Particle swarm optimisation driven low cost single event transient fault secured design during architectural synthesis

Anirban Sengupta, Deepak Kachave
Discipline of Computer Science and Engineering, Indian Institute of Technology Indore, Indore, India
E-mail: asengupt@iiit.ac.in

Published in The Journal of Engineering; Received on 5th January 2017; Accepted on 19th April 2017

Abstract: Owing to aggressive shrinking in nanometre scale as well as faster devices, particle strike manifesting itself into transient fault spanning multiple cycle and multiple units will be the centre-focus of application specific datapath generated through high-level synthesis (HLS)/architectural synthesis. Addressing each problem above separately leads to large area/delay overhead; thus tackling both problems concurrently, leads to huge incurred overhead. To tackle this complex problem, this paper proposes a novel low cost particle swarm optimisation driven dual modular redundant (DMR) based HLS methodology for generation of a transient fault secured design secured against its temporal and spatial effects. The authors’ approach provides a low cost optimised fault secured solution through a particle swarm optimisation exploration framework based on user area-delay constraints. Results indicated that proposed approach obtains an area overhead reduction of 34.08% and latency overhead reduction of 5.8% compared with a recent approach.

1 Introduction

Application specific computing hardware can become increasingly vulnerable to threats due to highly energised alpha particle strike [1–3]. Transient faults can be exploited to create security vulnerabilities such as affecting output computational value. More explicitly, standard vulnerability arising from a transient fault results into erroneous output value due to bit flips causing catastrophic consequences on the systems [1, 4–10].

Transient faults are temporary faults that can emanate in the following environments: (a) Space region due to radiation strike [4] (b) Terrestrial regions. Transients fault in terrestrial regions is caused due to ionising radiation resulting from alpha/neutron particles emitted from IC package impurities used in computing systems. These particles are known to be the major contributing factor to the transient on the ground [4]. In IC packages, some trace amounts of uranium and thorium impurities are found which are actually responsible for emitting alpha particle causing transient faults. Transient faults can be single or multi-cycle in nature. The upsurge in density per unit area is adversely impacting the device and general systems reliability.

Moreover, the availability of faster devices is a characteristic of current technologies that induces key apprehensions to the fault detection researchers. This is becoming more common for current technologies as particles with low linear energy transfer (LET) values even yield a transient extending more than the anticipated cycle time of circuits. Thus producing multi-cycle ($k_{m}$-cycle) transient fault (extended interval transient in the range of $\sim 0.01 \text{ ns}$–1 ns indicated in [8]) in a device, technology advancement and LET of particle influence both play a chief role. Furthermore, owing to huge scaling in nanometre technology ensuing in attenuation of geometric measurements of a device, prospect of multi-unit transient fault ($k_{m}$-unit) is greater now (particle strike resulting in nanometre range is available in [7, 11]). This phenomenon is called multi-unit transient fault (MTF) stemming from a single radiation hit. In other words, a particle strike will affect a number of units bi-directionally in a similar fashion causing common-mode transient faults (as evident through literatures [8, 12, 13]). An example is also shown in [12] which show the prospect of a single strike producing bi-directional error by upsetting two units in an identical manner. Present algorithms for physical designs do not take into account the spatial influence of transient fault (i.e. affecting multiple units simultaneously) before determining placement of units. This is because they only consider area, delay and power (not fault security) during determining an optimised placement. Thus a placement algorithm may place two identical units in a fashion that a single strike affects both equivalently, triggering same effect (even simple rotation of two identical units while placement is not effective) [1, 2, 4, 14–17].

However, developing low cost approach for handling both multi-cycle transient and multi-unit transient fault simultaneously is an expensive affair. Handling the above problems at the higher level of design abstraction i.e. behavioural level [during high-level synthesis (HLS)], may incur lower design overhead as well as ensure reliability awareness from the very initial stage of a design flow. Moreover, addressing this problem at the higher abstraction level also provides greater reliability from the beginning of the design flow. Considering worst possible value (strength) of transient fault in temporal domain ($k_{m}$) and spatial domain ($k_{n}$) as design specification/constraint during HLS (at behavioral level) in conjunction with an advanced design space exploration (DSE) framework, enables generation of a low-cost design solution that is concurrently multiple transient and multi-cycle transient fault secured. This paper presents a novel low-cost security to the dual problem (multi-unit transient and multi-cycle transient fault) through HLS [12, 18–23].

1.1 Motivation: low-cost optimised multi-cycle and multi-transient fault security aware HLS

Following are the reasons: (i) First reliability occurring due to single radiation strike is a key concern especially multi-cycle transient and multi-unit transient fault [24, 25], where handling it at higher abstraction level, i.e. during high level synthesis appears beneficial, owing to lesser available details at this level and greater chances of minimising overhead through advanced optimisation heuristics. (ii) Second, due to paradigm shift in the area of HLS from performance/area optimisation to fault security optimisation, automated techniques are required that can perform exploration of a low cost solution. (iii) Finally, handling both multi-cycle transient and multi-unit transient fault is expensive and time consuming and requires imposing multi-layer security constraints which sustain large area and delay overhead that do not abide by the user budget. Therefore, automated low cost economical design solution that satisfies the user budget as well as minimises chip area/delay is proposed in this paper.
2 Contributions of the paper

2.1 Novel contributions of the paper

Following are the novelties of the present paper:

(a) Novel low cost optimised multi-cycle and multi-unit transient fault secured design based on user area-delay constraints during fitness evaluation.

(b) Novel particle swarm optimisation (PSO)-based exploration methodology for high level synthesis for inclusive fitness assessment of a multi-unit and multi-cycle transient fault secured design solution.

(c) The proposed approach obtains an area overhead reduction of 34.08% compared with [26], while simultaneously attaining a latency overhead reduction of 5.8% compared with [26].

2.2 Advancement of proposed work over [26]

(a) The proposed approach presents low cost optimised multi-cycle and multi-unit transient fault security during HLS, while [26] only presents fault security methodology (with no optimisation in the secured design technique).

(b) The proposed work integrates PSO DSE framework with the fault security approach to yield optimal design solution based on user constraints. However, [26] does not consider user constraint during generating optimised fault secured design solution. This resulted in expensive designs where the user cost budget is not met. Further, non-optimised designs generated by [26] incur huge silicon overhead and delay expenditure.

