Abstract: The authors introduce two low-cost, modular, totally self checking (TSC), edge triggered and error propagating (code disjoint) flip-flops: one, a D flip-flop used in TSC and strongly fault secure (SFS) synchronous circuits with two-rail codes, the other a T flip-flop, used in a similar way as the D flip-flop but retaining the error as an indicator until the next presetting, to aid error propagation. Thus, the self checking T flip-flop can be used as an error indicator. The self checking D flip-flop is smaller than the duplicate D flip-flop circuitry by 30%. The self checking T flip-flop error indicator is 60% smaller than the previous error indicator in the literature. These circuits, unlike previously reported circuits, can also detect stuck-at faults in the clock inputs. The authors have also presented TSC/CD flip-flops: a counter and a shift register.

1 Introduction

Current circuit complexity makes the determination of errors in a circuit arduous. Totally self checking circuits, proposed by [1], use input output coding to determine whether a circuit is operating accurately. The code disjoint (CD) property is utilised to design TSC circuits at the system level. Circuits which are CD propagate the error through to indicate the error on the output. However, it is hard to devise circuits that are both TSC and CD.

A strongly fault secure circuit is a circuit that becomes a TSC circuit after a finite number of faults, and until then will operate correctly (or is fault secure). If, for the same faults, the circuit is code disjoint and then becomes self testing and still remains code disjoint, the circuit is said to be strongly code disjoint. Strongly fault secure circuits satisfy the goals of totally self checking circuits.

Unordered codes [2] are used for input output codings in circuits that are TSC and SFS. Techniques have been presented for the creation of TSC [3], SFS [4] and SFS/SCD [2] combinatorial circuits for a class of unordered codes.

As far as synchronous designs are concerned, there are no circuits to date that exhibit all the TSC properties. The sequential designs, which appear in [5], add some extra circuitry to the clock of the ordinary memories to make register files with self testing load signals. The register file in [5] is not self testing for multiple unidirectional faults at clock input lines. The sequential circuits used in [5] with the proposed register file are based on the model given in [6], which does not consider the faults in storage elements. Furthermore, these circuits did not incorporate preset and clear with unordered codes in their circuits. No single-chip flip-flop circuit with unordered coded clock, preset and clear that is TSC with the error propagating property, has yet been proposed.

The DD_n-FF is an edge-triggered D-FF with the TSC property that is also error propagating. TT_n-FF can be used as an error indicator circuit [7]. These circuits use two-rail code for the input output coding. For any two bits of a pair of two-rail code, we use one DD_n-FF or one TT_n-FF.

The size of the proposed DD_n-FF is 30% less than duplicate D flip-flops. The duplicate D flip-flops circuit is not self testing for multiple unidirectional faults at clock input lines. The size of an edge triggered register file with two rail codes using the DD_n-FFs is 33% less than the same register file proposed in [5] with Berger code. (Berger codes have the smallest number of bits among unordered codes.) The size of TT_n-FF is 60% less than the error indicator circuit in [7]. Notice that the [7] circuit is only an error indicator and uses system clock which is not even self tested. The error indicator in [7] was proposed as a simpler circuit than the error indicator circuit in [8].

We present simple applications for the above DD_n and TT_n flip-flops in the form of a TSC/CD counter, a TSC/CD shift register. DD_n-FF and TT_n-FF were introduced in [9, 10], respectively.

2 Definitions and notations

Input-output coding techniques are used to detect faults automatically. Single stuck at 0 and stuck at 1 faults and multiple unidirectional faults [11] are the faults modelled in this paper. It is also assumed that the time interval between two faults is long enough for the proper cycle and input code to pass through the circuit.

(i) A circuit is self-testing (ST) if, for every fault from a prescribed set, the circuit produces a noncodeword output for at least one codeword input [1].

(ii) A circuit is fault-secure (FS) if, for every fault from a prescribed set, the circuit never produces an incorrect codeword output for any codeword input [1].
(iii) A circuit is totally self-checking (TSC) if, for every fault from a prescribed set, the circuit is both ST and FS [1].

(iv) A circuit is code disjoint (CD) if it always maps codeword inputs to codeword inputs to noncodeword outputs [1].

