THE ASYNCHRONOUS DESIGN IMPLEMENTATION PROCESS

An asynchronous circuit is a circuit which is not governed by a global clock, but uses hand shaking communication protocol to communicate. It is largely an autonomous circuit and is delay insensitive.

Under a digital design paradigm, the design of asynchronous systems can be concisely described using high level languages. A programming notation,*Communicating Hardware Processes (CHP)*, is used to provide a basic set of constructs that provides the specification of the circuit to be synthesized. The core of this language is a sequential programming notation based on E.W. Dijkstra's language of guarded commands and C.A.R Hoare's *Communicating Sequential Process (CSP)*.

The language is distinguishable from many programmable languages in that it includes a notion of *non-determinism* as parts of its features. Non-determinism means that a program can have more than one output for the same sets of inputs. The CHP consists of a process or more, which operates in parallel and communicate with each other through *channels*. A channel connects two processes and the two ends of a channel are referred to as *ports*.

The next stage is called the *process decomposition*, which breaks up the CHP program into multiple parts and extracts, if possible, common program parts. Compiling smaller processes facilitates the rest of the synthesis procedure and sharing common program parts reduces the area of the final circuit. In addition, this process involves the separation of control and datapath, and enables temporary removal of the datapath and compilation of the control section.

The next stage is called *handshaking expansion (HSE)*, which 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.* The most prevalent ones are *two-phase handshaking* and *four-phase handshaking* protocols described later.

Usually, the HSE is optimized for concurrency while minimizing circuit areas of the implementation in another process called *reshuffling*. Besides, if there are states in a reshuffled HSE that cannot be distinguished, state variables are introduced to differentiate them. This process is known as *state variable insertion*.

The next stage involves a very simplified description of the CMOS transistor behavior called the *production rule expansion*. The result of this process produces what is called the *production rule set*, which could be considered to be a canonical representation of the digital circuit. This representation can be decomposed into several equivalent networks of digital operators based on the sets of building blocks used or the technology (Si or GaAs).

It is during this stage that issues, which include stability and non-interference, are examined carefully in the design. While stability ensures that circuit does not have hazards, non-interference will guarantee that the resulting CMOS circuit has no stable states whenever there is a short in the circuit. One way of achieving this is by *guard strengthening;* a process that reduces the number of states the production rules can fire thereby preventing incorrect *effective firing*. *Guards* are Boolean valued expressions, which could either be true or false.

A firing is said to be an *effective firing* when it changes the state of computation in a system; otherwise, it is called a *vacuous firing*. Moreover, to put the circuit in the proper state upon power up, a reset signal is added to the PRS. The *production rule expansion* is the final target of synthesis whose results are used for CMOS implementation (the hardware realization). A simplified design flow of an asynchronous QDI circuit involves the following stages:

- CHP specification
- Process Decomposition
- Handshaking Expansion
- Production Rule Exp.
- CMOS Implementation

Asynchronous circuits are promising. And the challenges in the deep-submicron technologies in form of electron thermal energy, tunneling leakage current that degrade performance and expend power consumption are potential issues that call for attention to asynchronous systems design. Though the development of the new generation QDI asynchronous systems has not made enormous commercial impacts, enormous ongoing projects project a bright future.

Even if the asynchronous architecture cannot be independent on its own, interfacing it with synchronous systems could help develop a new generation computing architecture that would possibly sustain the quest for performance, speed and reliability. Development of robust CAD and simulations tools for the asynchronous design as in the synchronous design would be a strategic milestone for all stakeholders in the semiconductors industry.

Source : http://goafrit.wordpress.com/2010/05/27/the-asynchronous-design-implementation-process/