Design of Control Modules for Use in a Globally Asynchronous, Locally Synchronous Design Methodology

Pradnya Deokar

Department of Electrical and Computer Engineering, VLSI Design Research Laboratory, Southern Illinois University Edwardsville, Illinois, USA, 62025

ECE595: Masters Project

Abstract

In the proposed blended GALS design methodology, macromodular control elements are used to facilitate communication across different clocked domains. This paper describes the design and implementation of seven delay-insensitive control elements (Transfer, Call, Merge, Branch, Rendezvous, Decision, and Interlock) as standard cells. All seven control elements have been designed, successfully simulated, and laid out using the TSMC 0.25 micron process, available through MOSIS. The results of our simulations prove that this nearly universal set of control modules are completely free of both combinational and metastability hazards.

1.0 Introduction

Circuit design methods can be classified in two major categories: synchronous and asynchronous. The inherent adaptive nature, low-power circuit operation, composability and most important no need of global verification make asynchronous
methods attractive. While possessing these advantages, asynchronous design is also generally viewed as more difficult and is foreign to the majority of present day digital designers.

The Globally Asynchronous Locally Synchronous (GALS) [1] design approach combines advantages of both synchronous and asynchronous operations, eliminating the need for global timing verification. The specific methodology that we advocate (one of many proposed GALS design methodologies), blends clockless and clocked systems using a restartable clock generator [2] and seven macromodular control modules. Use of a clock generator eliminates the need for a synchronizer between the local and global domains, if the local domain is clocked. Macromodular control elements facilitate communication across subsystems in different clocked domains. Control signals generated by macromodular control elements provide necessary sequence controls for events in the system.

By creating a universal set of macromodular control modules as standard cells that can be incorporated into an existing standard cell library, we make it possible for designers with little or no previous asynchronous design experience to quickly create these blended systems and achieve the advantages of both synchronous and asynchronous operation. This macromodular set of seven control elements is not universal but can be made universal by the addition of the $S$ element [8] which is not described in this paper.
2.0 Synthesis of Control Elements

The seven macromodular control elements are Transfer, Branch, Merge, Rendezvous, Call, Decision and Interlock. The control elements use 2-phase or transition signaling where transitions represent a meaningful event. Hazard-free implementations of the macromodular control elements prevent transient errors in combinational circuit outputs due to unintended component delays. The design and implementation of these seven control elements is explained in this paper, and the Interlock element is described in additional detail in [3].

All seven of the control elements are synthesized using a method which we will summarize here. In the design method we employ, Petri Net models for the elements and the associated reachability graphs drive our synthesis procedure. Using an improvement well suited to automation, of the synthesis procedure described by Molnar et. al. in [10], logic equations for the control element are derived. These equations are then subject to Unger's ternary test for hazards [11]. The combinational hazards (logic races) found are eliminated by a trial-and-error procedure that is repeated until the ternary test proves the absence of combinational hazards.

In addition to combinational hazards, some of the circuits used to implement the various control modules display metastability hazards. These metastability hazards are identified, and an appropriate metastability detection circuit is constructed that detects when metastability has subsided. The falling edge of the metastability detection
circuit’s output transfers the now stable intermediate outputs levels to the flipflops that drive the control element’s final output lines.

The target technology used to implement these control elements is the TSMC n-well 0.25 µm process, available through MOSIS. The concept of minimizing logical effort is used to size transistors for minimal delay through the cell, with an assumption that the output of the cell is loaded with an external capacitance representing three times the gate capacitance of an inverter with drive strength of 2 (inv_2 from the Virginia Tech Standard Cell Library [9]). This method of minimizing delay is explained in [4]. Simulations to verify performance were executed using Cadence’s electrical simulator, Spectre.