(c) Detailed sensitivity analysis of the PSO parameters on the quality of DSE results and exploration runtime is presented, which was not included in the seminal version.

3 Related prior work

Our work focuses on providing low cost optimised simultaneous fault security at behavioral (architecture) level against temporal and spatial effects of transient faults occurring due to single particle strike through physically aware high level synthesis methodology. The only work on similar topic is provided in authors’ previous work [26] on non-optimised transient fault security during HLS. However there is no effort (zero work) in prior art to provide concurrent low cost optimal solution to multi-cycle transient and multiple transient fault security during HLS (note: the difference between proposed work and [26] is discussed earlier).

3.1 Multi-cycle transient fault security

Multi-cycle transient fault security at behavioral level was handled in [6, 27, 28] through concurrent error detection scheme where replication of control data flow graph (CDFG) operations were performed. Once the scheduling of the dual modular redundant CDFG was generated, then specific hardware allocation rules were imposed to ensure detectability. Sengupta and Sedaghat’s approach [6] presents more comprehensive hardware allocation rules, as it encompasses the condition of minimal hardware availability as well. In other words, Sengupta and Sedaghat’s approach [6] can provide multi-cycle fault secure even with the availability of single hardware type which is not possible in [28]. In [28] minimum two distinctive hardware is obligatory for allocation to sister operations of original and duplicate unit in DMR. This approach [6] provides more flexibility and robustness.

3.2 Multiple transient fault security

Owing to rare occurrence of multiple transient faults in past technologies, it did not receive much attention in the literature. Further, besides the work presented [26], there has been absolutely no effort on handling multi-unit transient fault during HLS. Rusu et al. [11] used simulation-based technique to focus on multiple transient faults where magnitude of transients resulting from single radiation strike was assessed. Further modelling transient fault propagation was performed in [5] when a fault occurred at the gate output within a logic circuit. Thus the aforesaid techniques tackle multiple (or multi-unit) transient fault at lower level.

4 Background and motivation of PSO

PSO is an advanced optimisation technique widely deployed in solving complex optimisation problems across several domains [29–34]. In this paper we have integrated PSO based DSE with HLS-based fault security methodology by mapping its generic procedure to the requirements of the current problem. Generic PSO is a population-based search technique where particles fly through a search space. In PSO, the position of an ith particle is transformed by summing the velocity to the current position [32]:

\[ x_i(t + 1) = x_i(t) + v_i(t + 1) \]  

Updating of the velocity is performed by using the following function:

\[ v_i(t + 1) = \omega v_i(t) + b_1 r_1(x^g_i - x_i(t)) + b_2 r_2(x^b_i - x_i(t)) \]

where \( x_i(t + 1) \) and \( x_i(t) \) are the position (represented through ‘n’ dimensions) of a particle at time ‘t+1’ and ‘t’, respectively; \( v_i(t+1) \) is the velocity at time ‘t+1’ updated per dimension (it reflects the step size in each dimension, i.e. distance covered per unit time in each dimension); \( \omega \) is called the inertia weight, \( b_1 \) is the cognitive learning factor, \( b_2 \) is the social learning factor, \( r_1, r_2 \) are random numbers in the range \([0, 1]\), \( x^g_i \) is the best position of \( i \)th particle with respect to the minimisation problem, \( x^b_i \) is the global best position found so far.

4.1 Motivation of using PSO algorithm as DSE framework

PSO as an exploration backbone is considered more productive than co-existing evolutionary techniques such as genetic algorithm (GA) and Bacterial Foraging Optimisation Algorithm (BFOA) owing to the following reasons:

(i) Literature [33] suggests that clinically pre-tuned PSO framework yields better optimal results and quicker convergence to optimal during DSE compared with evolutionary algorithms such as GA.

(ii) PSO-based DSE ensures channelled searching such as alteration in search path using velocity vector when a particular direction is found un-productive.

(iii) Control parameter such as inertia weight inside PSO assures an optimal balance between exploration-exploitation during searching through linearly decrease within \([0.9–0.1]\).

(iv) Other control parameters such as acceleration coefficient within PSO enables balance between cognitive and social learning [33].

5 Proposed methodology: low cost Mct-Mtf secured physically aware high level synthesis

5.1 Problem formulation

Given a CDFG, explore its design space and find a low cost optimal design solution (\( X \)):

\[ X = \{N(R_1), N(R_2), N(R_3), \ldots, N(R_D)\} \]

that meets user constraints and reduces the overall design cost; where \( N(D) \) is a number of a particular resource type
(such as adder, multiplier etc.). Thus the problem formulation is as follows:

Find: Optimal($N(R_i)$) = $X_i$

With minimum hybrid

$$\text{Cost}(A_{\text{DMR-MFT}}, \text{DMR-MFT})$$

Subjected to: $A_{\text{DMR-MFT}} \leq A_{\text{con}}$ and $L_{\text{DMR-MFT}} \leq L_{\text{con}}$ as well

$k_c$-cycle transient and $k_m$-unit transient fault constraints; where

$A_{\text{DMR-MFT}}$ and $L_{\text{DMR-MFT}}$ are the chip area and latency of a $k_c$-cycle transient and $k_m$-unit transient fault secured DMR design; $A_{\text{con}}$ and $L_{\text{con}}$ are the user specified constraints.

Note: in the proposed approach, solutions which violate user area and latency constraints are discarded during iterative PSO driven exploration. Only the solutions which meet the user constraints are allowed to move forward into next iteration and finally evolve into a low cost solution.

5.2 General description of proposed methodology

The proposed approach shown in Fig. 1 accepts module library, application in the form of CDFG (available as standard benchmarks in [12, 35]), resource constraints ($X_i$), user constraints of area/latency and characteristics of transient fault in temporal (indicated as 'k-cycle') and spatial domain (indicated as 'k-units') as design constraints. The fault security measures for nullifying the spatial and temporal impacts of transient fault are handled exclusive of each other. However, both the algorithms used for nullifying the effects are connected through physical design driven HLS phenomenon. MCT is resolved during one of the high level synthesis steps while MFT is resolved during one of the physical design steps. However, inserting separate security constraints during physical design driven HLS incur massive chip area and schedule delay overhead. Therefore, optimisation needs to be performed to mitigate this problem and yield a low cost design. Our work employs a PSO driven DSE framework to produce a low cost multi-cycle transient and multi-unit transient fault secured design.

