Novel Architectures for TSC/CD and SFS/SCD Synchronous Controllers

S. M. Kia, S. Parameswaran
Department of Electrical and Computer Engineering
University of Queensland
St. Lucia, Queensland, 4072, Australia

Abstract

In this paper, we introduce design models for Totally Self Checking, Code Disjoint (TSC/CD) and Strongly Fault Secure, Strongly Code Disjoint (SFS/SCD) synchronous controllers. The TSC/CD and SFS/SCD models based on two new proposed low-cost, modular, Totally Self Checking (TSC), edge triggered and error propagating (code disjoint) Flip-Flops; one, a D Flip-Flop which can be 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 retains the error as an indicator until the next presetting, as an aid to error propagation.

Key words: Totally self checking circuits, Strongly fault-secure circuits, Flip-Flops, Synchronous circuits, Code disjoint, Error propagation.

1 Introduction

Increasing circuit complexity makes the determination of errors in a circuit arduous. Totally self checking circuits have been proposed by [1] which use input output coding to determine whether a circuit is operating accurately. In order to apply TSC techniques to system level, code disjoint (CD) property must be considered. However it is hard to devise circuits that are both TSC and CD.

Strongly fault secure circuit is a circuit which 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 then 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 which are TSC or SFS. Techniques have been presented for the creation of TSC[3], SFS[4]and weakly code disjoint[2], SFS/CD (modified from [4] by [2]) and SFS/SCD [2] combinatorial circuits for a class of unordered codes.

As far as synchronous designs are concerned, there have been so far no circuit which has exhibited the properties which satisfy the true TSC goals. The synchronous designs which appear in the literature [5] and [6], add some extra output circuitry to the ordinary FFs, but do not check the inner properties of the FFs. Furthermore these circuits in [5] did not have preset and clear in their circuits. For synchronous circuits, which need FFs with clock, preset and clear, there was so far, no single-chip FF circuit that is TSC and has error propagating properties which has been proposed.

We have introduced two kinds of these FFs, $DD_n - FF$ [7] and $TT_n - FF$ [8] The $DD_n - FF$ is an edge triggered D-FF with the TSC property which is error propagating. $TT_n - FF$ is a T-FF, but retains any error until the next presetting. The $TT_n - FF$ is thus an error indicator circuit. These FFs use two rail code for the input output coding.

For a pair of two-rail code, one $DD_n - FF$ or one $TT_n - FF$ is used. The proposed Flip-Flops (DD_n and TT_n), approximately halve the size of memory for TSC and SFS synchronous circuits proposed in [6] and [5].

In this paper, we have presented the models for TSC/CD and SFS/SCD synchronous controllers, applying $DD_n - FF$s and $TT_n - FF$s. Because of the modularity, the Flip-Flops are suitable for an automated environment of self checking VLSI circuit design.

In this paper, in section 2, the main definitions and notations are given. In section 3, the $DD_n - FF$ and $TT_n - FF$ specifications are reviewed. In section 4, the TSC/CD and SFS/SCD synchronous system models are given and details are given through an example. Sections 5, concludes this paper.

2 Definitions

1. 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].

2. 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].

3. A circuit is Totally Self-Checking (TSC) if, for every fault from a prescribed set, the circuit is both ST and FS [1].

4. A circuit is Code Disjoint (CD) if it always maps codeword inputs to codeword outputs and noncodeword inputs to noncodeword outputs [1].

0-8186-5440-6/94 $03.00 © 1994 IEEE

5. 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 {f}, or the circuit is FS for {f} 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].