Each of the control modules were laid out so that they could, at some point in the future, be fully incorporated into the TSMC 0.25 micron standard cell library created by researchers at Virginia Tech University [9]. In this standard cell library, cells are 15.2 µm high and the supply rails are 1.32 µm wide. The area occupied by each control element will be expressed in terms of the area of the Virginia Tech TSMC standard cell library buffer of driving strength two (buf_2).

2.1 Transfer Element

The Transfer element [5] is useful when a timed process must be initiated by a clockless sequencing network; thus, providing an interface between a clockless domain and a timed domain. The circuit used to implement the T element is shown in Figure 1.
The INIT and COMPLETE events belong to the clockless domain which is shown on the left in Figure 1, and the DONE and RUN events belong to the timed domain on the right. The INIT event produces a rising-edge RUN pulse which initiates a finite set of operations in a timed domain. After completion of these operations in the timed domain a DONE pulse produces a COMPLETE transition in the clockless sequencing network. The T element (as it was implemented) can be shown to be free of all combinational and metastability hazards if the INIT event is constrained to occur only after a COMPLETE event and if the RUN and DONE events are separated by a finite interval.

As shown in Figure 1(a), the output of the lower T-FF (flip-flop) is delayed by about 1 ns and fed back to one of the inputs of XOR to produce a RUN pulse of width of about 1 ns whenever there is an INIT event \textit{i.e.} a transition. The following equations for the T-FF are given in [6].

\[
Y_1 = \bar{x}y_2 + y_1y_2 + xy_1; \quad Y_2 = \bar{x}y_2 + \bar{y}_1y_2 + x\bar{y}_1 = z \quad \text{Eq. 1}
\]

The logic diagram for the T-FF used in the T element is shown in Figure 1(b). The CLR signal is used to initialize outputs RUN and COMPLETE to low. It is important that the user of the T-module ensures that the INIT signal is low when CLR is made active.
Figure 1a. Block diagram of Transfer element

Figure 1b. Logic diagram for T-FF

Note - Subscript ' represents inverted signal
Simulation results shown in Figure 2(a), demonstrate correct operation of the T element. An INIT event (a transition) at 5.0ns causes a RUN pulse of width approximated 1ns, and DONE pulse at 8.3ns causes a COMPLETE. The physical layout is shown in Figure 2(b). The area (167 μm by 15 μm) occupied by the T element is equivalent to that occupied by 34 standard cell buffers.

2.2 Call

The Call element [7] effectively allows two or more processors to use a shared resource. It provides mutually exclusive access to a shared resource (perhaps another processor) and after use of the shared resource, returns control to the calling processor.

The Petri net model of the hazard free single return Call element is shown in Figure 3(a). The synthesis of the hazard-free C element without feedback delay is explained in [5]. The resulting logic equations for a single return C element are as follows,

\[
\begin{align*}
\quad C' &= C(B + D \oplus E) + (B + C)(\overline{A}DE + \overline{A}\overline{D}E) \\
\quad E' &= E(D + B \oplus C) + (D + E)(\overline{A}BC + \overline{A}\overline{B}C) \\
\quad F' &= B \oplus D
\end{align*}
\]

Eq. 2

The superscript ’ represents the next state of output. These equations describe the three output variables C’, E’ and F’ inferred from the Petri net reachability graph. Our implementation of these logic equations is given in Figure 3(b). The CLR signal is added to the outputs of the Call element in order that the outputs C’ and E’ can be initialized to a LOW logical level. Simulation results are presented in Figure 4(a). The layout of the Call module is shown in Figure 4(b). The area (186μm by 15μm) occupied by the C element is equivalent to 38 standard cell buffers.
Figure 2a. Simulation results for T element

Figure 2b. Layout of T element
Figure 3a. Petri net model of single return Call element

Figure 3b. Logic diagram of Call element
Figure 4a. Simulation results for Call element

