MASTER THESIS

Electronic Design Automation Algorithms for Radiation-Hardening Standard Cell Legalisation with Respect to Circuit Performance

Supervisor:
Christos Sotiriou
chsotiriou@e-ce.uth.gr
Committee:
Georgios Stamoulis
georges@e-ce.uth.gr
Fotios Plessas
fplessas@e-ce.uth.gr

Student:
Christos Georgakidis
cgeorgakidis@uth.gr

A thesis submitted in fulfillment of the requirements
for the degree of Master
in the
Circuits & Systems Laboratory (CASlab)
Department of Electrical and Computer Engineering

October 18, 2019

Abstract

Faults caused by ionising radiation have become a significant reliability issue in modern Integrated Circuits (ICs). The solution to this problem is to apply Radiation-Hardening techniques. However, the Radiation-Hardening design flow differs from the standard design flow. Thus, there is not sufficient support from industrial Electronic-Design-Automation (EDA) tools. In current master thesis, we present a Triple-Modular-Redundancy (TMR) Radiation-Hardening approach, based on (i) replacement of Flip-Flops (FFs) to a TMR structure, consisting of a FFs triplet and a majority voter, as well as (ii) a custom Legalisation algorithm, able to satisfy user-specified, minimum distances among the FFs of each triplet. Our Radiation-Hardening Legaliser is fully compatible with existing industrial EDA tools. By ensuring a minimum spacing among triplet FFs of each TMR structure, we reduce the probability of a particle strike affecting more than one triplet instances. Comparing our customised legalisation algorithm with an industrial EDA tool supporting spacing constraints, the latter provides slightly better results. It is worth to mention that the industrial tool failed to terminate successfully for large designs due to lack of enough memory. Finally, we implemented a series of optimisations (MinMax-Bounded, HPWL-driven, Timing-driven). In some cases, these optimisations result in slightly worse results compared to the industrial tool, while in other cases, it provides better results. Thus, our Radiation-Hardening flow can be attractive for reducing radiation faults.

Περίληψη

Σφάλματα που προκαλούνται από ιονίζουσα ραδιενέργεια έχουν καταλήξει να αποτελούν ένα σημαντικό πρόβλημα στα σύγχρονα ολοκληρωμένα κυκλώματα (ICs). Η λύση σε αυτό το πρόβλημα είναι η εφαρμογή Ραδιατιον-Ηαρδενινγ τεχνικών. Ωστόσο, η Radiation-Hardening ροή σχεδίασης ολοκληρωμένων κυκλωμάτων διαφέρει από την συμβατική ροή σχεδίασης. Αυτό έχει ως αποτέλεσμα να μην υπάρχει επαρκής υποστήριξη από τα βιομηχανικά Electronic-Design-Automation (EDA) εργαλεία. Στην παρούσα διπλωματική εργασία παρουσιάζουμε μία Triple-Modular-Redundancy (TMR) Radiation-Hardening προσέγγιση, που βασίζεται (α) στην αντικατάσταση όλων των Flip-Flops (FFs) του κυκλώματος με TMR δομές, αποτελούμενες από μία τριπλέτα από FFs και έναν voter, ο οποίος επιστέφει την πλειοψηφία της τριπλέτας, καθώς επίσης και (β) ενός τροποποιημένου αλγορίθμου εγκυροποίησης κυκλώματος, ο οποίος ικανοποιεί μία ελάχιστη απόσταση μεταξύ των FFs κάθε τριπλέτας, η οποία ορίζεται από το χρήστη. Ο Radiation-Hardening αλγόριθμός μας είναι πλήρως συμβατός με τα υπάρχοντα βιομηχανικά EDA εργαλεία. Εγγυώντας μία ελάχιστη απόσταση μεταξύ των FFs μίας τριπλέτας, μειώνουμε την πιθανότητα ένα φορτισμένο σωματίδιο να επηρεάσει περισσότερα από ένα μέλη της τριπλέτας. Συγκρίνοντας τα αποτελέσματα για Ισχύ, Απόδοση και Εμβαδό του κυκλώματος (PPA) του δικού μας αλγορίθμου και ενός βιομηχανικού EDA εργαλείου, ακολουθώντας την ροή του για την ικανοποίηση περιορισμών απόστασης μεταξύ στοιχείων του κυκλώματος, το τελευταίο οδηγεί σε οριακά καλύτερα αποτελέσματα αλλά είναι συγκρίσιμα με τα αποτελέσματα του δικής μας προσέγγισης, ενώ αξίζει να σημειωθεί ότι το βιομηχανικό εργαλείο απέτυχε να τελειώσει επιτυχώς σε μεγάλα κυκλώματα εξαιτίας της έλλειψης μνήμης. Τέλος, υλοποιήσαμε μία σειρά από βελτιστοποιήσεις (MinMax-Bounded, HPWL-driven, Timing-driven) οδηγώντας είτε σε οριακά χειρότερα αλλά συγκρίσιμα ή ακόμη και καλύτερα αποτελέσματα από το βιομηχανικό εργαλείο, κάνοντας τη Radiation-Hardening ροή που ακολουθήσαμε ελκυστική για την μείωση των σφαλμάτων που προκαλούνται από ακτινοβολία.