6. A circuit is Strongly Code Disjoint(SCD) for every fault set F iff, for every fault f in F, either

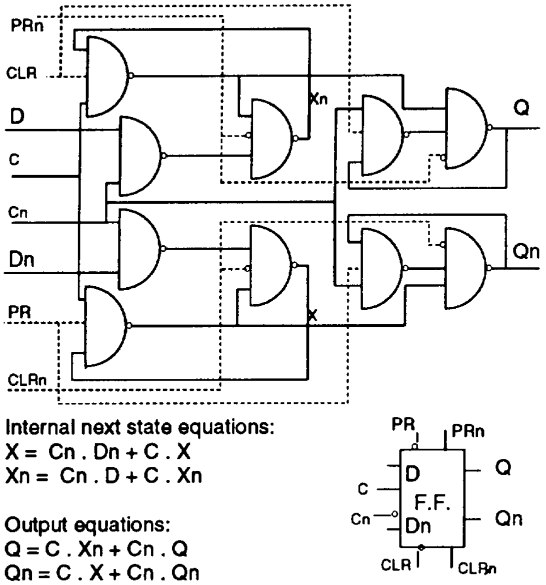
   - the circuit is CD, and is ST for f, or

   - the circuit is CD, and if fault f occurs, the resultant circuit is still SCD for F - f [9].

3 $DD_n - FF$ and $TT_n - FF$ circuits and properties

This section gives two new Flip-Flops which are completely TSC and CD. One is a D type Flip-Flop named DD_n Flip-Flop and the other a T type Flip-Flop called TT_n Flip-Flop. Each Flip-Flop's input is a complimentary pair of a two rail code, thus the name DD_n[1] or TT_n is used.

Internal next state equations, output equations, circuit diagram and symbolic representation of $DD_n - FF$ and $TT_n - FF$ are shown in Figure 1 and Figure 2.

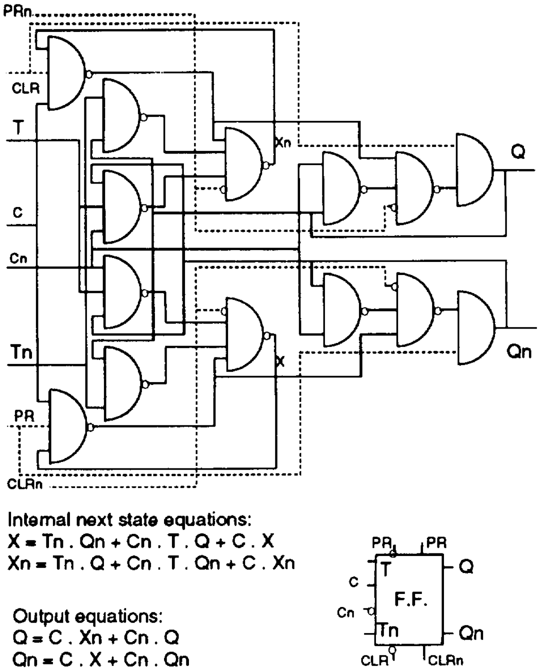


Internal next state equations:
X = Cn . Dn + C . X
Xn = Cn . D + C . Xn

Output equations:
Q = C . Xn + Cn . Q
Qn = C . X + Cn . Qn

Figure 1: The equations, circuit diagram and symbolic representation of a $DD_n - FF$

$'T,T_n'$, $'D,D_n'$, $'X,X_n'$, $'Q,Q_n'$, $'C,C_n'$, $'PR,PR_n'$ and $'CLR,CLR_n'$ are complementary pairs in fault free operation with correct input codewords. CC_n pair represents the clock in the circuit. QQn represents the output pair and XX_n denotes the internal state pair of each Flip-Flop. Two sets of equations are given: the first for the internal state; and second the output. In each set of equations for the internal state pair and the output pair, the minterms are in two parts, dubbed the operating and latching parts respectively. For the internal state pair XX_n, $C.X$ and $C.X_n$ are the latching part, because they retain the value of X and X_n constant when $C = 1(C_n = 0)$. Similarly $C_n.Q$ and $C_n.Q_n$ retains the value of Q and Q_n constant when $C_n = 1(C = 0)$. The values of Q and Q_n will change due to values of X_n and X respectively, when C becomes 1 ($C_n = 0$). When $C_n = 1(C = 0)$, the values of X and X_n change due to inputs (DD_n or TT_n in the $DD_n - FF$ or $TT_n - FF$). Therefore, the FFs are positive edge triggered, since at this edge, X and X stop changing (due to inputs) and Q and Q_n accept the values of X_n and X.