The proposed approach comprises of three major processing segments:

(a) Fault security segment: responsible for converting a normal design into a multi-cycle transient and multiple transient fault secured design. This segment further contains dependent blocks (i) $k_c$-cycle transient fault secured block (ii) $k_m$-unit transient fault secured block.

On the basis of the resource constraint (provided as particle position during PSO process) information, scheduling of a DMR design is performed. The DMR design is created by duplicating all the operations of CDFG along with existing original unit. Duplicating operations of CDFG (i.e. duplicate unit) is required for providing detection capability in case transient fault occurs. This is because adding a duplicate unit with 'proposed hardware allocation security rules’ enables to produce distinct output value than its original counterpart. Subsequent comparison of respective output value from original and duplicate units yields a difference in magnitude, thereby providing transient fault security (detection). Scheduling of DMR design is performed by providing greater priority to original operations over duplicate ones, in case of conflicts. Once the DMR scheduling is generated, then it is fed into the proposed $k_c$-cycle transient fault secured block along with temporal strength of transient fault ($k_c$-cycle). The output of the block yields a ‘$k_c$-cycle secured DMR schedule and allocation’. This block independently provides security against multi-cycle transient fault. Next, based on the ‘$k_c$-cycle secured DMR schedule’ generated, a list ‘$L(k]$’ of hardware modules (functional resources, multiplexers/demultiplexers units etc.) is extracted. The hardware list information is fed into proposed $k_m$-unit transient fault secured block along with spatial strength of transient fault ($k_m$-units). The output of the block yields a $k_m$-unit physical level floorplan design. This block independently provides security against multiple transient fault, however the output from first stage is fed as an input into the second stage. The major steps of proposed approach are shown in Fig. 1 while details are highlighted in Fig. 2.

![Proposed methodology for generation of low cost $k_c$-cycle and $k_m$-unit transient fault secured design](image-url)
The details of our output of Information such as latency of DMR schedule (derived from fault secured design produced by the fault security segment. The process of PSO driven exploration operates as follows: firstly, the particles are initialised which after processing by the fault security segment is fed to the fault security segment for subjecting to physical design floorplanning. The primary inputs to the approach including configuration represented as particle position, the details of major PSO steps are described as follows: (i) Assign $\operatorname{opn}(v) \in U^{\text{OG}}$ and $\operatorname{opn}(v') \in U^{\text{DP}}$ to dissimilar resources (hardware units).

(ii) If unavailable, then:

Retain same allocation for $v'$ (as $v$) in $U^{\text{DP}}$ so that:

$$t(v') - t(v) > k_c$$

(iii) If (i) is false, then:

Push $v'$ (and its children) $\in U^{\text{DP}}$ one CS below such that condition (i) becomes true.

Hazards occur between sister operations of original and duplicate units, in case hardware allocation rules (i), (ii) or (iii) of duplicate unit is violated. In other words, TFH occurs when the following condition is satisfied:

$$t(v') - t(v) \leq k_c \quad \text{where} \quad v \in U^{\text{OG}} \quad \text{and} \quad v' \in U^{\text{DP}}$$

In case of above condition, the hazards are resolved by shifting the operation (and its successors) of duplicate unit which is affected in lower CSs such that $t(v') - t(v) > k_c$. Therefore, once the $k_c$-cycle secured DMR schedule and allocation is generated then the outputs are fed into a multi-phase set-up that comprises of comparators (C1, C2 and C3) in the first phase, followed by voter in the second phase. Inspired from [36], this set-up is designed to safeguard against a likely faulty comparator. The possibility of fault arises when the comparator is itself susceptible against particle strike.

5.4 Constructing a $k_m$-unit multiple transient fault secured design

As discussed earlier, once a $k_m$-cycle secured DMR schedule and allocation is obtained, its latency from schedule is derived for transient fault, and library of available hardware are fed into the proposed approach. The process of generating a DMR schedule involves design of original and duplicate unit ($U^{\text{OG}}$ and $U^{\text{DP}}$) concurrently scheduled based on resource constraint using list scheduling algorithm. Once scheduling is complete, the hardware allocation step is executed for operations of both original and duplicate units. The output of this approach yields a ‘$k_m$-cycle secured DMR schedule and allocation’ which is used for fitness assessment through proposed cost function. The hardware allocation rules are as follows [6]:

(i) If $t(v') - t(v) > k_c$:

Set $t(v') = t(v) + k_c$

(ii) If unavailable, then:

Retain same allocation for $v'$ (as $v$) in $U^{\text{DP}}$ so that:

$$t(v') - t(v) < k_c$$

(iii) If (i) is false, then:

Push $v'$ (and its children) $\in U^{\text{DP}}$ one CS below such that condition (i) becomes true.

Hazards occur between sister operations of original and duplicate units, in case hardware allocation rules (i), (ii) or (iii) of duplicate unit is violated. In other words, TFH occurs when the following condition is satisfied:

$$t(v') - t(v) \leq k_c \quad \text{where} \quad v \in U^{\text{OG}} \quad \text{and} \quad v' \in U^{\text{DP}}$$

Push $v'$ (and its children) $\in U^{\text{DP}}$ one CS below such that condition (i) becomes true.