Acknowledgements

I would like to thank my thesis supervisor Prof. Christos Sotiriou, who was always available and willing to help whenever I ran into trouble or had a question about my research and guided me in the right direction whenever he thought I needed it. I would also like to thank Prof. Georgios Stamoulis and Prof. Fotios Plessas for the confidence they showed in my face.

I would like also to thank the members of the CASlab, who created a friendly environment, and especially Nikos Sketopoulos and Stavros Simoglou, who helped me in my thesis research.

Finally, I would like to express my acknowledgements to my parents and my friends supporting me throughout my years of study at the University of Thessaly.

Christos Georgakidis
Volos, 2019

Electronic Design Automation Algorithms for Radiation-Hardening Standard Cell Legalisation with Respect to Circuit Performance

Christos Georgakidis
cgeorgakidis@uth.gr

Αλγόριθμοι Ηλεκτρονικού Αυτοματισμού Σχεδίασης Κυκλωμάτων για την Ανθεκτική σε Ακτινοβολία Εγκυροποίηση Στοιχείων Δεδομένου της Απόδοσης του Κυκλώματος

Χρήστος Γεωργακίδης
cgeorgakidis@uth.gr

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Due to technology downscaling, the modern integrated circuits (ICs) have become more susceptible to errors caused by ionising radiation and radiation particle strikes. The effects of these faults can be destructive or not. In the first case, permanent damage is caused to the device, while in the second case, we can notice only a temporary malfunction of the circuit.

To increase the resistance of the circuits to errors caused by radiation, we have to apply a radiation-hardening method. There are various radiation-hardening techniques in the literature. These techniques perform changes either to the fabrication process of the circuit or to the design of the circuit.

The radiation-hardening design flow differs from the standard design flow. Thus, there is not sufficient support from the industrial Electronic-Design-Automation (EDA) tools.

In our thesis, we present a Radiation-Hardening Legalisation Algorithm, utilising the technique of Triple-Modular (TMR) technique. In our approach, we triplicate each sequential element of the circuit and add a voter, which returns the majority of the three instances. Moreover, to reduce the probability of a particle strike affecting more than one member of the same TMR triplet, we apply a spacing constraint among them. A sufficient way to satisfy the spacing constraints is during the placement stage of the ICs design flow, and more specifically during the legalisation process. The most significant benefit of our approach is the freedom of using standard cell libraries instead of radiation-hardened, reducing the cost of fabrication.

In the first approach, our legalisation algorithm is displacement-driven, legalising each cell to its nearest legal position, satisfying at the same time the spacing constraints. Next step was the exploration of the effect of various optimisation techniques to the Power, Performance and Area (PPA) of the circuit. Specifically, we tried the following optimisation:

- Insert a bound avoiding Legaliser to place cells far away from the other members of the TMR triplet
- Place each cell to the position which leads to the minimum total wire length overhead

- Place the cells in a way it improves the timing of the circuit

Finally, to evaluate the quality of work, we compared our radiation-hardening legalisation approach with a flow of an industrial EDA tool, which supports spacing constraints among specified cells.

Chapter 2

Theoretical Background

2.1 Introduction to EDA

Integrated circuits (ICs) have contributed significantly towards the development of all the technological wonders that populate the world today. The ICs have found various applications from cars, televisions, computers, cell phones, music players to ship, aeroplane and spacecraft equipment. Integration density and performance of ICs have gone through an outstanding revolution in the last few decades. According to Moore's law, as shown in Fig. 2.1, integration complexity doubles approximately every 1 to 2 years. The above has led to ICs comprised of hundreds of millions of transistors. The design and the optimisation of ICs are essential to the production of new semiconductor chips. So the design process of very large-scale integrated (VLSI) circuits is highly complex and strongly depends on electronic design automation (EDA) tools.

