

TESTABLE PROGRAMMABLE DIGITAL CLOCK PULSE CONTROL ELEMENTS

Kenneth D. Wagner*
 Synopsys, Inc.
 700 East Middlefield Road
 Mountain View, CA 94043-4033

Bernd K. Koenemann
 IBM Corporation
 M/S P-121, P.O. Box 950
 Poughkeepsie, NY 12602

Abstract

Digital clock pulse control elements - delay lines and pulse-shaping elements - are used widely for clock generation and clock tuning in synchronous digital logic. However, they are intrinsically redundant circuits: without special modifications, DC logic testing cannot completely verify their static behavior (including the correct operation of the decoders and selectors used for their programming and control). This paper demonstrates low overhead circuit modification techniques that can be applied to all classes of programmable clock control elements, ensuring their complete single stuck-at fault testability.

1. Introduction

In synchronous digital systems, a sizable percentage of circuits may be devoted to the clock system, so their testability is a significant concern. Circuits used for clock system design [Wag88] include buffers, delay lines and pulse-shaping elements. Digital clock pulse control elements - the delay lines and pulse shapers - adjust clock pulse edge positions and modify clock pulsewidths. They are used frequently in the clock generation portion of clock systems.

Fig. 1 shows a generic clock system design for a synchronous digital system. A key design issue is control of clock skew - differences in clock pulse edge arrival times between desired and actual values. To make clock system design flexible and to control clock skew, programmable versions of the clock pulse control elements are often used to implement a clock tuning strategy [Hun91] [Koe88] [Mur91] [Dej89][Fis91]. Generally, clock pulses are distributed from the source OSCillator with attention paid to only one clock pulse **reference edge** - clock pulse control elements in clock generation then derive clock signals that need high correlation from this single edge [Def 88]. Pulsewidths are adjusted late in the clock signal paths to meet the requirements of the clocked sequential elements at their destinations - flip-flops, latches, memories, etc.

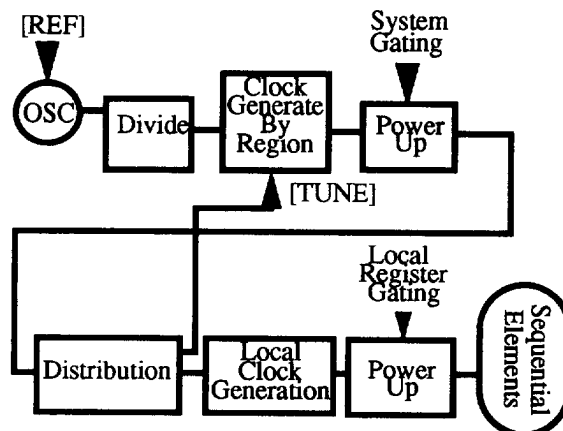


Figure 1. Generic Clock System Design

Both programmable and non-programmable pulse-shaping elements as well as programmable delay lines rely on redundancy for correct functional operation. Standard logic optimization (redundancy removal) techniques cannot be applied to them to enhance their testability, since they preserve only static (boolean) operation and not dynamic function. Since the correct operation of these elements depends on their dynamic characteristics, special modifications must be made to ensure their complete single stuck-at (DC) fault testability without destroying their dynamic properties.

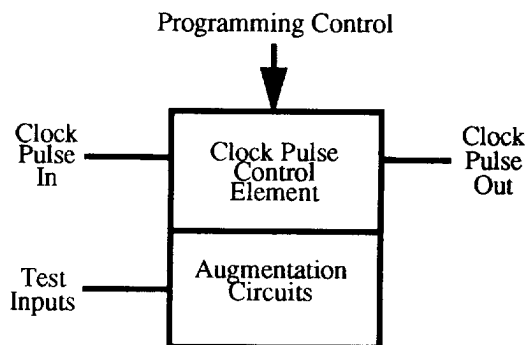


Figure 2. Testable Clock Pulse Control Element

* The author performed the majority of this work while at IBM Corporation.