Internal next state equations:
X = Tn . Qn + Cn . T . Q + C . X
Xn = Tn . Q + Cn . T . Qn + C . Xn

Output equations:
Q = C . Xn + Cn . Q
Qn = C . X + Cn . Qn

Figure 2: The equations, circuit diagram and symbolic representation of a $TT_n - FF$

3.1 Clock considerations

Errors in the clock inputs have also been taken consideration.

As can be seen from Figure 1 and Figure 2, C will be 00 when CC_n=00 for both $DD_n - FF$ $TT_n - FF$. Since QQ_n = 00 is a noncodeword error has been indicated.

When CC_n = 11 (for $DD_n - FF$), if DD_n = then the values of XX_n and QQ_n will not change

[1] DD_n stands for the active high line (D) and the complimentary line(D_n)

their respective previous values. However, when $DD_n = Q_nQ$ both XX_n and QQ_n will become 11 which being a noncodeword will indicate an occurrence of an error.

For normal operation of the clock circuit, CC_n has to pass a transient state of 00 or 11 when changing 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 small delay can be achieved by passing the original clock through a Nand latch circuit. The changes of inputs, DD_n or TT_n, should be avoided during clock transition time. Such a change may produce an unknown output. Note that this design consideration should also be considered for other fault tolerant design methods ([6]and [5]). The Flip-Flops are positive edge triggered. The CC_n lines can be swapped to make them negative edge triggered.

3.2 Preset and clear

The complementary PR, PR_n and CLR, CLR_n are shown with dash lines. The circuits of both $DD_n - FF$ and $TT_n - FF$ are formed with Nand gates to simplify the design in VLSI circuits. Thus, PR, PR_n and CLR, CLR_n lines are activated by 01 value. 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. The clear and preset lines are necessary when the Flip-Flops (especially $TT_n - FF$) go to a noncodeword state. Either the preset or clear line pair will restore the circuit to an error free state.

The TSC property of CLR, CLR_n, PR and PR_n line pairs is obvious, since any incorrect stuck-at 0 or 1 fault will create a 00 or 11 combination for the CLR, CLR_n or PR, PR_n. These noncodeword combinations latch one of the X, X_n, Q or Q_n to a fixed value (0 or 1) which will eventually result in an error condition.

3.3 Input equations

Both $DD_n - FF$ and $TT_n - FF$ Flip-Flops are TSC/CD. Two rail code are used for every input and output line of the FFs.

The $DD_n - FF$s and $TT_n - FF$ are used as the state register of the proposed models. The excitation equations of $DD_n - FF$ for different D_j and D_{jn} are directly derived from the next state table with two rail codes. For $TT_n - FF$, the table should change to TT_n values due to next desired state as follows:

1. If the codeword values of Y_j, Y_{jn}[2]for next state are not changed, then T_j, T_{jn} = 01.
2. If the codeword values of Y_j, Y_{jn} for next state are changed to another codeword, then T_j, T_{jn} = 10.
3. If the values of Y_j, Y_{jn} for next states are changed to 00 noncodeword, then T_j, T_{jn} = 00.
4. If the codeword values of Y_j, Y_{jn} for next state are changed to 11 noncodeword, then T_j, T_{jn} = 11.
5. The noncodeword values cannot be changed to codeword with TT_n.

State	Inputs=$I_1I_{1n}I_2I_{2n}$			
$Y_1Y_{1n}Y_2Y_{2n}$	0101	0110	1010	1001
A 0101	0101	0101	0110	0110
B 0110	0110	1010	1010	0110
C 1010	0101	1010	1010	1010

Table 1: Next state function of S (excitation table for $DD_n - FF$s)

4 Synchronous controller

The general form of TSC and SFS/SCD synchronous machines (controllers) with $DD_n - FF$s or $TT_n - FF$s are shown in Figure 3. The $DD_n - FF$ or $TT_n - FF$ are used as elements of the state register.