5.4 Constructing a $k_m$-unit multiple transient fault secured design

As discussed earlier, once a $k_m$-cycle secured DMR schedule and allocation is obtained, its latency from schedule is derived for

$$C_f(X_i) = f(t_{\text{DMR}}^{\text{MCT-MFT}}, A_{\text{DMR}}^{\text{MCT-MFT}})$$

where $A_{\text{DMR}}^{\text{MCT-MFT}}$ and $t_{\text{DMR}}^{\text{MCT-MFT}}$ are defined earlier in Section 5.1. The details of our fitness function are discussed in Section 5.6.
cost evaluation. Next, the list of hardware modules are extracted so that it can be fed an input into $k_u$-unit transient fault secured block. The algorithm executed in this block converts a ‘$k$-cycle secured DMR’ into a ‘$k_u$-unit and $k$-cycle secured DMR’. The algorithm is a physical design floorplan which is achieved such that the hardware modules (from the list $L[k]$) are placed without any $k_u$-unit constraint violation, i.e. hardware modules allocated to two sister operations of original and duplicate units are always placed at least $k_u$-units apart. We note here that $k_u$-unit is considered worst case spatial impact of transient fault. This enables two hardware modules of similar operations to never produce identical wrong outputs despite transient fault, as they are not placed within $k_u$-units range. Let us consider that 1 unit = 768 nm (particle strike resulting in nanometre range is available in $[7, 11]$). Note: $k_u = 4$ is only an example value used for explaining proposed algorithm. $k_u$-unit can be any worst case value fed as input for transient fault security which is based on the estimated spatial strength of particle strike.

5.5 PSO driven exploration: major steps

(i) In $k_u$-cycle secured SDFG DMR, a pair of sister operations ($v$ and $v'$) is chosen.
(ii) Corresponding to sister operations chosen in previous step, assign hardware functional modules ($Mv$ and $Mv'$) in DMR SDFG.
(iii) Design a floorplan in such a way so that the sister hardware modules ($Mv$ and $Mv'$) are at least $k_u$ units apart, i.e. $I(Mv') > I(Mv) + k_u$; the starting point of placement of modules $Mv$ and $Mv'$ are $I(Mv')$ and $I(Mv)$ along x-axis or y-axis in a floorplan.
(iv) Steps 2 and 3 is iterated for residual pair of sister operations present in the $k_u$-cycle secured DMR SDFG.

The rules framed safeguards that any sister hardware modules are prohibited to be placed in a floorplan within the vicinity of $k_u$ units. This is owing to the fact if within $k_u$ units both sister hardware modules are placed in a floorplan, then a radiation strike causing multi-unit transient fault may impact both the units equally producing equivalent incorrect outputs. Thus error detection block (or comparator) will not be able to detect fault. The proposed diagonal floorplan growth rules are highlighted in Fig. 3.

5.5.1 Particle encoding: Particle position ‘$X_i$’ of an ‘$i$’th particle for a given CDFG is represented as follows:

$$X_i = (N(R_1)), (N(R_2)), \ldots, (N(R_d)) \ldots (N(R_d))$$

where, $N(R_d)$ = number of occurrences of $d$th resource type.

5.5.2 Calculation of new particle position: New particle position is calculated in this step with initiation of iteration process. In each iteration of PSO-DSE, new resource value of a particle $X_i$ is calculated in $d$th dimension: $R_{d_i}^* = f(V_d^i, R_d)$ which can be expanded as specified in the following equation [32]:

$$R_{d_i}^* = R_{d_i} + V_{d_i}$$

where, $R_{d_i}^*$ = new resource value of particle $X_i$ in $d$th dimension and $R_{d_i}$ = previous resource value of particle $X_i$ in $d$th dimension; $V_{d_i}^*$ = new velocity of particle $X_i$ in $d$th dimension (i.e. step length taken per unit time in $d$th dimension) updated through the following equation:

$$V_{d_i}^* = \omega V_{d_i} + b_1r_1[R_{d_{i_1}} - R_{d_i}] + b_2r_2[R_{d_{i_2}} - R_{d_i}]$$

where, ‘$R_{d_{i_1}}$’ is the resource value of $X_{i_1}$ in $d$th dimension and ‘$R_{d_{i_2}}$’ is the resource value of $X_{i_2}$ in $d$th dimension.

Note: $X_{i_1} = \{R_{i_1}, R_{i_2}, \ldots, R_{D_d}\}$ and $X_{i_2} = \{R_{i_1}, R_{i_2}, \ldots, R_{D_d}\}$

5.5.3 Determination of local and global best position: In iteration 1, current position ($X_i$) assumes the value of local best position for an $i$th particle $X_{i_{lb}}$. This is because in first iteration there is no previous local best position for an $i$th particle. In addition, (8) is used to determine the global best position ($X_{gb}$) of the population:

$$X_{gb} = X_i[Min(C_{lb1}^{k_u}, C_{lb2}^{k_u}, C_{lb3}^{k_u}, \ldots, C_{lbn}^{k_u})]$$

where, $C_{lb1}^{k_u}$ = local best fitness of particle ‘$X_i’ and ‘$X_{gb}$’ designates the global best particle position with minimum cost among all particle positions ($X_1, ..., X_n$) [32].

5.5.4 Terminating criteria: The proposed approach stops should one of following is true: maximum count of iteration has surpassed ($M = 100$) or $S^2$: No enhancement is seen in $R_d$ over ‘$E$’ iteration count. ($E = 10$); $S^2$: all particle’s velocity become zero ($V = 0$).

5.5.5 Cost evaluation: Subsequent to generation of multi-cycle and multi-unit transient fault secured design corresponding to a particle position, the magnitudes of DMR schedule latency and chip

Fig. 3 Proposed rules for $K_u$-secured Floorplan

Fig. 4 Pseudo-code for PSO driven DSE
area are determined and passed into the cost evaluation block. The proposed fitness \( C_f(X_i) \) evaluation function is shown in the following equation:

\[
C_f(X_i) = \phi_1 \frac{A_{DMR}}{A_{max}} - L_{cons} + \phi_2 \frac{A_{DMR}}{A_{max}} - A_{cons} \tag{9}
\]

where, \( \phi_1 = \phi_2 \) are the user quantified weights of latency and floor-plan area, respectively, typically maintained at 0.5 each; \( A_{DMR} \) = latency of \( k \)-cycle transient fault secured DMR scheduled CDFG based on user resource constraint \( X_i \); \( A_{max} \) and \( A_{cons} \) are maximum values of latency and chip area of \( k \)-cycle transient and \( k \)-unit multiple transient fault secured design; \( A_{DMR} \) and \( A_{cons} \) are the user quantities of latency and chip area of \( k \)-unit transient and \( k \)-unit multiple transient fault secured DMR design; \( L_{cons} \) and \( A_{cons} \) have been defined earlier.