(v) A circuit is strongly fault-secure (SFS) for a fault set F iff, for every fault f in F, either the circuit is TSC for cfs, or the circuit is FS for v) and, if fault f occurs, the resultant circuit is still SFS for F - f [4]. It has been shown that TSC goal can be achieved by SFS circuits [4].

(vi) A circuit is strongly code disjoint (SCD) for every fault set F iff, for every fault f in F, either (a) the circuit is CD, and is ST for f, or (b) the circuit is CD, and if fault f occurs, the resultant circuit is still SCD for F - f [12]. Note that the checker circuits need to be TSC/CD. It has been shown that TSC goal can be achieved by SFS circuits [4]. The strongly code disjoint (SCD) property in SFS circuits has the same goal as CD property in TSC circuits. A checker can be SFS/SCD [12].

3 DD2n,FF equations and properties

Internal next state and output equations of DD2n,FF with the circuit and symbolic diagram are shown in Fig. 1. In equations of Fig. 1 the complementary parts are labelled with an extra n (i.e. C = clock and Cn = C', Dn = D'). 'X, Xn' and 'Q, Qn' are complementary pairs in fault-free operation with input codewords.

![Circuit Diagram](image)

**internal next state equations:**

\[
X = C_n \cdot D_n + C \cdot X
\]

\[
X_n = C_n \cdot D + C \cdot X_n
\]

**output equations:**

\[
Q = C \cdot X_n \cdot C_n \cdot Q
\]

\[
Q_n = C \cdot X \cdot C_n \cdot Q_n
\]

**Fig. 1 Equations, circuit diagram, clock and symbolic representation of DD2n,FF**

In each set of equations for internal states and output, the min-terms are separated in two parts, called the operating min-terms and latching min-terms, respectively. For the XXn pair, the C.X and C.Xn are latching part, because they keep the value of X and Xn constant when C = 1(Cn = 0). Similarly, Cn.Q and Cn.Qn keep the value of Q and Qn constant when C = 0(Cn = 1).

The values of Q and Qn will change due to values of Xn and X, respectively, when C = 1(Cn = 0). When C = 0(Cn = 1), the values of X and Xn change due to DDn. Therefore, the flip-flops are positive edge triggered, because at this edge, X and Xn stop changing (due to inputs) and Q and Qn accept the values of Xn and X.

During the fault-free operation of DD2n,FF, any DDn codeword values (01 or 10) are transferred to QQn at the following positive edge of clock (Cn change from 01 to 10). Similarly, the noncodeword values (00 or 11) of DDn are transferred to QQn. In the DD2n,FF, any noncodeword output can be changed to any other codeword or noncodeword by the input codeword or noncodeword. Therefore, the DD2n,FF satisfies the error propagating property.

The function table of the DD2n,FF is shown in Table 1. Sixteen possible conditions are marked with d1, d2, ..., d16.

<table>
<thead>
<tr>
<th>Table 1: Function table for DD2n,FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>QQn(t + 1)</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>00</td>
</tr>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

Every possibility of a stuck at fault has been checked. For instance, for any stuck-at fault, there is a condition for the circuit that can check that fault. The test conditions will occur during circuit clocking and DDn input changes. To check any possibility of stuck-at fault, the input lines of AND gates are marked with a number. In Table 2, both stuck-at faults for each line are checked to produce the noncodeword output with one of input codeword conditions (d1, d2, d3, d4). Some errors appear instantly, while other errors appear after the next positive edge. In Table 2, all conditions with a dot above it (d4) appear immediately, while others appear after the next positive edge. Both kinds of error indication for line 1 are shown in Fig. 2. L1 shows the signal at line 1. The thicker section of the line indicates that the line is stuck at fault and output is a noncodeword for that period. If, as shown in Fig. 2 (2), L1 is
stand-at-1, at the output a noncodeword appears immediately. On the other hand, if there is a stuck-at-0 fault as shown in Fig. 2 (5), the noncodeword appears after the next positive going edge. Any noncodeword output remains at least for one clock edge and it can be detected by the error indicator circuit. The stuck-at fault of OR gates are checked automatically. The stuck-at faults of inputs and clock which have multiple fan-out is explained in the next Section for the code disjoint property.