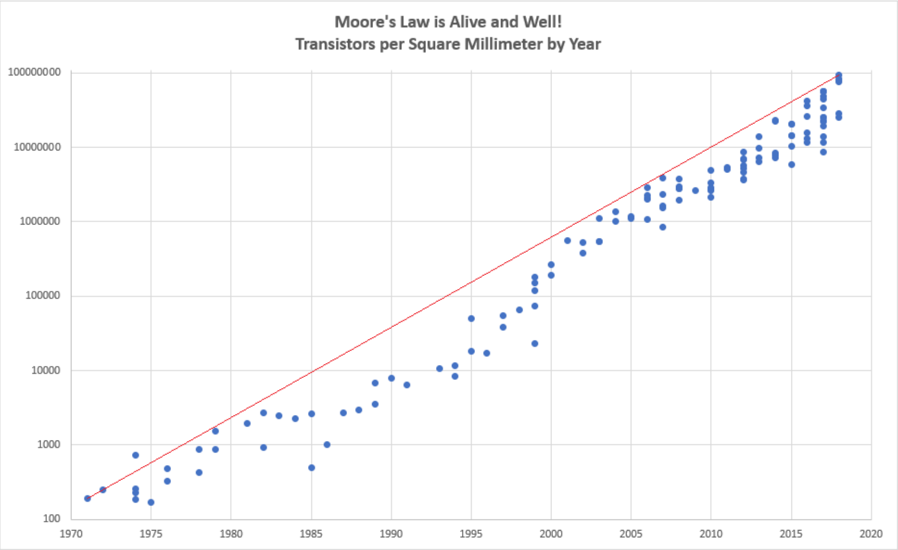


Figure 2.1: Moore's Law from 1970 till 2020

The EDA industry develops such specialised software to support engineers in the creation of new IC designs. The high complexity of modern ICs established EDA almost in all stages of ICs' design flow, from high-level system design to fabrication. That is, EDA tools are used to mostly automate design steps like logic design, simulation, physical design and verification. EDA tools have always targeted on automating the entire design process and linking the various design steps into a complete design flow. However, this integration is challenging since some design steps need additional freedom degrees, while scalability requires tackling some design steps independently. On the other hand, technology downscaling has made the boundaries of the different design steps fuzzy [1].

2.1.1 Placement

Circuit placement is one of the most significant steps of EDA flow. After partitioning the circuit into finer modules and floorplanning the layout to determine block outlines and pin locations, placement consists of assigning a physical location to each standard cell or logic element with each block. The main objective of the placement process is to determine cells locations and orientations within a layout, specifying solution constraints and optimisation goals, such as wire length optimisations, timing optimisation, e.t.c [1]. Placement algorithms treat logic cells as rectangles and, in general, their size is not identical. So, the physical size of each cell must be known so that placement does not overlap cells in the layout. Some standard cell systems support the use of large array macros, such as RAMs. Although, the automated placement of these cells is challenging, and they might have to be placed manually [2].

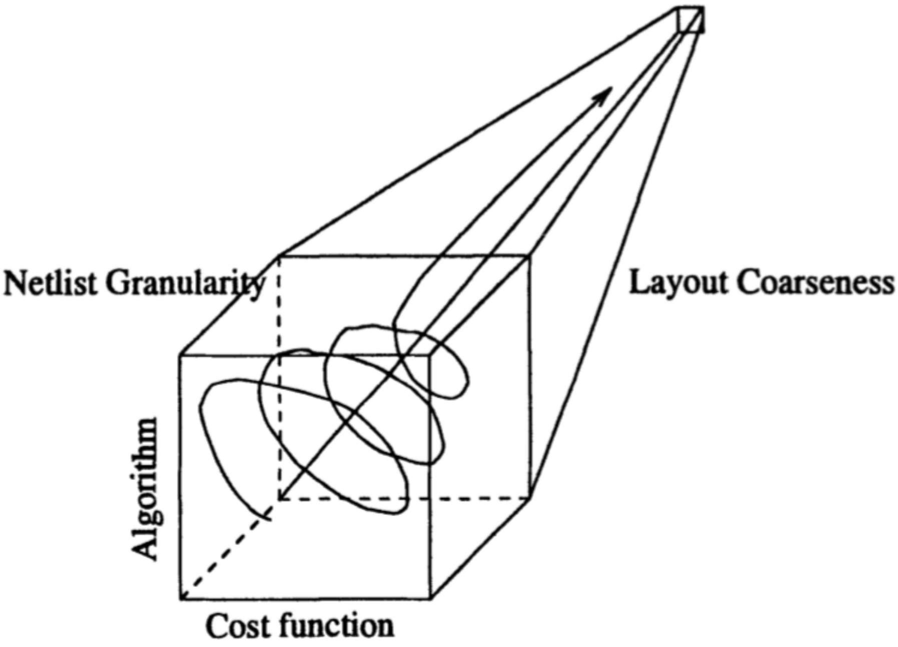


Figure 2.2: Placement cube with four dimensions representing the placement elements interaction