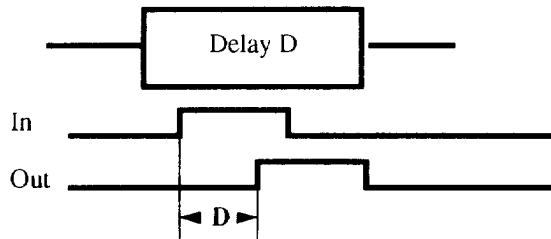
Fig. 2 shows a generalized programmable clock pulse control element. Modifying it to make it 100% single stuck-at fault testable will require the addition of test control inputs and extra logic. The motivation for such complete DC testability of a dynamic element is discussed in Section 2.

To begin, we describe programmable delay lines and pulse-shaping elements.

1.1 Delay Lines

A digital delay line adds a known delay to an input pulse. The fixed delay type delay line is shown schematically in Figure 3. In clock system design, the digital delay line is usually relied upon to accurately reposition (delay) the reference edge of the input pulse. Since logic devices have asymmetric rising and falling delays, both the digital delay line and any logic gates in the downstream clock distribution will distort the clock pulse. To compensate, the pulsewidth is carefully adjusted later in local clock generation (see Fig. 1).

Fixed Delay



Programmable Delay

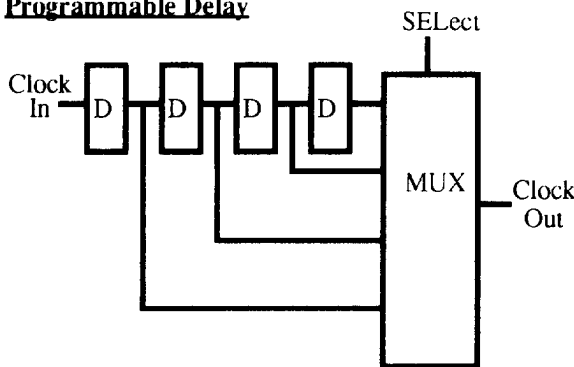


Figure 3. Delay Lines

Fig. 3 also shows a programmable delay line, where the input signal may be delayed by D , $2D$, $3D$ or $4D$. The SElect lines control the choice of delayed input through the MULTiplexer switch. Programmable digital delay lines and system tuning strategies using such elements are

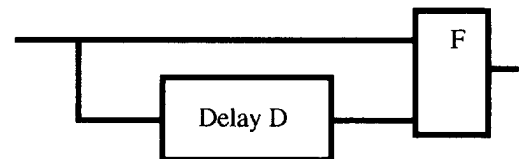
described in [Hun91] [Koe88] [Mur91] [Dej89] [FIS91] as well as many other sources.

1.2 Pulse-Shaping Elements

Figure 4 shows the generic pulse-shaping elements. They implement symmetric two variable Boolean functions, where one input to the reconvergent gate is a delayed and possibly inverted version of the other input. A symmetric Boolean function F or G with two inputs $X1$, $X2$ has the property $F(X1, X2) = F(X2, X1)$. There are $2^4/2 = 8$ such functions. Discarding the trivial $F=1$ and $F=0$ cases leaves six functions. They are AND, NAND, OR, NOR, XOR and EQU. We consider only the AND, OR and XOR cases. The three inverted output functions, NAND, NOR and EQU have identical modifications to achieve complete single stuck-at fault testability as their AND, OR and XOR counterparts.

In Fig. 4, if $F=AND$ or OR , the upper circuit where the Delay block is non-inverting (so the reconvergent block F receives inputs of the same polarity) may operate to shrink or stretch the input pulse - hence it is called a shrinker or stretcher; the lower circuit has an inverting Delay block (so the reconvergent block G receives inputs of opposite polarity) and may act to chop the input pulse - a chopper [Def85]. The interpretation of element operation depends on the defined active polarity of the input pulse and is always relative to a reference edge. In either case, if F or $G = XOR$ the circuit operates as an edge detector.