3.1 Clock considerations
Errors in the clock inputs have been taken into consideration. As can be seen from Fig. 1, QO will be 00 when CC = 00. Since QO = 00 is a noncodeword, an error is indicated. When CC = 11 (for the DD,-FF), if DD = QO, the values of XX, and QO will not change from their previous values. However, when DD = QO both XX, and QO will become 11, which being a noncodeword will indicate an occurrence of an error.

For normal operation of the clock circuit, CC has to pass a transient state of 00 or 11 when change from 01 to 10 or from 10 to 01. However, it is desirable to have a 11 intermediate transient state, which should last for less than the propagation delay of the flip-flop circuit. This can be achieved by passing the original clock through a NAND latch circuit. The changes of DD inputs should be avoided during clock transition time. Such a change may produce an unknown output. This problem also exists for other fault-tolerant design methods [5].

3.2 Preset (PR, PR,) and clear (CLR, CLR)
The complementary parts of PR, PR, and CLR, CLR, are shown in Fig. 1 with dashed lines. Note the flip-flop is positive edge triggered. The CC lines can be swapped to make the flip-flop negative edge triggered.

The circuits of DD,-FF in Fig. 1 are formed with AND-OR gates. For case of use in VLSI circuits, the DD,-FF can be made with NAND gates as given in [9]. Therefore, their PR, PR, and CLR, CLR, lines in Fig. 1 are activated by 10 values. These clear and preset lines are added with TSC/CD property. These lines are used for normal operation and presetting of the flip-flop circuits. These preset and clear lines restore the circuit to an error-free state.

The TSC property of CLR, CLR, PR and PR, line is apparent, since any incorrect stuck-at-0 or -1 will create a 00 or 11 combination for the CLR, CLR, or PR, PR, lines. These noncodeword combinations latch one of the X, X, Q or Q to a fixed value (0 or 1). This fixed value will result in an error condition.

For the CD (error propagating) property different situations should be considered. If these clear and preset lines are not used (PRPR = 10, CLRCLR = 10) and they are connected to a logical level, only the stuck-at error situation has to be considered. The other cases are where only one of these two pairs is connected to constant value (01) and the other is under control of data. In this case, if controlled lines have a noncodeword (00 or 11), one of Q or Q, values will latch to 0 or 1 without affecting the value of the complementary pair. If this situation continues long enough, a noncodeword will appear. Since, clear or preset is connected to a large number of flip-flops in a circuit, the probability of instantaneous error indication is high. The last possible connection is that both PR, PR, and CLR, CLR, are under the control of the input data and the load signal. In this case, the noncodeword data inputs (00 or 11) will make PR, PR, CLR, CLR, 0000 or 1111. When PR, PR, CLR, CLR, 0000, the output becomes 00 and when it is 1111 the output becomes 11. These are error indications.

3.3 Simple application (shift register)
The self checking shift register circuit with the DD,-FFs and four to one multiplexers is shown in Fig. 3. The combinational part of the circuit (multiplexers), attached to DD,-FFs, can be designed in different ways. The design method of combinational part makes the circuit to be TSC/CD or SFS/SCD. The different mode of shift register operation can be chosen by S0, S1, S1, as given in Table 3.

| Table 3: TSC register operation due to input control signals |
|-------------|--------|--------|--------|--------|
| S0, S1, S1, | 0101   | 0110   | 1001   | 1010   |
| Action      | no change | shift left | shift right | parallel load |