Modern placement problem is very complex and involves several cost functions at the same time, such as cut, wire length, timing and congestion. As shown in Fig. 2.2, the placement problem consists of four essential elements: cost function, algorithm, netlist granularity and layout coarseness. Thus it is critical to discover a practical methodology which understands the interactions between these placement elements and picks the right combination at the right time [2]. The placement process for large-scale circuits, as shown in Fig. 2.3 is separated into three subsequent steps: global placement (GP), legalisation and detailed placement. Global placement emphasises on proper global cell positioning and overall density distribution, allowing at the same time cell overlaps or other constraint violations. These overlaps and violations are then resolved during the legalisation step, while detailed placement improves the quality of the legal placement performing local optimisations. Following we describe the different stages of the placement procedure [1].

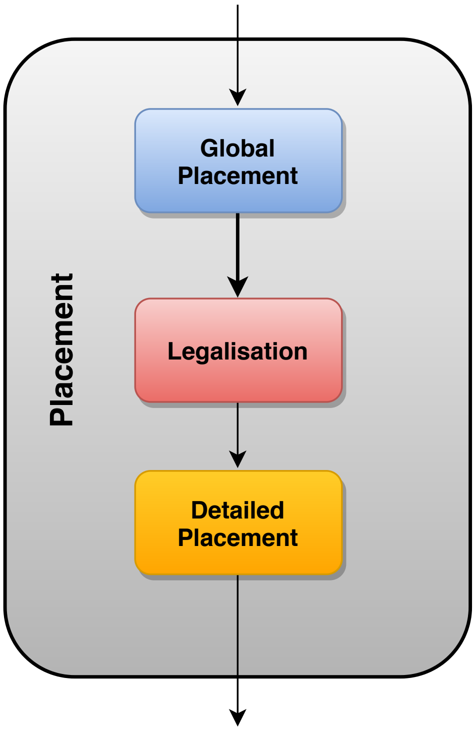


Figure 2.3: Placement Flow Steps

Global Placement

Global placement is the first step of the placement process. It focuses on finding an initial placement of the logic cells. Global placement often ignores cells' specific shapes and sizes and does not attempt to align their location with valid grid rows and columns. Since it ignores the dimensions of the cells, it treats them as points allowing some overlaps among them. Performance optimisations can also take place during global placement. However, timing estimation can be inaccurate during the early stages of global placement. Also, other optimisations, such as total wire length, may restrict the placement algorithm from spreading the logic cells across the layout and achieving density distribution. Thus, it is more common to perform optimisations during the last stages of, or after, global placement. There are many placement algorithms used during global placement, with the most common of them to be shown in Fig. 2.4.

In partitioning-based algorithms, the netlist and the layout are divided into finer sub-netlists and sub-sections respectively, according to a cut-based cost function. It is an iterative process and is repeated until each sub-netlist and sub-section is small enough to be handled optimally. An example of this method is min-cut partitioning placement, which aims to minimise the number of cuts among each layout subsection.

Analytical placement algorithms minimise a given cost function, like total wire length or circuit delay, using mathematical techniques such as numerical analysis or linear programming. Such methods often require certain assumptions, such as to