Non-Inverted Reconvergence



Inverted Reconvergence

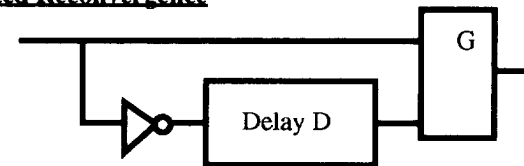


Figure 4. Generic Clock Pulse-Shaping Elements

The corresponding timing waveforms in Figure 5 show the effect of the pulse-shaping elements on positive active and negative active input clock pulses. They operate on a single (reference) edge of the input clock pulse, the other edge of the pulse has no effect. The delay through block F or G and rise time and fall time are all set to zero to simplify the tim-

ing diagrams. In all cases, the pulsewidth W must exceed the delay D for the element to have the desired effect. Programmable versions of these control elements allow the delay D of the Delay block to be selectable.

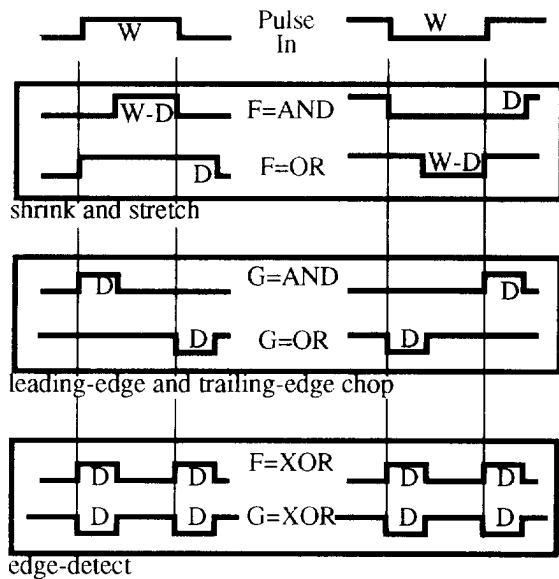


Figure 5. Operation of Pulse-Shaping Elements

The pulse-shaping operations depicted in Fig. 5 are sufficient to determine all other operations differing only by inversion. For example, to produce a positive pulse of width D at the trailing-edge of a positive input pulse, the $G=OR$ operation need only be inverted, i.e., $G=NOR$ will produce the correct trailing-edge clock chopper.

2. Conversion to 100% Single Stuck-At Fault Testable Circuits

2.1 DC Test of Clock Pulse Control Elements

Clock pulse-shaping elements require reconvergence to operate dynamically, and programmability for all clock pulse control elements requires multiplexing, which itself requires reconvergence. Therefore, ensuring such elements are 100% single stuck-at fault-free requires modifications carefully designed not to destroy reconvergence.

Without modifications, clock pulse control circuits have obvious undetectable faults. For instance, in a static test of a programmable delay line or non-inverted reconvergence pulse-shaping element, any value placed at the circuit's pulse input will appear (or appear inverted) at the circuit output. This behavior occurs regardless of whether decoding circuits select the wrong delay block or blocks. And in an inverted reconvergence pulse-shaping element, the final (static) circuit output stabilizes at a fixed one or zero level

regardless of the input. Thus, clock not shaped (not chopped, not shrunk or not stretched) faults, when delayed reconvergent lines are stuck at non-controlling values at the input of the reconvergent gate, are undetectable in a static test without special circuit modifications.

A fundamental approach to logic testing is the use of DC tests to verify the correct static operation of a network. Of course, DC testing cannot verify the dynamic operation of clock pulse control elements (e.g., the pulse has been delayed 3 nanoseconds or chopped to 1.5 nanoseconds) - such testing requires an additional AC or delay testing component. Dynamic network operation may be verified through observation of clock network primary outputs, or indirectly through the timed observation of storage element contents under control of such clocks.

A clock pulse control element is typically embedded in a larger clock generation network. Therefore, in a DC test, the control of the clock pulse input and the observation of the delayed and shaped clock pulse output must be considered in addition to modifications for 100% single stuck-at fault testability of the clock pulse control elements themselves. All modifications must be carefully constructed to have minimal impact on clock skew in the clock system. If no other means is available, the pulse input may be controlled by adding a 2-input AND gate before the element, one input the dynamic clock and the other a controllable level signal (a clock input stuck-at fault must then be considered separately). At the output side, primary output observation or two-image sequential test at one of the clock destinations will suffice.

