In the asynchronous circuits, the system senses when computation has completed thereby enabling average case performance. For circuits like ripple carry adders with significantly worst-case delay than average-case delay, this can be an enormous saving in time.
- ♦ Design flexibility and cost reduction, with higher level logic design separated from lower timing design
- ♦ Separation of timing from functional correctness in certain types of asynchronous design styles thereby enabling insensitivity to delay

variance in layout design, fabrication process, and operating environments.

- ♦ The asynchronous circuits consume less power than synchronous since signal transitions occur only in areas involved in current computation.
- ♦ The problem of clock skew evident in synchronous circuit is eliminated in the asynchronous circuit since there is no global clock to distribute. The clock skew, difference in arrival times of clock signal at different parts of the circuit, is one of the major problems in the synchronous design as feature size of transistors continues to decrease.

Asynchronous circuit design is not entirely new in theory and practice. It has been studied since the early 1940's when the focus was mainly on mechanical relays and vacuum tube technologies. These studies resulted to two major theoretical models (Huffman and Muller models) in the 1950's. Since then, the field of asynchronous circuits went through a number of high interest cycles with a huge amount of work accumulated. However, problems of switching hazards and ordering of operations encountered in early complex asynchronous circuits resulted to its replacement by synchronous circuits. Since then, the synchronous design has emerged as the prevalent design style with nearly all the third (and subsequent) generation computers based on synchronous system timing.

Despite the present unpopularity of the asynchronous circuits in the mainstream commercial chip production and some problems noted above, asynchronous design is an important research area. It promises at least with the combination of synchronous circuits to drive the next generation chip architecture that would achieve highly dependable, ultrahigh-performance computing in the 21st century.

The design of the asynchronous circuit follows the established hardware design flow, which involves in order: *system specification*, *system design*, *circuit design*, *layout*, *verification*, *fabrication* and *testing* though with major differences in concept. A notable one is the impractical nature of designing an asynchronous system based on ad-hoc fashion. With the use of clocks as in synchronous systems, lesser emphasis is placed on the dynamic state of the circuit whereas the asynchronous designer has to worry over hazard and ordering of operations. This makes it impossible to use the same design techniques applied in synchronous design to asynchronous design.

The design of asynchronous circuit begins with some assumption about gate and wire delay. It is very important that the chip designer examines and validates the assumption for the device technology, the fabrication process, and the operating environment that may impact on the system's delay distribution throughout its lifetime. Based on this delay assumption, many theoretical models of asynchronous circuits have been identified.

There is the delay-insensitive model in which the correct operation of a circuit is independent of the delays in gates and in the wires connecting the gate, assuming that the delays are finite and positive. The speed-independent model developed by D.E. Muller assumes that gate delays are finite but unbounded, while there is no delay in wires. Another is the Huffman model, which assumes that the gate and wire delays are bounded and the upper bound is known.

For many practical circuit designs, these models are limited. For the examples in this discussion, *quasi delay insensitive (QDI)*, which is a combination of the delay insensitive assumption and isochronic-fork assumption, is used. The latter is an assumption that the relative delay between two wires is less than the delay through a sequence of gates. It assumes that gates have arbitrary delay, and only makes relative timing assumptions on the propagation delay of some signals that fan-out to multiple gates.

Over the years, researchers have developed a method for the synthesis of asynchronous circuits whose correct functioning do not depend on the delays of gates and which permitted multiple concurrent switching signals. The VLSI computations are modeled using Communicating Hardware Processes (CHP) programs that describe their behavior algorithmically. The QDI circuits are synthesized from these programs using semantics-preserving transformations.

In conclusion, as the trend continues to build highly dependable, ultrahigh-performance computing in the 21st century, the asynchronous design promises to play a dominant role.

Source : <http://goafrit.wordpress.com/2010/05/25/asynchronous-integrated-circuits-design/>