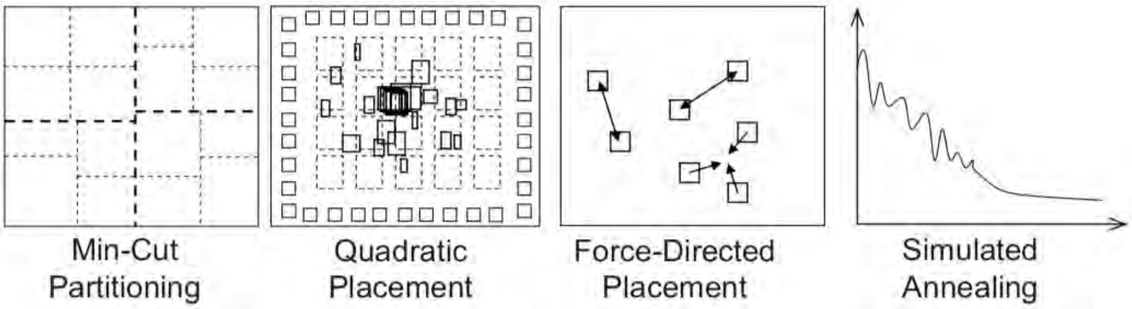


Figure 2.4: Most common global placement techniques

know if the objective is convex or not or to treat the placeable cells as dimensionless points. Examples of analytic techniques include quadratic placement and force-directed placement.

In stochastic algorithms, randomised moves are used to optimise the cost function. An example of this approach is simulated annealing. Simulated annealing is an iterative optimisation method that was inspired by the metal cooling process. The main objective is to achieve a better placement solution via a set of predefined moves and a cost function. A move that results in a better solution is always accepted regardless of the temperature. When a move leads to a worse solution, the algorithm depends on randomly accepting moves during the initial phases of the algorithm, where the temperature is high. As the algorithm proceeds, the temperature decreases and the algorithm accepts fewer worse moves. The acceptance of worse move is essential for the placement algorithm to be able to escape from local minimums of cost function and overall achieve the global optimum solution.

Legalisation

Global placement produces a placement solution trying to optimise a defined cost function. Since many of the placement techniques treat the placeable cells as dimensionless points, the global placement positions do not align with the power rails. Also, some overlaps among the cells may exist. Therefore, global placement must be legal. Legalisation seeks to align placeable cells with rows and columns and remove overlaps. During the overlap elimination, legalisation aims to minimise displacement from global placement locations as well as the impact on wire length and circuit delay. Legalisation step is necessary not only after global placement but also after incremental changes, such as cell resizing and buffering. Unlike global placement algorithms, legalisation requires the cells to be distributed enough across the layout region and have small overlap. From the above, it is noticeable that the legalisation process strongly depends on the initial global placement solution quality.

Detailed Placement

Once the legalisation solution is produced, it can be improved, during detailed placement, concerning a given objective. Detailed placement incrementally improves the location of each cell by local operations, such as swapping neighbouring cells to reduce total wire length, or shifting several cells in a row to create room for another object when whitespace is available.

2.2 RADHARD Background

Earth's atmosphere acts as a semipermeable filter allowing light and heat to get through while blocking most of the radiation existing in space. Outside the protective shield of Earth's atmosphere, there is a universe full of radiation. Space radiation is different from those experienced on Earth, such as X-rays or Gamma rays. It is comprised of atoms accelerated to speeds equivalent to the speed of light, causing their electrons to strip away and only their nucleus to remain [3].

Modern electronic circuits due to transistor downscaling became very sensitive to the radiative space environment. Some particle radiation is so energetic that it can penetrate the device and interact with its electronic circuit. This interaction can cause a wide variety of effects that range from the degradation of performance to functional disruptions affecting any system operations. Radiation effects, depending on their consequence to electronic circuits, are often divided into two general categories: Hard Errors and Soft Errors. Hard Errors cause permanent damage to the electronic device, while Soft Errors lead to a circuit malfunction without damaging it.

2.2.1 Types of Radiation in Space

The radiation profile beyond Earth's atmosphere comprises of 4 major radiation sources:

- Galactic Cosmic Rays
- Solar Wind
- Van Allen Radiation Belts
- Solar Flares and Coronal Mass Ejections

Cosmic rays are a form of high-energy radiation, originating in outer space, that travels at nearly the speed of light and strike the Earth from random directions. These high-energy charged particles consist of mainly (89%) protons but also nuclei of helium (10%) and heavier nuclei (1%). Upon impact with the nuclei of atoms in the upper layers of Earth's atmosphere, cosmic rays can produce showers of secondary particles, mainly pions, that sometimes reach the surface [4].

The solar wind is a stream of charged particles released from the Sun's upper atmosphere, called corona. This plasma mainly consists of electrons, protons and alpha particle with kinetic energy between 0.5 and 10KeV. Its density and speed may vary over time and solar latitude and longitude [5].