### 5.6 Motivational example

As shown in Fig. 5, a two-cycle transient fault is assumed (corresponding to a transient pulse size of 200ps for a particle LET extracted from [8]) for example. Where, one-cycle or control step = 100 ps. It is to be noted that the values used for explanation is only an example and any other value can be used (as shown in results in Section 6, four-cycle transient fault strength is shown). Transient pulse width can range from ~0.01 ns to ~1 ns depending on particle LET [7, 8]. By taking into account the worst case transient magnitude, \( k \)-cycle value can be quantified as input. Then a list \( L[k] \), which contains list hardware modules present in the DMR SDFG, is prepared subsequent to generation of two-cycle transient fault secured schedule (DMR SDFG). For example list of hardware modules for SDFG DMR of Fig. 5 is:

\[
L[k] = \{ \text{Functional units: A1, A2, A3, A4, M1, M2, M3, M4} \} \\
(1:4 \text{ demux: } d3, d4) \\
(1:8 \text{ demux: } d5, d6, d7, d8) \\
(1:16 \text{ demux: } d1, d2) \\
(4:1 \text{ mux: } m03, m13, m04, m14) \\
(8:1 \text{ mux: } m01, m11, m02, m12, m05, m15, m06, m16, m07, m17, m08, m18)
\]

After extracting the list \( L[k] \), a conflict table is generated that indicates the conflicts between all sisters operations in DMR SDFG. Table 1 shows the conflict between sisters operations and their corresponding hardware allocation. Based on the identified conflicts, the \( k_m \)-unit secured design is generated. As discussed in Section 5.2, proposed MTF security algorithm adds converts a \( k \)-cycle transient fault secured design into a \( k_m \)-unit secured design by performing placement of units such that it achieves \( k_m \)-unit security. We note that the geometric dimensions of the units (based on 15 nm NanGate technology library) are highlighted in Table 2.

Fig. 6 shows the two-unit \( k_m = 2 \) transient fault secured valid floorplan generated using the rules stated in Section 5.4 (white space indicates dead space in chip floorplan). For example assuming the strength of the multiple transient fault is \( k_m = 2 \) units (i.e. worst case impact is 1536 nm). In this example, since size of smallest functional module is 768 nm, hence the worst possible impact of MTF is consecutive neighbourhood functional modules) and M1 and M2 is one such conflicting hardware pair due to allocation to sister operations 1 and 1 (shown in Fig. 4). Thus, M1 and M2 should be placed at least \( k_m = 2 \) units apart (i.e. at a gap of 1536 nm) to follow two-unit MTF security. This is because violation of this would impact both neighbouring modules M1 and M2 equally, causing both 1 and 1 to produce identical wrong output. As seen, M1 and M2 have been placed four units apart. Similarly, M1 and M3 and M2 and M3 have conflicts due to allocation to sister operations 3 and 3’ and 5 and 5’ respectively, hence M3 and M4 and A1 and A2 are also placed at least two units apart. However, since M1 and M3 are not conflicting hardware, hence they can be placed as neighbours during floorplan. The corresponding \( k_m \)-unit multiple transient fault secured floorplan is shown in Fig. 6 which is checked for normalisation through slicing floorplan tree and normalised polish expression [37, 38].

---

*Fig. 5 Two-cycle transient fault secured dual modular redundant schedule of DCT*
and also shows the 4 4 indicate that based on the user constraints speci-

(ii) Final design solution in terms of particle position in 2D search

Fig. 6 Two-unit fault secured floorplan based on two-cycle transient fault

(iii) Fitness (cost) value of the final \( k_m \)-unit and \( k_c \)-cycle transient

(iv) Fitness value achieved for the final \( k_m \)-unit and \( k_c \)-cycle transient

(v) Iteration of convergence of final design solution for various

(vi) Average implementation/exploration runtime of final solution

The proposed approach has been realised in java and run on Intel

6 Experimental results

The proposed approach constructs by high level synthesis low cost
designs that are concurrently multi-cycle (\( k_c \)) and multi-unit transient
(\( k_m \)-unit) fault secured. The low cost fault secured design is

tin fault secured design solution for various user constraint values.

As evident in Tables 3 and 4, for numerous user resource con-

6.1 Proposed results with respect to user constraints

As evident in Tables 3 and 4, for numerous user resource con-

Note: due to imposing security constraint of multiple transient faults, area overhead may be likely for the
designs. However, the schedule delay does not change, as MTF

(i) Based on user constraints for chip area and latency, the area of

eveloping rectangle (chip area) and DMR schedule latency of

(ii) Final design solution in terms of particle position in 2D search space for various user constraint values.
Table 3 Results of proposed approach for user constraints towards maximum value (for \( S = 7 \))

| Sl. No. | Benchmark name \[33, 34\] | Constraints | Final solution |
|---------|--------------------------|-------------|----------------|
|         | Chip area, \( \text{nm}^2 \) | Latency, ns | Rectangular (chip) area with dead space, \( \text{nm}^2 \) | DMR schedule latency, ns | Particle position in 2D space | Fitness value |
| 1.      | ARF                       | 748         | 2.304          | 321                       | 1.248               | 2A 2M         | –0.5145       |
| 2.      | BPF                       | 1516        | 2.472          | 945                       | 1.384               | 3A 1M         | –0.4083       |
| 3.      | DCT                       | 1895        | 3.764          | 1260                      | 1.052               | 5A 2M         | –0.5278       |
| 4.      | DWT                       | 748         | 1.376          | 251.25                    | 0.782               | 3A 3M         | –0.5478       |
| 5.      | EWF                       | 1301.25     | 3.424          | 945                       | 1.504               | 4A 1M         | –0.4172       |
| 6.      | FFT                       | 1800.25     | 3.184          | 1023.75                   | 0.856               | 6A 2M         | –0.5812       |
| 7.      | FIR                       | 500.75      | 1.952          | 279                       | 0.048               | 4A 2M         | –0.5466       |