3.4 Hardware cost
In any fault-tolerant method, there are some extra bits for coding and most synchronous circuit use two rail code. Two other methods are currently proposed in the literature, one the duplication method, the other by Nanya in [5]. In our method, for every pair of complementary data, only one flip-flop is used. The duplicate circuit of edge triggered D flip-flops for a pair of two rail code is shown in Fig. 4. If CK, CK, in Fig. 4 becomes stuck-at 00 or 11, it will not produce noncodeword output. Therefore, the duplicate D flip-flops is not self testing for clock input lines. Nanya has proposed a circuit to make the register load signal self testing, which is connected to the clock input of the memory [5]. This is an edge-triggered D flip-flop with self testing load signal which can be used for two-rail clock. But, for multiple unidirectional stuck-at faults in ‘CP’ lines of register file with bit slice circuit in Fig. 5, the circuit may not produce noncodeword outputs. Therefore, the register file with bit slice circuit in Fig. 5 is not self testing for clock input lines. If an 8-bit of data is coded with Berger code which has the least number of check bits as used in [5], the codewords are
12 bits. Hardware cost comparison of an eight data bit edge triggered self checking register file without proposed DDn-FFs and other circuits is shown in Table 4. Notice that the code translation from two rail to Berger and vice versa is also added to the cost of method in [5].

4 TTn-FF

4.1 Equations and circuits
Internal next state and output equations of TTn-FF, circuit diagram and its symbolic representation are given in Fig. 6. It is contrasted to the previous circuit proposed in [7] in Figs. 7 and 8.

In the equations of Fig. 6, the complementary parts are labelled with an extra n (i.e. $C = \text{clock}$ and $C_n = C'$, $T_n = T$). Also ‘X, Xn’ and ‘Q, Qn’ are complementary pairs in a fault-free operation with input codewords.

In these equations for internal states and outputs, the min-terms can be separated in two parts, as the operating and latching parts.

For X and Xn, the CX and C.Xn are the latching part, because they keep the value of X and Xn constant when $C = 1 (C_n = 0)$. In the same way, $C_n Q$ and $C_n Q_n$...
Table 4: Hardware cost comparison of an eight data bit edge-triggered self checking register file with our proposed DD\(_n\)-FFs and other circuits

<table>
<thead>
<tr>
<th>Bit slice units</th>
<th>Coding</th>
<th>Number of units</th>
<th>CMOS trans. count</th>
<th>Extra cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate D flip-flops (Fig. 4)</td>
<td>2 rail</td>
<td>8</td>
<td>464</td>
<td>30%</td>
</tr>
<tr>
<td>Self testing load flip-flop (Fig. 5)</td>
<td>Berger</td>
<td>12</td>
<td>480</td>
<td>33%</td>
</tr>
<tr>
<td>Proposed DD(_n)-FF (Fig. 1)</td>
<td>2 rail</td>
<td>8</td>
<td>320</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6  Equations, circuit diagram, clock and symbolic representation of TT\(_n\)-FF (error indicator)

Fig. 7  Error indication circuit diagram
keep the value of \( Q \) and \( Q_n \) constant when \( C_n = 1 \). The values of \( Q \) and \( Q_n \) will change due to values of \( X_n \) and \( X \), respectively, when \( C = 1 \). When \( C_n = 1 \), the values of \( X \) and \( X_n \) change due to \( T \) and \( T_n \). The flip-flop is positive edge triggered, because at this edge, \( X \) and \( X_n \) stop changing (due to inputs) and \( Q \) and \( Q_n \) accept the values of \( X_n \) and \( X \). The \( C_n \) lines can be swapped to make the flip-flop negative edge triggered.

The function table of the \( TT_n \)-FF is shown in Table 5. Sixteen possible conditions are marked with \( t_1, t_2, \ldots, t_{16} \). To check any possibility of stuck-at fault, the input lines of AND gates are marked with a number. Every possibility of a stuck-at fault has been checked like \( DD_n \)-FF. In Table 6, both stuck-at faults for each line are checked to produce the noncode word output with one of input codeword conditions \( (t_1, t_2, t_3, t_4) \).

### 4.2 Properties

For the \( TT_n \)-FF, four possible cases can occur. One, if \( TT_n = 01 \), the internal state \( XX_n \) will be \( Q, Q \) during the clock period \( CC_n = 01 \). Thus \( QQ_n \) remains unchanged. Two, when \( TT_n = 10 \), the internal state \( XX_n \) will be \( QQ_n \) during the clock period \( CC_n = 01 \). \( QQ_n \) is then equal to \( XX_n \) when \( CC_n = 10 \). In this case the outputs have swapped values. Cases three and four cases where \( TT_n \) are noncodeword values (i.e. operation is under error conditions). When \( TT_n \) is 11 or 00, \( XX_n \) will be the same as the input. Under this error condition, \( QQ_n \) will have the same noncodeword value of 11 or 00 after the clock edge. The output changes can occur in the circuit when the input exists before the clock edge and remains stable during the clock transient states. The nonsynchronised input changes (during clock edge) produces an unknown output in an edge triggered synchronous circuit and must be avoided.