As in Figure 3 (a), the general case of the TSC controller is designed with input logic (I.L.) followed by a state register which is followed by output logic (O.L.) the input signals are also fed to the output logic directly.

Since the input signals are also fed into the output logic, the TSC circuit of the output logic needs to be CD. However, if the CD property cannot be satisfied by the output logic, then a checker must be added for inputs and state variables.

The SFS/SCD controller of Figure 3 (b) is also formed by input logic, output logic, state register and two rail checker. Both input logic and output logic share a large number of product terms. In the general case of Mealy model machine, only the output logic needs to be SCD. In the case of a Moore model the outputs are dependent only on the outputs of the Flip-Flops. Therefore SCD property of the inputs must be satisfied by the input logic.

The TSC/CD Flip-Flops are used for both TSC and SFS/SCD synchronous circuits. Due to the definitions, it is clear that the SFS/SCD circuits will satisfy the TSC/CD property after a certain number of faults. This property allows us to have $DD_n - FF$s or $TT_n - FF$ inside a SFS/SCD circuit. Any fault of the $DD_n - FF$s or $TT_n - FF$s are tested by input codewords produced by changing of a pair of input data and clocking. Some practical points are mentioned in the following example.

4.1 Example

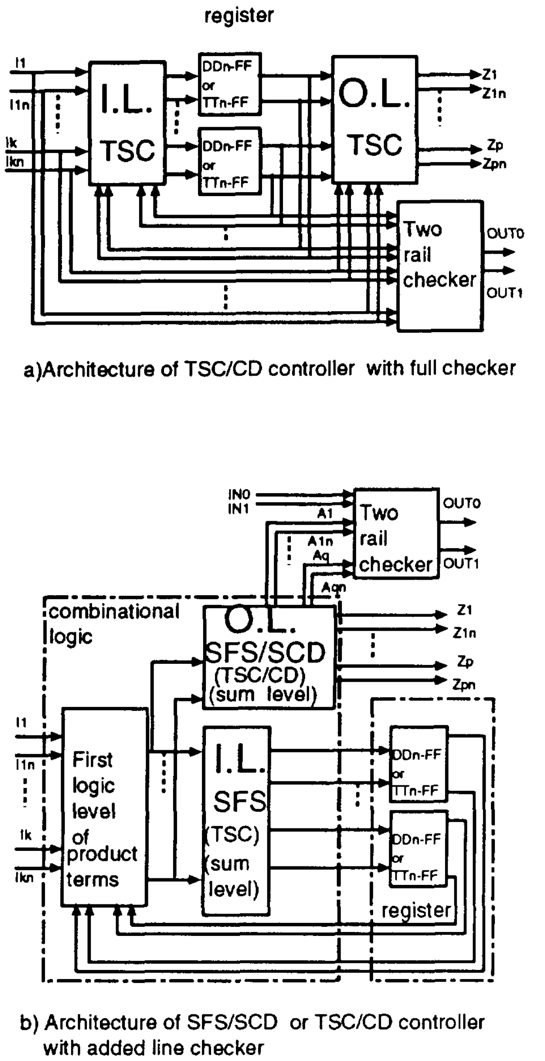
Consider the state flow diagram of a synchronous machine S which is shown in Figure 4.

For state assignment, two pairs of Y_1, Y_{1n} and Y_2, Y_{2n} are used. The next state table is shown in Table 1.

The monotone function for $DD_n - FF$ excitation circuit can be derived directly from Table 1. The excitation functions for $TT_n - FF$s can be derived from Table 2 (while the next state table and the excitation table for the $DD_n - FF$ is the same, it is not the case when considering $TT_n - FF$s).

The monotone functions for $TT_n - FF$s excitation circuit can be written from Table 2. For the TSC design, any set of excitation monotone functions can be chosen. These equations are simplified by the procedure given in [3].