Table 4 Results of proposed approach for user constraints at mid- value (for \( S = 7 \))

| Sl. No. | Benchmark name | Constraints | Final solution |
|---------|----------------|-------------|----------------|
|         | Chip area, \( \text{nm}^2 \) | Latency, ns | Rectangular (chip) area with dead space, \( \text{nm}^2 \) | DMR schedule latency, ns | Particle position in 2D space | Fitness value |
| 1.      | ARF            | 534.5       | 1.512          | 321                       | 1.248               | 2A 2M         | –0.2000       |
| 2.      | BPF            | 1230.5      | 1.693          | 251.25                    | 0.782               | 3A 3M         | –0.1566       |
| 3.      | DCT            | 1301.875    | 2.242          | 1023.75                   | 0.856               | 5A 2M         | –0.1961       |
| 4.      | DWT            | 500         | 1.078          | 251.25                    | 0.782               | 3A 3M         | –0.2738       |
| 5.      | EWF            | 1005        | 2.298          | 945                       | 1.504               | 4A 1M         | –0.1390       |
| 6.      | FFT            | 1254.5      | 1.822          | 1023.75                   | 0.856               | 5A 2M         | –0.2157       |
| 7.      | FIR            | 445.375     | 1.298          | 300                       | 0.910               | 3A 6M         | –0.2227       |

Table 5 Results of proposed approach with respect to fitness value achieved for various swarm size for user constraints at mid-value

| Sl. no. | Benchmark name | Swarm size |
|---------|----------------|------------|
| 3       | ARF            | –0.2000    |
| 2.      | BPF            | –0.1566    |
| 3.      | DCT            | –0.1691    |
| 4.      | DWT            | –0.2738    |
| 5.      | EWF            | –0.1390    |
| 6.      | FFT            | –0.2157    |
| 7.      | FIR            | –0.1906    |

Table 6 Results of proposed approach with respect to iteration of convergence

| Sl. no. | Benchmark name | Iteration convergence |
|---------|----------------|-----------------------|
|          | Swarm size = 3 | Swarm size = 5        | Swarm size = 7 |
| 1.      | ARF            | 11                    | 11           |
| 2.      | BPF            | 13                    | 13           |
| 3.      | DCT            | 11                    | 11           |
| 4.      | DWT            | 13                    | 12           |
| 5.      | EWF            | 12                    | 12           |
| 6.      | FFT            | 11                    | 11           |
| 7.      | FIR            | 15                    | 14           |

6.2 Proposed result in terms of fitness value of final solution with varying swarm size

The results of the proposed approach in terms of fitness (cost) value achieved for the final fault secured design solution with respect to varying swarm sizes is shown in Table 5. As evident from the results, for swarm size = 7, the proposed approach explores the lowest cost (global best) design solution for almost all benchmarks. Therefore the quality of solution found is highest at \( S = 7 \). Only for FIR, the lowest cost solution is explored at \( S = 5 \), as it is a medium size benchmark.

6.3 Proposed result: iteration convergence and execution runtime

The proposed methodology first creates a \( k \)-cycle transient fault secured DMR schedule based on MCT fault security constraint rules. Subsequently, information from this design output is fed into the \( k \)-unit MTF secured block. The respective iteration of convergence where low cost \( k \)-cycle transient and \( k \)-unit multi transient fault secured design is generated for varying swarm sizes is shown in Table 6. As evident from Table 6, it is seen that with increase in swarm size, the iteration of convergence either remains constant or decreases during exploration a low cost optimal solution. Further, the average exploration/implementation runtime to find the final design solution through proposed PSO driven exploration process during physically aware HLS is also reported in Table 7. As seen in Table 7, the proposed approach produces low cost fault secured solutions for small, medium and large
size benchmarks within acceptable runtime. In other words, the implementation runtime of the proposed approach is in few seconds irrespective of the size of the benchmark handled. Therefore our approach is scalable for large problem size.

6.4 Comparison with related work

To the best of the authors’ knowledge, there is no low cost optimised framework that provides simultaneously security during high level synthesis against multi-cycle ($k_1$) and multi-unit ($k_u$) transient fault impacted. We simultaneously achieve this temporal and spatial security through a novel DMR design driven physically high level synthesis. There is no prior art that provides low cost optimised simultaneously security against temporal and spatial security through a novel DMR design driven physically high level synthesis. We therefore compared proposed approach with [26] which does not tackle minimisation of overhead attained due to imposing multi-layered fault security rules and do not yield a low cost economical design solution. Note: The comparison results with un-hardened designs for similar benchmarks have been already discussed in [26], thus have not been included here. Results of comparison with [26] have been presented for two types of user constraints namely ’towards maximum value’ and ’at mid-value’ in Tables 8 and 9 respectively. ‘Towards maximum value’ indicates relaxed constraints, while ’mid-value’ indicates tighter constraints. Results in Table 8, indicate that for all benchmarks, the proposed approach always yields a better quality fault secured solution than [26] for all benchmarks. However, the proposed approach never attains a higher cost solution than [26] for fully relaxed constraints. More specifically, for FIR benchmark, approach [26] yields a fault secured solution with final cost as $-0.200$ while approach [26] attains a highest cost solution with final cost as $-0.215$.

### Table 8 Comparison results of proposed approach with [26] for constraints at max. value ($S = 7$)

| Benchmark name | Resource constraint | Rectangular (chip) area with dead space comparison in sq. unit | Latency comparison in, ns | Fitness comparison |
|----------------|---------------------|-------------------------------------------------------------|---------------------------|--------------------|
|                | [26] | Proposed | Constraints | [26] | Proposed approach | Constraints | [26] | Proposed approach | [26] | Proposed approach |
| BPF            | 2A 2M | 3A 1M | 1516 | 1041 | 945 | 2.472 | 1.240 | 1.384 | $-0.405$ | $-0.408$ |
| DCT            | 4A 2M | 5A 2M | 1895 | 1301.25 | 1260 | 3.764 | 1.052 | 1.052 | $-0.516$ | $-0.527$ |
| DWT            | 2A 3M | 3A 3M | 748.0 | 347.75 | 251.25 | 1.376 | 0.916 | 0.782 | $-0.434$ | $-0.547$ |
| EWF            | 2A 1M | 4A 1M | 1301.25 | 945 | 945 | 3.424 | 1.824 | 1.504 | $-0.370$ | $-0.417$ |
| FFT            | 4A 2M | 6A 2M | 1800.25 | 1301.25 | 1023.75 | 3.184 | 0.856 | 0.856 | $-0.504$ | $-0.581$ |
| FIR            | 4A 4M | 4A 2M | 590.75 | 500 | 279 | 1.952 | 0.714 | 0.848 | $-0.393$ | $-0.546$ |
| ARF            | 2A 2M | 2A 2M | 748 | 321 | 321 | 2.304 | 1.248 | 1.248 | $-0.514$ | $-0.514$ |