Figure 4b. Layout of Call element
2.3 Merge

The Merge element [7], as its name suggests merges two input into one output. In other words, if a transition occurs on either of its input lines, the output should make a transition. It is implemented as a XOR gate. The M element does not have a feedback path, and only one input can change at a time, if it is to be a hazard-free element. The Petri net model of the M module is presented in Figure 5(a). The logic equation for the Merge is derived from the Petri net model and its reachability graph in [7] and is

\[ c' = a \oplus b \]  

Eq. 3

The logic diagram of the Merge is shown in Figure 5(b). Simulation results demonstrating successful implementation of the XOR gate is presented in Figure 5(c). The layout is shown in Figure 5(d). The area (16 μm by 15 μm) occupied by the M element is equivalent to 3 standard cell buffers. The maximum speed of operation when loaded with our agreed upon standard load is 2.5GHz.

2.4 Branch

The Branch element [7] includes two buffers placed between its single input and two outputs. This is also hazard-free element as long as input transitions are properly placed.

\[ b' = a \]
\[ c' = a \]  

Eq. 4

The Petri net model of the Branch is shown in Figure 6 (a). A logic diagram is shown in Figure 6(b). Simulation results are presented in Figure 6(c), which shows two outputs b’ and c’ as buffered version of the input a.
Figure 5a. Petri net model of Merge element

```
Merge (M)
```

Figure 5b. Logic diagram of Merge element

```
Merge
```

Figure 5c. Simulation results of Merge element

Figure 5d. Layout of Merge element
The layout is shown in Figure 6(d). The area (14μm by 15μm) occupied by the B element is equivalent to 3 standard cell buffers.

2.5 Rendezvous

The Rendezvous element [7] effectively combines two inputs into single output with the output also providing feedback to the input. A positive-going transition must occur on both of its inputs before the output will transition high. When both inputs have returned to zero, the output will return to zero. At initialization it is important that the user ensure that both inputs are low or both inputs are high. The Petri net model of the R element used to drive the synthesis procedure is given in Figure 7(a). The derivation of the logic equations for the R element is detailed in [7]. The resulting logic equation is

\[ c' = ab + ac + bc \]  \hspace{1cm} \text{Eq. 5}

The superscript ' represents the next state of the output. Our implementation of these logic equations is given in Figure 7(b). The R element is also conflict-free and hazard-free. Simulation results are shown in Figure 7(c). The layout is shown in Figure 7(d). The area (22μm by 15μm) occupied by the R element is equivalent to 4 standard cell buffers.

2.6 Decision

The Decision element [5] selects one or the other of its two outputs depending on two biasing level inputs. The D element has an input A and two inputs B and C (for biasing
levels) as well as two outputs D and E. The biasing inputs (B and C) will decide whether a transition is to occur on either the D or the E output line.
Figure 6a. Petri net model of Branch

Figure 6b. Schematic diagram of Branch element

Figure 6c. Simulation results of Branch element

Figure 6d. Layout of Branch element
**Rendezvous (R)**

**Figure 7a.** Petri net model of Rendezvous element

**Figure 7b.** Schematic diagram of Rendezvous

**Figure 7c.** Simulation results of Rendezvous element

**Figure 7d.** Layout of Rendezvous element
The inputs B and C can be considered as transition signals or levels, but care should be taken to avoid them from being concurrent with input A.

The Petri net model of the D element is shown in Figure 8(a). The logic equations for D’ and E’ are

\[
D' = (B + D)(A \oplus E) + \overline{B}D \\
E' = (C + E)(A \oplus D) + \overline{C}E
\]

Eq.6

The superscript ‘ represents the next state of an output. The resulting logic diagram for the D element is shown in Figure 8(b). When both level inputs B and C are false, A has no effect on D and E. Thus D and E remain unchanged. When one of the inputs B or C is asserted, it is responsible for its corresponding output to make a transition whenever input A transitions. With both B and C true, a transition is selected randomly either on output D or on output E.