[2] Y_j and Y_{jn} represent the input equations of the $DD_n - FF$s and $TT_n - FF$s which generate the next state values.



a)Architecture of TSC/CD controller with full checker

b) Architecture of SFS/SCD or TSC/CD controller with added line checker

Figure 3: Structure of a)full checker b)added line checker, TSC/CD and SFS/SCD controller with $DD_n - FF$s or $TT_n - FF$s

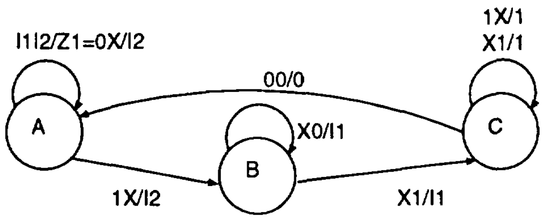


Figure 4: The state/output flow diagram of synchronous machine S

State		Inputs=$I_1 I_{1n} I_2 I_{2n}$			
$Y_1 Y_{1n} Y_2 Y_{2n}$		0101	0110	1010	1001
A	0101	0101	0101	0110	0110
B	0110	0101	1001	1001	0101
C	1010	1010	0101	0101	0101

Table 2: Excitation table of S for $TT_n - FF$s

Considering the $DD_n - FF$s monotone functions:
$D_1 = Y_{1n}Y_2I_{1n}I_2 + Y_1Y_2I_{1n}I_2 + Y_{1n}Y_2I_1I_2 + Y_1Y_2I_1I_2 + Y_1Y_2I_1I_{2n}$
$D_{1n} = Y_{1n}Y_{2n}I_{1n}I_{2n} + Y_{1n}Y_2I_{1n}I_{2n} + Y_1Y_2I_{1n}I_{2n} + Y_{1n}Y_{2n}I_{1n}I_2 + Y_{1n}Y_{2n}I_1I_2 + Y_{1n}Y_{2n}I_1I_{2n} + Y_{1n}Y_2I_1I_{2n}$
$D_2 = Y_{1n}Y_2I_{1n}I_{2n} + Y_{1n}Y_2I_{1n}I_2 + Y_1Y_2I_{1n}I_2 + Y_{1n}Y_{2n}I_1I_2 + Y_1Y_2I_1I_2 + Y_{1n}Y_2I_1I_2 + Y_{1n}Y_{2n}I_1I_{2n} + Y_{1n}Y_2I_1I_{2n} + Y_1Y_2I_1I_{2n}$
$D_{2n} = Y_{1n}Y_{2n}I_{1n}I_{2n} + Y_1Y_2I_{1n}I_{2n} + Y_{1n}Y_{2n}I_{1n}I_2.$

After simplifying of above functions by the method of [3], the following ST functions are found which used for the architecture of Figure 3(a).
$D_1(ST) = Y_2I_2 + Y_1I_1$
$D_{1n}(ST) = I_{1n}I_{2n} + Y_{2n} + Y_{1n}I_{2n}$
$D_2(ST) = Y_2Y_{1n} + I_2Y_2 + I_1$
$D_{2n}(ST) = Y_{2n}I_{1n} + Y_1I_{1n}I_{2n}$

For output logic the monotone functions for output is found from Table 3.
$Z = Y_{1n}Y_{2n}I_{1n}I_2 + Y_1Y_2I_{1n}I_2 + Y_{1n}Y_{2n}I_1I_2 + Y_{1n}Y_2I_1I_2 + Y_1Y_2I_1I_2 + Y_{1n}Y_2I_1I_{2n} + Y_1Y_2I_1I_{2n}$
$Z_n = Y_{1n}Y_{2n}I_{1n}I_{2n} + Y_{1n}Y_2I_{1n}I_{2n} + Y_1Y_2I_{1n}I_{2n} + Y_{1n}Y_2I_{1n}I_2 + Y_{1n}Y_{2n}I_1I_{2n}$

The ST function after simplifying [3] will be as follows:
$Z(ST) = Y_1I_2 + Y_{2n}I_2 + Y_2I_1 + I_1I_2$
$Z_n(ST) = Y_{2n}I_{2n} + Y_{1n}Y_2I_{1n} + I_{1n}I_{2n}.$

These functions are TSC but they are not CD for input data and state output of $DD_n - FF$s. Therefore, a two rail checker must be added to the circuit or the circuit be redesigned to make it SFS/SCD.