### Table 9 Comparison results of proposed approach with [26] for constraints at mid- value ($S = 5$)

| Benchmark name | Resource constraint | Rectangular (chip) area with dead space comparison in sq. unit | Latency comparison in, ns | Fitness comparison |
|----------------|---------------------|-------------------------------------------------------------|---------------------------|--------------------|
|                | [26] | Proposed | Constraints | [26] | Proposed approach | Constraints | [26] | Proposed approach | [26] | Proposed approach |
| BPF            | 2A 2M | 2A 1M | 1230.5 | 1041 | 945 | 1.693 | 1.240 | 1.384 | $-0.154$ | $-0.156$ |
| DCT            | 4A 2M | 3A 2M | 1301.87 | 1260 | 1023.75 | 2.242 | 1.052 | 1.318 | $-0.169$ | $-0.196$ |
| DWT            | 2A 3M | 3A 3M | 500 | 347.75 | 251.25 | 1.078 | 0.916 | 0.782 | $-0.160$ | $-0.273$ |
| EWF            | 2A 1M | 4A 1M | 1005 | 945 | 945 | 2.298 | 1.824 | 1.504 | $-0.092$ | $-0.139$ |
| FFT            | 4A 2M | 5A 2M | 1254.5 | 1181.25 | 1023.75 | 1.822 | 0.856 | 0.856 | $-0.172$ | $-0.215$ |
| FIR            | 4A 4M | 3A 6M | 445.375 | 428.0 | 300 | 1.298 | 0.714 | 0.910 | $-0.164$ | $-0.222$ |
| ARF            | 2A 2M | 2A 2M | 534.5 | 321 | 321 | 1.512 | 1.248 | 1.248 | $-0.200$ | $-0.200$ |

### Table 10 Results of comparison of proposed approach with [26] in terms of design overhead

| Benchmark name | Baseline approach (non-fault secured designs) | Final fault secured design solution | Design overhead w.r.t baseline approach |
|----------------|-----------------------------------------------|-----------------------------------|--------------------------------------|
|                | Chip area, Sq. units | Latency, ns | Chip area, Sq. units | Latency, ns | Chip area, Sq. units | Latency, ns |
|                | [26] | Proposed | [26] | Proposed approach | [26] | Proposed approach |
| BPF            | 710.0 | 0.652 | 1041 | 1.240 | 945.0 | 1.384 | 331 | 0.588 | 235.0 | 0.732 |
| DCT            | 925.5 | 0.590 | 1301.2 | 1.052 | 1260.0 | 1.052 | 375.7 | 0.462 | 334.5 | 0.462 |
| DWT            | 157.2 | 0.714 | 347.7 | 0.916 | 251.2 | 0.782 | 190.5 | 0.202 | 94.0 | 0.068 |
| EWF            | 462.7 | 1.104 | 945 | 1.824 | 945.0 | 1.504 | 482.2 | 0.720 | 482.2 | 0.400 |
| FFT            | 925.5 | 0.460 | 1301.2 | 0.856 | 1023.7 | 0.856 | 375.7 | 0.396 | 98.2 | 0.396 |
| FIR            | 210.0 | 0.580 | 500 | 0.714 | 279.0 | 0.848 | 290.0 | 0.134 | 69.0 | 0.268 |
| ARF            | 218.0 | 0.720 | 321 | 1.248 | 321.0 | 1.248 | 103 | 0.528 | 103.0 | 0.528 |

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/). J Eng, 2017, Vol. 2017, Iss. 6, pp. 184–194 doi: 10.1049/joe.2016.0378
the comparison of design overhead (i.e. latency and chip area) for proposed approach and [26] with baseline approach (non-fault secured design). As evident from Table 10, for almost all benchmarks, the design overhead of proposed approach wrt to baseline (non-fault secured design) is lower than design overhead of [26] w.r.t. to baseline. This is due to lack of optimisation performed in [26] for fault secured designs, resulting in expensive design solutions. Simple calculations revealed that the proposed approach obtains a reduction in area overhead of 34.08% compared with [26], while simultaneously attaining a latency overhead reduction of 5.8% compared with [26]. This analysis provides strong evidence that not only simultaneously resilient design against multi-cycle and multiple transient fault is crucial for future generation of systems, but its low cost model that satisfies user chip and delay budget is equally critical from economic stand point.

6.5 Security evaluation

In a DMR system, hardware assigned to every operation is vulnerable to transient fault. The number of potential transient fault in a DMR system is directly proportional to the number of hardware assignments. For example, the DMR system of DCT application (Fig. 5) has 56 hardware assignments to operations resulting into 56 vulnerable. Similarly, for other benchmark applications, the total security vulnerabilities that are secured through proposed approach are shown in Fig. 7.

7 Conclusion and future work

This paper presented the first approach on developing a low cost concurrent multi-cycle and multi-unit transient fault secured design during physically aware HLS driven by DMR concept. In addition, it proposes the unification process of high level synthesis and physical design floorplanning to leverage the accuracy involved during design point evaluation based on user chip area-delay constraints specified. The future work targets detailed placement and wirelength estimation besides floorplanning during physically aware high level synthesis. Further, we intend to refine floorplan quality by imposing internal optimisation prior to fitness evaluation.

8 Acknowledgment

This work acknowledges the financial support provided by Media Lab Asia, Ministry of Electronics and Information Technology, Government of India.