After an occurrence of an error, the output will not revert back to a code word until the next presetting of the circuit. When an error has occurred (i.e. \( QQ_n = 11 \) or 00), if \( QQ_n = 00 \), no input (i.e. \( TT_n = 10, 01 \)) can change its value. If \( QQ_n = 10 \), \( TT_n = 01 \) can change the output \( QQ_n \) to 00, which is still an error condition.

### 4.3 Clock considerations

Errors in the clock inputs have also been taken into consideration.

As can be seen from Fig. 6, \( QQ_n \) will be 00 when \( CC_n = 00 \). Since \( QQ_n = 00 \) is a noncodeword, an error has been indicated.

When \( CC_n = 11 \), if \( TT_n = 01 \), the values of \( XX_n \) and \( QQ_n \) will not change from their respective previous values. However, when \( TT_n = 10 \), both \( XX_n \) and \( QQ_n \) will become 11 which, being a noncodeword, will indicate an occurrence of an error.

As explained for \( DD_n \)-FF, \( CC_n \) has a transient state of 11 when changing from 01 to 10 by passing the original clock through a NAND latch circuit. The \( CC_n \) transient 11 value occurs for one gate delay, which is less than (minimum) two level logic circuit of \( XX_n \) or \( QQ_n \). The \( CC_n \) is 10 before \( QQ_n \) changes can have feedback effect. Therefore, the \( CC_n \) transient state cannot produce a noncodeword when \( TT_n = 10 \).

The changes of \( TT_n \) should be avoided during clock transition time. Such a change may produce an unknown output. This problem also exists for other fault tolerant design methods [7].

### 4.4 Preset (PR, PR_n) and clear (CLR, CLR_n) circuit

The circuit of the \( TT_n \)-FF with the complementary PR, PR_n and CLR, CLR_n is shown by dashed lines in the \( TT_n \)-FF circuit in Fig. 6.

The circuit of the \( TT_n \)-FF in Fig. 6 is made with AND-OR gates. Therefore, their PR, PR_n and CLR, CLR_n lines are activated by 10 value. These clear and preset lines also exhibit the TSC/CD property. These lines are used for normal operation and presetting of the flip-flop circuits. The clear and preset lines restore the circuit to an error-free state when an error occurs. The \( TT_n \)-FF circuit can be made with NAND gates as given in [10].

Any incorrect stuck-at 0 or 1 can create a 00 or 11 combination for the CLR, CLR_n or PR, PR_n. These noncodeword combination latch one of the \( X, X_n, Q \) or \( Q_n \) to a fixed value (0 or 1). This fixed value will result in an error condition. Thus the TSC property is satisfied.

### Table 5: Function table for \( TT_n \)-FF

<table>
<thead>
<tr>
<th>( QQ_n(t) )</th>
<th>( TT_n = 01 )</th>
<th>( TT_n = 10 )</th>
<th>( TT_n = 00 )</th>
<th>( TT_n = 11 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>01 (t_1)</td>
<td>10 (t_2)</td>
<td>00 (t_3)</td>
<td>11 (t_12)</td>
</tr>
<tr>
<td>10</td>
<td>10 (t_3)</td>
<td>01 (t_2)</td>
<td>00 (t_13)</td>
<td>11 (t_14)</td>
</tr>
<tr>
<td>00</td>
<td>00 (t_2)</td>
<td>00 (t_14)</td>
<td>00 (t_11)</td>
<td>00 (t_15)</td>
</tr>
<tr>
<td>11</td>
<td>11 (t_4)</td>
<td>11 (t_4)</td>
<td>00 (t_12)</td>
<td>11 (t_16)</td>
</tr>
</tbody>
</table>

This property of \( TT_n \)-FF can be used as an error indicator. This also keeps the error to be propagated through the system. It is useful in a system that an 'error blocking path', as mentioned in [13], exists.