The following conversion techniques require modifying the clock pulse control elements by adding test inputs and logic. The additional test inputs are level signals, so they may be either primary inputs, or, more efficiently, easily controlled (scannable) latch or flip-flop outputs. The elements are transformed to irredundant networks. Since all logical redundancies are removed they become fully single stuck-at fault testable. Then DC (static) testing can cover (control to both one and zero and sensitize the effect to the observable output) all internal circuit lines and circuit I/O.

Note. Many logically equivalent irredundant representations of the clock control elements are possible (e.g., a NOR-NOR implementation is equivalent to the OR-AND implementation shown in Figure 4). In addition, variations on the basic building blocks of the introduction section may use inverted outputs or inputs. The choice of the most efficient implementation is dictated by the technology and circuit library, and may be found through repeated application of DeMorgan's Laws.

2.2 Testable Programmable Digital Delay Line

The programmable digital delay line is used to allow a selectable choice of delays. Only one specific path is chosen. Since all paths propagate the same input signal, failure to select the correct path or the selection of multiple paths simultaneously cannot be detected in a conventional DC (static) test.

One method to make such an element testable requires the addition of two circuit inputs, both providing level signals. In practice, the (Test) Mode circuit input would come from a primary input and the Parity circuit input from a controlled (scannable) latch or flip-flop.

The method uses “bad parity” to add controllability at the selector. The Hamming distance between the bit patterns of any two valid (sensitizing a single path from input to output) input selections is made two or more. Then any single failure in one of the selection bits is a distance one change - no other path will be selected and testability will be maintained. Since parity is a distance two code, it is the simplest choice. To assure proper function in normal operational mode, the effect of the parity bit is suppressed by degating the derived parity signals based upon a control input.

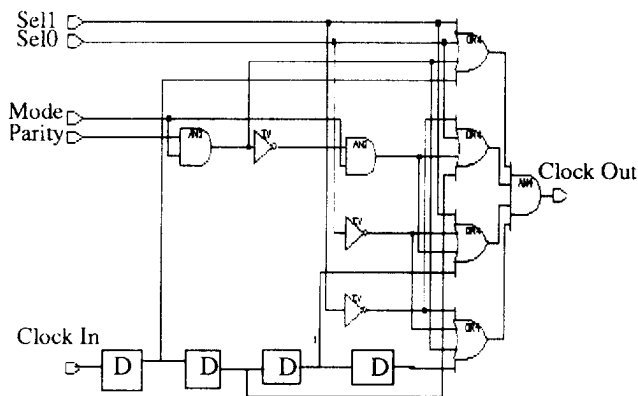


Figure 6. Testable Programmable Digital Delay Line

Figure 6 shows an example of a programmable digital delay line with four possible delay selections. The circuit uses a 4-to-1 selector with an external parity input, Parity, attached. The Mode control input is set to logic 0 in functional mode. This forces the parity inputs to the 4-input OR-selector gates to the non-controlling value (logic 0). Thus, the parity logic does not affect the functional operation.

For testing, it is assumed that the Mode signal is switched to logic 1, and the parity value, Parity, can be switched as

needed for test generation. The true and complement phases of the parity signal are connected to the OR gates in the selector as needed for odd or even parity. This technique is extensible to any number of selections.

An alternative implementation assuring complete testability trades area to save one circuit test input. Instead, Parity is calculated by adding a parity tree to the test logic. Thus, this implementation requires only an additional Mode input.

2.3 Testable Programmable Pulse-Shaping Elements

The programmable clock shrinker, stretcher and chopper are used for pulse-shaping. They produce an output pulse derived from the leading or trailing edge of an input pulse with a well-controlled pulsewidth. Fig. 7 shows a generic unoptimized design for such elements, where the presence or absence of the inverter depends on the dynamic operation required.

When these programmable elements function correctly, the output pulse is expected to be glitch-free and the output pulsewidth to be greater than the minimum required for proper sequential element behavior. A necessary but not sufficient condition to guarantee such pulse properties is that the input pulsewidth must exceed the sum of all delays in the selectable delay chain.