9 References

[1] Omana M., Rossi D., Metra C.: ‘Latch susceptibility to transient faults and new hardening approach’, IEEE Trans. Comput., 2007, 56, (9), pp. 1255–1268
[2] Omana M., Rossi D., Metra C.: ‘High-performance robust latches’, IEEE Trans. Comput., 2010, 59, (11), pp. 1455–1465
[3] Benini L., Micheli G.D.: ‘The data-link layer in NoC design’, in Benini L., De Micheli G. (Eds.): ‘Networks on Chips’ (Morgan Kaufmann, San Francisco, 2006)
[4] radhome.gsfc.nasa.gov/radhome/sec.htm, 2015
[5] Miskov-Zivanov N.: ‘Multiple transient faults in combinational and sequential circuits: A systematic approach’, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2010, 29, (10), pp. 1614–1627
[6] Sengupta A., Sedaghat R.: ‘Swarms intelligence driven design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis based on user power-delay budget’, J. Microelectron. Reliab., 2015, 55, (6), pp. 990–1004
[7] Kumar S., Agarwal S., Jung J.P., et al.: ‘Soft error issue and importance of low alpha sorders for microelectronics packaging’, Rev. Adv. Mater. Sci, 2013, 34, pp. 185–202
[8] Lisboa C.A., Carro L.: ‘System level approaches for mitigation of long duration transient faults in future technologies’, 12th IEEE European Test Symp. – ETS 2007, 20–24 May 2007, pp. 165–172
[9] Sengupta A.: ‘Exploration of k-cycle transient fault-secured datapath and loop unrolling factor for control data flow graphs during high-level synthesis’, IET Electron. Lett., 2015, 51, (7), pp. 562–564
[10] Sengupta A., Mohanty S.P., Lombardi F., et al.: ‘IEEE access special section editorial: security and reliability aware system design for mobile computing devices’, IEEE Access J., 2016, 4, pp. 2976–2980
[11] Rini C., Bougerol A., Anghel L., et al.: ‘Multiple event transient induced by nuclear reactions in CMOS logic cells’, 13th IEEE Int. On-Line Testing Symp., July 2007, pp. 137–145
[12] Mohanty S.P., Ranaganathan N., Kouignantos E., et al.: ‘Low-power high-level synthesis for nanoscale CMOS circuits’ (Springer, 2008), pp. 5–40
[13] Rossi D., Omana M., Tomo F., et al.: ‘Multiple transient faults in logic: an issue for next generation ICs’, 20th Int. Symp on Defect and Fault Tolerance in VLSI Systems, 2005, pp. 352–360
[14] Lakshminarayana G., Raghunathan A., Jha N.K.: ‘Behavioral synthesis of fault secure controller/datasheets based on aliasing probability analysis’, IEEE Trans. Comput., 2000, 49, pp. 865–885
[15] Lala P.K.: ‘Fault-tolerant and fault-testable hardware design’ (Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1985)
[16] Polian I., Hayes J.P., Kundu S., et al.: ‘Transient fault characterization in dynamic noisy environments’, Test Conf., 2005. Proc. ITC 2005. IEEE Int., November 2005, vol., no., pp. 10–1048
[17] Mukherjee S.S., Emer J., Reinhardt S.K.: ‘The soft error problem: an architectural perspective’. 11th Int. Symp. on High-Performance Computer Architecture, February 2005, pp. 243–247
[18] Zhang M., Mitra S., Mak T.M., et al.: ‘Sequential element design with built-in soft error resilience’, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2006, 14, (2), pp. 1368–1378
[19] Gomaa M., Vijaykumar T.N.: ‘Opportunistic transient-fault detection’. Proc. 32nd Int. Symp. on Computer Architecture, ISCA'05, June 2005
[20] Pignol M.: ‘How to cope with SEU/SET at system level?’, 11th IEEE Int. in On-Line Testing Symp., July 2005, pp. 315–318
[21] Mizan E., Amineur T., Jacome M.F.: ‘Self-imposed temporal redundancy: An efficient technique to enhance the reliability of pipelined functional units’. IEEE 19th Int. Symp. on Computer Architecture and High Performance Computing, 2007, pp. 45–53
[22] Martinez-Alvarez A., Restrepo-Calle F., Cuena-Asensi S., et al.: ‘A hardware-software approach for on-line soft error mitigation in interrupt-driven applications’, IEEE Trans. dependable Secure Comput., 2014, 13, (4), pp. 502–508
[23] Fazeli M., Ahmadian S.N., Miremadi S.G., et al.: ‘Soft error rate estimation of digital circuits in the presence of multiple event transients (METs): Design, Automation and Test in Europe Conf. and Exhibition (DATE), 2011, pp. 1–6
[24] Okommenakos P., Zwolinski M.: ‘An integrated high-level-on-line test synthesis tool’, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2006, 25, (11), pp. 2479–2491
[25] Inoue T., Henri H., Yoshikawa Y., et al.: ‘High-level synthesis for multi-cycle transient fault tolerant datapaths’. Proc. of the 17th IEEE Int. On-Line Testing Symp., 2011, pp. 13–18
[26] Kachave D., Sengupta A.: ‘Integrating physical level design and high level synthesis for simultaneous multi-cycle transient and multiple transient fault resiliency of application specific datapath processors’, J. Microelectron. Reliab., 2016, 60, pp. 141–152
[27] Wu K., Karri R.: ‘Algorithm level recomputing—a register transfer level concurrent error detection technique’. Proc. IEEE/ACM Int. Conf. Computer-Aided Design, November 2001, pp. 537–543
15th Int. Symp. on Quality Electronic Design (ISQED 2014), California, USA, March 2014, pp. 60–67

[33] Trelea I.C.: ‘The particle swarm optimization algorithm: convergence analysis and parameter selection’, Inf. Process. Lett., 2003, 85, (6), pp. 317–325

[34] Engelbrecht A.P.: ‘Fundamental of computational swarm intelligence’ (John Wiley and sons limited, England, 2005)

[35] Express high level synthesis benchmark, University of California Santa Barbara. Available at http://express.ece.ucsb.edu/benchmark/, December 2015

[36] Dubrova E.: ‘Fault-tolerant design’ (Springer, New York, USA, 2013), ISBN 978-1-4614-2112-2

[37] Sait S., Youssef H.: ‘VLSI physical design automation: theory and practice’, (World Scientific Publishing Company, 1999), pp. 91–152

[38] Hu J., Lienig J., Markov I.L., ET AL.: ‘VLSI physical design: from graph partitioning to timing closure’ (Springer, Netherlands, 2011), ISBN 978-90-481-9591-6