The function table of the \( TT_n \)-FF is shown in Table 5. Sixteen possible conditions are marked with \( t_1, t_2, \ldots, t_{16} \). To check any possibility of stuck-at fault, the input lines of AND gates are marked with a number. Every possibility of a stuck-at fault has been checked like \( DD_n \)-FF. In Table 6, both stuck-at faults for each line are checked to produce the noncode word output with one of input codeword conditions \( (t_1, t_2, t_3, t_4) \).

### Table 6: Test conditions for stuck-at fault in \( TT_n \)-FF

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Test condition for ( TT_n )-FF</th>
<th>Noncode ( QQ_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>var.</td>
<td>( T_n )</td>
<td>( Q_n )</td>
</tr>
<tr>
<td>s-0</td>
<td>( t_1 )</td>
<td>( t_2 )</td>
</tr>
<tr>
<td>s-1</td>
<td>( t_2 )</td>
<td>( t_3 )</td>
</tr>
</tbody>
</table>

For the CD property different situations should be considered. If these clear and preset lines are not used (PR\textsubscript{PR} = 10, CLR\textsubscript{CLR} = 10) and they are connected to a logical level, only the stuck-at error can occur. The other case is where only one of these two pairs is connected to a constant value (01) and the other is under control of input data. In this case, if the controlled lines have a noncodeword (00 or 11), one of Q or \overline{Q}, values will latch to 0 or 1 without affecting the value of complementary pair. If this situation continues long enough, a noncodeword will appear. Since clear or preset is connected to a large number of flip-flops in a circuit, the probability of instantaneous error indication will be high. The last possible case is when both PR, PR, and CLR, CLR, are under control of one pair of input data. In this case, the noncodeword data inputs (00 or 11) will make PR, PR, CLR, CLR, = 0000 or 1111. When PR, PR, CLR, CLR, = 0000 the output becomes 00 and, when it is 1111, the output becomes 11. These are error indications.

4.5 Simple application (counter)

TT\textsubscript{T}-FF\textsubscript{x} with TSC/CD combinational circuits are used for the counter. The circuit diagram is shown in Fig. 9.

4.6 Hardware cost

Hardware cost comparison of TT\textsubscript{T}-FF with the modified error indicator circuit in [7] is shown in Table 7. Clear and preset lines are not considered. Notice that the circuit in [7] is only an error indicator, it has extra delay elements and uses system clock which is not self tested.

Table 7: Hardware cost comparison of TT\textsubscript{T}-FF with the error indicator circuit in [7]

<table>
<thead>
<tr>
<th>Circuit</th>
<th>CMOS trans. count</th>
<th>Extra cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error indicator in [7] (Fig. 7 and 8)</td>
<td>136</td>
<td>30%</td>
</tr>
<tr>
<td>Proposed TT\textsubscript{T}-FF (Fig. 6)</td>
<td>56</td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusions

New flip-flops for TSC and SFS synchronous circuit design with the error propagating property are introduced and simple applications are given. The TSC and error propagation of flip-flops and unordered codings of all input lines (data, clock, clear and preset) are included. The method of design and coding is simple, and its modular property gives better opportunity for an automated design. These two new flip-flops are called DD\textsubscript{DD}-FF and TT\textsubscript{T}-FF. The DD\textsubscript{DD}-FF is a TSC/CD D-type flip-flop and is 30% smaller than the conventional methods which were not self testing for clock input lines. The TT\textsubscript{T}-FF is a T-type flip-flop. This flip-flop can also be used as an indicator and as a memory element for TSC/CD or SFS/SCD synchronous circuits.

The cost of proposed circuit is 60% less than a conventional circuit, which is only an error indicator and is not self testing for its clock input lines.

6 References


12 NICOLAIDIS, M., JANSCH, I., and COURTOIS, B.: 'Strongly code disjoint checkers'. Proceedings of the 14th Fault tolerant computing symposium FTCS-14, 1984, pp. 16-21