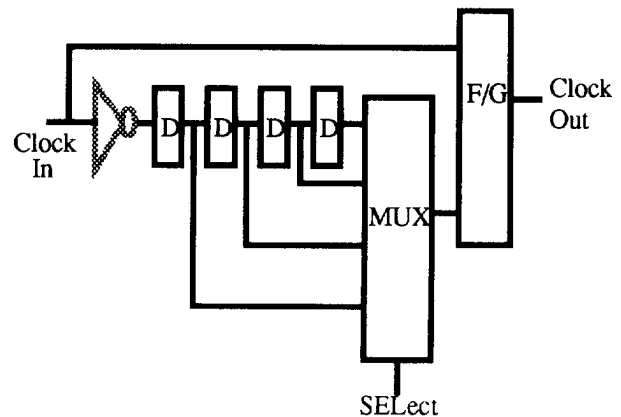


Figure 7. Generic Programmable Pulse-Shaping Element

Fig. 8 shows an alternative optimized implementation for the generic pulse-shaping elements. This implementation forms the basis for our testability modifications.

The $n-1$ OR gates in series with the clock input are also fixed delay elements. Assume that each has a delay D and all other circuit delays are neglected. Then the final recon-

reconvergent AND gate received the undelayed clock input and may also receive versions delayed by D , $2D$, ..., $(n-1) \cdot D$. An edge of the undelayed pulse determines an edge of the circuit output pulse; and an edge of the most delayed ungated clock pulse (all further delayed clock inputs are non-controlling at the reconvergent gate based on the programming) determines the complementary edge position of the output pulse. The controlling input edges depend on the pulse shaping function and can be easily found. In all cases, the input pulsewidth W must exceed $(n-1) \cdot D$.

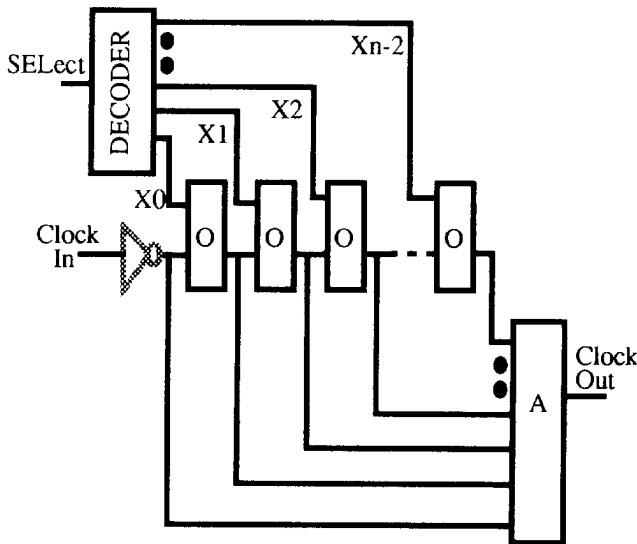


Figure 8. Minimized Programmable Pulse-Shaping Element

Fig. 9 shows the basic modifications used to make the pulse-shaping elements testable. They rely upon minimizing the decoding network used to generate selection signals. For instance, an eight selection network would normally be realized with a complete decoder (i.e., three inputs fully decoded to eight one-hot select signals). Instead, we recognize the redundancy in this subcircuit when coupled to the reconvergent gate F or G. To solve the testability problem, we minimize the decoding network to eliminate reconvergence through its terms. A separate minimized decoder of comparable size (requiring three independent test only inputs) is then added to fully control path selection.

Note. Minimization of the original decoding network to reduce logic gate requirements may be performed independent of any testability related modifications, i.e., the Test_Sel inputs and their internal circuit propagation may be dropped in each of the following examples and pulse-shaping function (but not testability) is still preserved.