State		Inputs=$I_1 I_{1n} I_2 I_{2n}$			
$Y_1 Y_{1n} Y_2 Y_{2n}$		0101	0110	1010	1001
A	0101	01	10	10	01
B	0110	01	01	10	10
C	1010	01	10	10	10

Table 3: Output Z, Z_n values due to inputs and states of machine S

In order to have a SFS/SCD controller, the input logic must be SFS. The monotone functions that map unordered input code to unordered output are SFS[2]. So, the monotone functions of last input logic before eliminations are suitable for SFS input logic circuit.

To design the SFS/SCD output logic circuit, the density properties must be considered for code spaces [2]. For two rail, if, all possible code words are used then, the density properties are satisfied. In our example the four possible code words of input are used. For internal states, only three of four codewords are used and the last one is added. These terms are added to satisfy the density property and then outputs are chosen in such a manner that they are suitable for SCD design. If the unexpected codes of states appear then the input logic circuit produces a noncode state (of all 0s) as an error indication. The new table for SFS/SCD will be as Table 4. The output for added term (1001) are left for next step of the design and are denoted by 'd'.

State		Inputs=$I_1 I_{1n} I_2 I_{2n}$			
$Y_1Y_{1n}Y_2Y_{2n}$		0101	0110	1010	1001
A	0101	01	10	10	01
B	0110	01	01	10	10
C	1010	01	10	10	10
.	1001	dd	dd	dd	dd

Table 4: Output table with added code word for SFS/SCD design

The two procedures of SFS/SCD design of [2] (*covering-nc-CD* and *covered-nc-CD*) are identical for two rail codes. However, by following the example, one extra output pair A_1, A_{1n} must be added with different values. The next mapping of outputs are given in Table 5.

State		Inputs=$I_1 I_{1n} I_2 I_{2n}$			
$Y_1Y_{1n}Y_2Y_{2n}$		0101	0110	1010	1001
A	0101	01,10	10,10	10,01	01,01
B	0110	01,01	01,10	10,10	10,01
C	1010	01,10	10,10	10,01	10,10
.	1001	10,dd	01,10	01,01	01,10

Table 5: Output table with the extra added outputs A_1, A_{1n} for SFS/SCD design

There is one output pair left with 'dd' which can be arbitrary set to 01 or 10. The final output table for SFS/SCD design is shown in Table 6.

State		Inputs=$I_1 I_{1n} I_2 I_{2n}$			
$Y_1Y_{1n}Y_2Y_{2n}$		0101	0110	1010	1001
A	0101	01,10	10,10	10,01	01,01
B	0110	01,01	01,10	10,10	10,01
C	1010	01,10	10,10	10,01	10,10
.	1001	10,01	01,10	01,01	01,10

Table 6: Final output table for SFS/SCD design

There are a few points to be noted:

1. The SFS/SCD equations of outputs can be used for the model (b) of Figure 3. The simplification process of TSC method can be applied to these equations and due to definitions the result will be TSC/CD which has the same model as (b) in Figure 3. The TSC/CD equations seem simpler than SFS/SCD equations, but this is not always the case. For example, in terms of circuit hardware cost, especially in Mealy model controllers, the SFS/SCD circuit may give a better solution.
2. The two rail codes need to be used for state assignments. The inputs can use any unordered codes that satisfy the density property [2] (e.g. Berger codes). Both procedures of SFS/SCD design of [2] should be followed if inputs are not two rail coded. If inputs are two rail coded, with a suitable arrangement, the SFS/SCD design of the circuit can be done with a maximum of one added pair. A few number of lines may need to be added for other kind of input codings. Any kind of unordered codes can be used for outputs.
3. As mentioned in the above example, for density property of two rail code state assignments, some assignments are added to the table which do not occur during normal operation of the circuit. These added state assignments can support the code joint property for state assignments before any stuck at fault in their product terms. Because of impossible occurrence condition, the stuck at fault (0)in these product terms remain untested. The untested faulty terms are ineffective for code disjoint property. The $TT_n - FF$s are suitable to retain the non codewords of state assignments as the stuck at faults and SFS or TSC output circuit will pass the stuck at faults to the output. Therefore, the $TT_n - FF$ state registers are used for the state assignments which are not complete and all the 2^n possible assignments for n state variables do not occur. With $TT_n - FF$ state registers, it is not necessary to add extra state assignments to the table. However, the non code inputs are transferred to the outputs or next states, because the code disjoint property exists for the the input codewords and they satisfy the density property [2] by themselves.