This last case may force the system into a metastable state leading oscillations on the outputs. Addition of a mutual exclusion circuit postpones any final output until both intermediate outputs have become fully stable. As shown in Figure 9(a), oscillations occurred on D’ and E’ at 140ns and 190ns are not seen on D” and E”. The metastability detector circuit prevents these hazards from being reflected at outputs D” and E”.

The layout for this hazard-free D element is shown in Figure 9(b). The area (340μm by 15μm) occupied by the D element is equivalent to 68 standard cell buffers.
Figure 8a. Petri net model of biased Decision element

Figure 8b. Logic diagram of biased Decision

Note - Subscript ‘ shows inverted signal
Figure 9a. Simulation results of biased Decision element

Figure 9b. Layout of biased Decision element
2.7 Interlock

The Interlock element ensures mutual exclusion between concurrent requests for a shared resource originating in remote processors. The Petri net model of the I element [3] is shown in Figure 10(a). The Interlock described in detail in [3] contained combinational hazards, which are removed by augmenting the following logic equations from [7].

\[
\begin{align*}
a' &= c \\
d' &= b \cdot (g \oplus h) + c \cdot d \cdot (g \oplus h) + b \cdot c \cdot d \\
e' &= g \\
h' &= f \cdot (c \oplus d) + g \cdot h \cdot (c \oplus d) + f \cdot g \cdot h
\end{align*}
\]

Eq.7

The superscript \(^\prime\) represents the next state of an output. Implementation of these logic equations is shown in Figure 10(b). When inputs to the interlock transition simultaneously, the interlock exhibits oscillatory behavior while trying to resolve the conflict. As in the D element, addition of an appropriate metastability detector circuit prevents oscillations from appearing in the final outputs. The metastability detector used in the I element is described in [3].

Simulation results presented in Figure 11(a) demonstrate how the interlock resolves the conflict between the two contenders successfully without any hazards. The layout for this hazard-free I element is shown in Figure 11(b). The area (298\(\mu\)m by 15\(\mu\)m) occupied by the I element is equivalent to 60 standard cell buffers.
Figure 10(a). Petri net model of Interlock Element

Figure 10(b). Logic Diagram of Interlock Element

Note- Subscript * represents inverted signal
Figure 11(a). Simulation results for Interlock Element

Figure 11(b). Layout of Interlock Element
3.0 **Summary**

The set these seven delay insensitive is not a universal set, but it can be made by addition of ‘S’ element described well in [8]. Delay-insensitive control and data paths are composed from this set of control elements implemented as standard cells and designed to be completely free of both combinational and metastability hazards. This guarantee is valid providing the composition of elements follows a few simple rules. Table 1 compares sizes and maximum speeds of all seven control elements discussed above.

**Table 1. Comparison of control modules in terms of area and speed**

<table>
<thead>
<tr>
<th>Input Type</th>
<th>Metastability</th>
<th>Size</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>T element</td>
<td>SIC</td>
<td>No</td>
<td>167um x 15um (34 beq)</td>
</tr>
<tr>
<td>C element</td>
<td>SIC</td>
<td>No</td>
<td>186um x 15um (38 beq)</td>
</tr>
<tr>
<td>M element</td>
<td>SIC</td>
<td>No</td>
<td>16um x 15um (3 beq)</td>
</tr>
<tr>
<td>B element</td>
<td>SIC</td>
<td>No</td>
<td>14um x 15um (3 beq)</td>
</tr>
<tr>
<td>R element</td>
<td>MIC</td>
<td>No</td>
<td>22um x 15um (4 beq)</td>
</tr>
<tr>
<td>D element</td>
<td>MIC</td>
<td>Yes</td>
<td>340um x 15um (68 beq)</td>
</tr>
<tr>
<td>I element</td>
<td>MIC</td>
<td>Yes</td>
<td>298um x 15um (60 beq)</td>
</tr>
</tbody>
</table>
References


[6] Personal communication with Dr. J.R. Cox