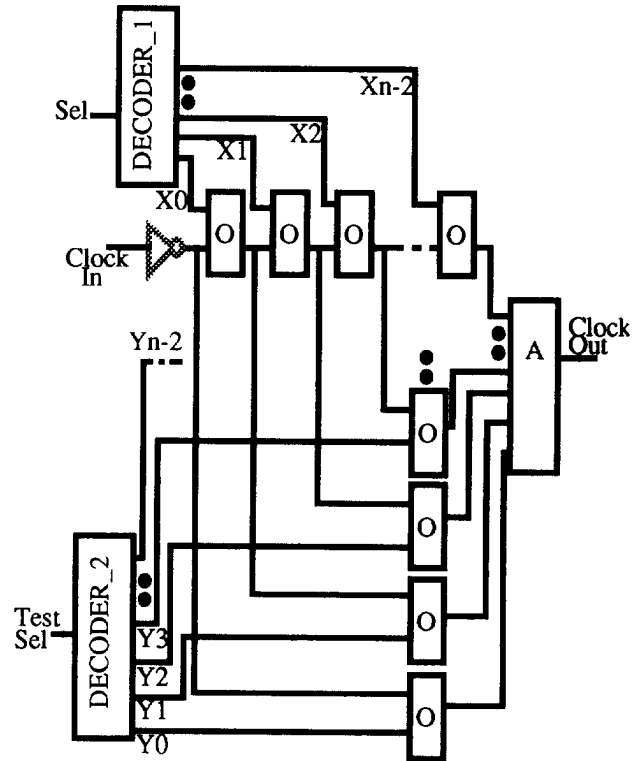


Figure 9. Testable Programmable Pulse-Shaping Element

2.3.1 Testable Programmable Shrinker

A fully testable programmable shrinker is shown in Figure 10. The three serial OR gates represent the delay elements of the selectable delay chain, each with delay D . In functional operation, test selection inputs Test_Sel0 and Test_Sel1 are set to zero, and a positive active input pulse will be shrunk by 0 , D , $2D$, or $3D$, based on decoding the selection inputs Sel0 and Sel1 to 11, 10, 01 and 00 respectively.

In test operation, the added decoder with inputs Test_Sel0 and Test_Sel1 will be controlled by the test generator to specifically sensitize the path where the fault is being propagated and degate all other paths in conjunction with the Sel0 and Sel1 based original decoder. The added decoder must also be minimized and can be derived from the analysis done to minimize the original decoder (see Section 2.3.2).

Note. As shown in Figure 5, a negative active input pulse will be stretched in this circuit by 0 , D , $2D$, and $3D$ respectively based on the selection inputs.

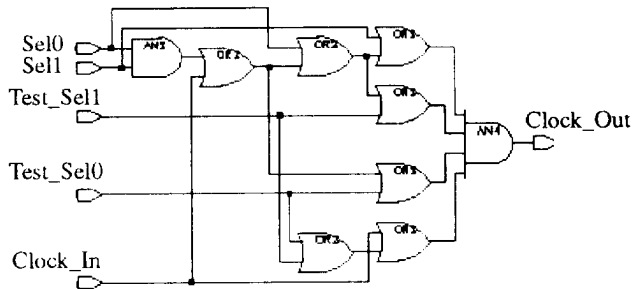


Figure 10. Testable Programmable Shrinker

2.3.2 Extension to Larger Decoders

Building minimized decoders for the shrinker, stretcher and chopper is quite simple. No Selection inputs are ever required in both complemented and uncomplemented form. To construct the decoder, consider the following eight selection example, where $X_i=1$ means i delay elements in the delay chain are selected (an "x" entry means don't care) as shown in Figs. 8. A second decoder is added to provide the Y_i (fault sensitization path) select signals derived from the added Test_Select inputs as shown in Fig. 9.

Both decoders can be developed from the same information: boolean AND in the X_i equations is replaced by boolean OR in the minimized Y_i equations.

Example

select			Decoder Signal							
0	1	2	X0	X1	X2	X3	X4	X5	X6	
1	1	1	1	x	x	x	x	x	x	
1	1	0	0	1	x	x	x	x	x	
1	0	1	0	0	1	x	x	x	x	
1	0	0	0	0	0	1	x	x	x	
0	1	1	0	0	0	0	1	x	x	
0	1	0	0	0	0	0	0	1	x	
0	0	1	0	0	0	0	0	0	1	
0	0	0	0	0	0	0	0	0	0	

$$\begin{aligned}
 X_0 &= sel_0 \cdot sel_1 \cdot sel_2; & Y_0 &= test_sel_0 + test_sel_1 + test_sel_2 \\
 X_1 &= sel_0 \cdot sel_1; & Y_1 &= test_sel_0 + test_sel_1 \\
 X_2 &= sel_0 \cdot sel_2; & Y_2 &= test_sel_0 + test_sel_2 \\
 X_3 &= sel_0; & Y_3 &= test_sel_0 \\
 X_4 &= sel_1 \cdot sel_2; & Y_4 &= test_sel_1 + test_sel_2 \\
 X_5 &= sel_1; & Y_5 &= test_sel_1 \\
 X_6 &= sel_2; & Y_6 &= test_sel_2
 \end{aligned}$$