4.2 A Large Example

The complexity of SFS/SCD design procedure increases exponentially with the number of inputs and outputs [2]. This exponential complexity favours the breaking up of circuits into smaller sections. CAD tools have been developed for our models whose outputs are equations for input logic and output logic and complete VHDL[3] structural descriptions of the controller. These tools were applied to a controller of a 32 bit processor given in [10]. The next state diagram is derived with 22 states. The inputs are decoded to get 7 input two rail code pairs. Five pairs of two rail codes are used for state assignments. The outputs are

[3] The Very High Speed Integrated Circuits hardware Description Language

decoded to 24 pairs of two rail codes. The next state and output table of the controller has 128 columns and 22 rows.

For TSC/CD design of circuit as a Mealy model, the circuit has about 3100 gates and 5 self checking Flip Flops as state registers.

5 Conclusions

A new model and method of design for TSC/CD and SFS/SCD synchronous controllers was introduced. The method of design and coding is simple, and its modular property gives better opportunity for an automated design. The two rail codes are used which gives a better solution and every single line has its own check bit and can be used alone. The methodology is achieved using the two new low cost $DD_n - FF$s and $TT_n - FF$s.

References

[1] D. A. ANDERSON & G. METZE, "Design of totally self checking check circuits for m-out- of-n codes ," *IEEE Transactions on Computers* C-22 (Mar. 1973), 263–269.

[2] S. R PAGEY, S. D. SHELEKAR & G. VENKATESH, "A method for the design of SFS/SCD circuits for a class of unordered codes ," *Journal of Electronic testing: Theory and Applications* 2 (1991), 261–277 .

[3] M. DIAZ, P. AZEMA & J. M. AYACHE, "Unified design of Self-Checking and Fail-Safe combinational circuits and sequential machines ," *IEEE Transactions on Computers* C-28 (Mar. 1979), 276–281.

[4] J. E. SMITH & G. METZE, "Strongly Fault Secure logic networks ," *IEEE Transactions on Computers* C-27 (June 1978), 491–499.

[5] T. NANYA, "Error Secure/Propagating Concept and its Application to the Design of Strongly Fault-Secure Processors ," *IEEE Transactions on Computers* 37 (Jan. 1988), 14–24.

[6] T. NANYA & T. KAWAMURA, "On error indication for Self-Checking systems ," *IEEE Transactions on Computers* C-36 (Nov. 1987), 1389–1392.

[7] S. M. KIA & S. PARAMESWARAN, "Design of TSC and SFS/SCD Synchronous Circuits with TSC/error propagating Flip-Flops," *Proc. 12th Australian Microelectronics Conference* MICRO'93 (Oct. 1993), 75–80.

[8] S. M. KIA & S. PARAMESWARAN, "Synchronous TSC/CD Error Indicator for Self Checking Systems," *The 1993 Pacific Rim International Symposium on Fault Tolerant Systems* PRFTS'93 (Dec. 1993), 156–160.

[9] M. NICOLAIDIS, I. JANSCH & B. COURTOIS, "Strongly code disjoint checkers," *Proc. of 14th Fault Tolerant Computing Symposium* FTCS-14 (1984), 16–21.

[10] P. J. ASHENDEN, *The VHDL Cookbook*, Internet, Dept.of Computer Science, University of Adelaide, South Australia, 1990.