Notice that such decoders are very simple compared with complete decoders [McC86]. These decoding elements actually perform the inverse function of highest priority encoding. They are also convenient because complemented selection bits (both Sel and Test_Sel) are never required and the functional operation can always be satisfied by setting all Test_Sel bits to zero. Their complexity is predictable by the combinatorics of ${}^N C_i$, where N is the number of selections and i is the number of decoded inputs (ranging from 1 to N). For instance, for a 6 variable decoder (up to 64 selections), one requires:

- 1 decode of all 6 inputs
- 6 decodes of 5 inputs
- 15 decodes of 4 inputs
- 20 decodes of 3 inputs
- 15 decodes of 2 inputs
- 6 "decodes" of 1 input (trivial - direct connection)

where each X_i decode can be satisfied by an AND gate and each Y_i decode with an OR gate, both with the appropriate number of uncomplemented inputs (or their boolean equivalent).

2.3.3 Testable Programmable Stretcher

We next examine a fully testable programmable stretcher, shown in Fig. 11. The circuit is a simple modification of the shrinker. Examining Fig. 5, it can be seen that inverting a positive active input clock to an $F=AND$ reconvergent block circuit and then reinverting the final output has the same effect as stretching a positive active pulse in an $F=OR$ reconvergent block element (since $(A \cdot B)'$ = $A+B$). Testability modifications required of this element are identical to the shrinker example.

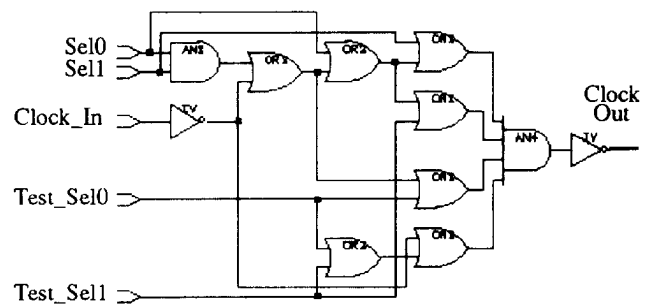


Figure 11. Testable Programmable Stretcher

As in Fig. 10, the three serial OR gates represent the delay elements of the selectable delay chain, each with fixed delay D . In functional operation, test selection inputs Test_Sel0 and Test_Sel1 are set to zero, and a positive active

input pulse will be stretched by 0, D, 2D, or 3D, based on decoding the selection inputs Sel0 and Sel1 to 11, 10, 01 and 00 respectively.

In test operation, the added decoder with inputs Test_Sel0 and Test_Sel1 will be controlled by the test generator to specifically sensitize the path where the fault is being propagated and degate all other paths in conjunction with the Sel0 and Sel1 based original decoder.

Note. As shown in Figure 5, a positive active input pulse will be shrunk in this circuit by 0, D, 2D, and 3D based on the selection inputs.

2.3.4 Testable Programmable Chopper

As an inverted reconvergence example, a fully testable programmable chopper is shown in Fig. 12. The implementation shown is identical to the testable programmable shrinker of Figure 10 with the exception that the delayed (upper) leg of the reconvergent gate G is inverted. And the same functional and test operations apply.

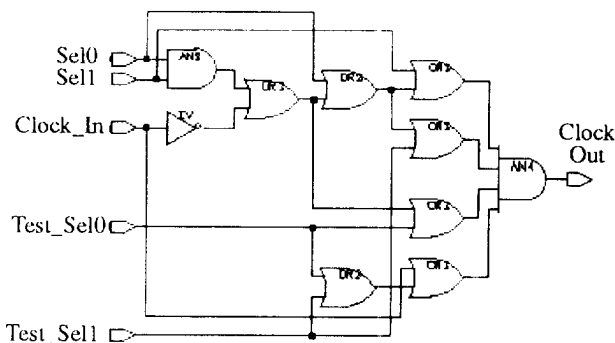


Figure 12. Testable Programmable Chopper

2.3.5 Testable Edge Detector

As a more trivial example, we last examine the testable edge detector shown in Fig. 13. In general, such an element would not need programmability so it will not be provided here (although the techniques used for the previous testable programmable elements could be used if programmability were a concern). The relative overhead in gates and inputs in this small circuit is large. The circuit is constructed such that all four possible two bit patterns can appear at the XOR gate. If the XOR circuit is implemented such that only three patterns are required for test, one of the test inputs and the AND gate it feeds can be eliminated.

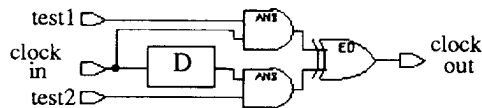


Figure 13. Testable Edge Detector

3. Implementation Cost

The methods described can be used to modify all of the clock pulse control element types, making them fully stuck-at fault testable. All methods using in series delay elements assume that they are precision delay elements that introduce negligible skew. The circuit overhead depends on the programmability of the element; the more selections required, the lower the relative testability overhead.

For the programmable delay lines, the overhead is one to two circuit inputs (one usually a primary input that may already exist) plus an extra parity value input at each select gate. For the pulse-shaping elements, the overhead is an added (minimized) decoder plus N-1 two-input select gates to support N selections. $\lfloor \log_2 N \rfloor$ additional static test inputs are also required; these may be sourced by latches, flip-flops or I/O.

The addition of gates or fanin in the dynamic signal path does add some tolerance (clock skew) to the element operation, affecting critical edge placement. For the programmable delay line, an additional fanin is added to each select gate. For the programmable pulse-shaping elements, the skew of one two-input circuit is added (a dummy two input element may be added on the single ungated line into the reconvergent gate F or G to equalize skew across all selections) on each path from the clock pulse input to controlled pulse output.

4. Conclusions

Digital clock pulse control elements - delay lines and pulse-shaping elements - are used widely for clock generation and clock tuning in synchronous digital logic. However, they are intrinsically redundant circuits: without special modifications, DC logic testing cannot completely verify their static behavior (including the correct operation of the decoders and selectors used for their programming and control). This paper demonstrates low overhead circuit modification techniques that can be applied to all classes of programmable clock control elements, ensuring their complete single stuck-at fault testability.

References

- [Wag88] Wagner, K.D., "Clock System Design", IEEE Design & Test of Computers, Vol. 5, No. 5, October 1988.
- [Hun91] Hunt, W.J. and J.C. Purkett, "Self-Calibrating Adjustable-Time Digital Delay," IBM Technical Disclosure Bulletin, Vol. 33, No. 8, January 1991.
- [Koe88] Koerner, S. et al, "Programmable LSSD Clock Generator," IBM Technical Disclosure Bulletin, Vol. 31, No. 2, July 1988.
- [Mur91] Murakami, D. and T. Kuwabara, "A Digitally Programmable Delay Line and Duty Cycle Controller with Picosecond Resolution," Proc. IEEE Bipolar Circuits and Technology Meeting, 1991.
- [Dej89] Dejjhan, K. et al, "A New High-Performance Programmable Delay Line IC," IEEE Trans. on Consumer Electronics," Vol. 35, No. 4, November 1989.
- [Fis91] Fischer et al., Programmable Delay Line Utilizing Measured Actual Delays to Provide a Highly Accurate Delay, United States Patent Number 5013944, May 7, 1991.
- [Def 88] Defazio, John J., "On-Chip LSSD System Clock Generation," IBM Technical Disclosure Bulletin, Vol. 30, No. 11, April 88.
- [Def 85] Defazio, John J., "Single-Edge Pulse Generator," IBM Technical Disclosure Bulletin, Vol. 27, No. 10A, March 1985.
- [McC86] McCluskey, E.J., Logic Design Principles, Ch. 11, Englewood Cliffs, New Jersey: Prentice-Hall, 1986.