A Van Allen radiation belt is a zone of radiation particles trapped by and held around Earth by its magnetic field. Earth has two such radiation belts, and sometimes others may temporarily be created. The belts trap mostly energetic electrons and protons, with other nuclei like alpha particles being less prevalent. The outer belt is made up of billions of high-energy particles that originate from the Solar

Wind while the inner belt results from the interactions of cosmic with the Earth's atmosphere [6].

A Solar Flare is associated with the ejection of plasmas and particles from the solar corona into outer space. Powerful flares are often, but not always accompanied by a Coronal Mass Ejection, which is a significant release of plasma and magnetic field. The ejected plasma is released into the solar wind and the particles, associated with it, can penetrate the upper atmosphere [7].

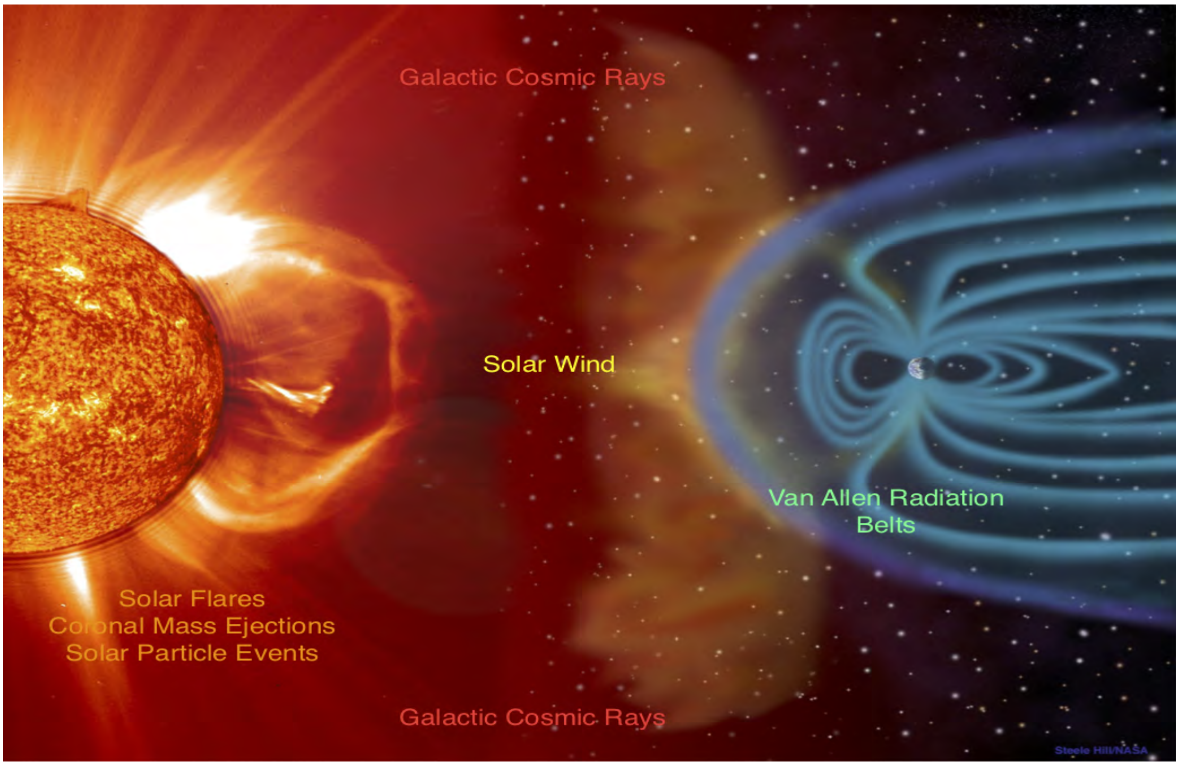


Figure 2.5: Major radiation sources in space

2.2.2 Radiation Effects in electronics

The rapid evolution in the field of VLSI technology over the past decades has brought the emergence of Integrated Circuits (ICs) that operate at high frequencies and with low power requirements. Although modern chips become more and more efficient, their susceptibility to cosmic radiation, due to the reduction in device feature sizes and supply voltage, constitutes a constant concern. A charged particle can strike the electronic device and cause non-destructive or destructive effects depending on the particle's energy, type and the strike location. The radiation effects in microelectronics can be divided into two general categories: the Cumulative Effects and the Single-Event Effects (SEEs).

Cumulative effects are long-term effects and produce gradual changes in the operational parameters of the device. They require the device to operate under continuous levels of radiation before device malfunction becomes obvious. The most major cumulative effects in electronics are Total Ionising Dose (TID) and Displacement Damage (DD):

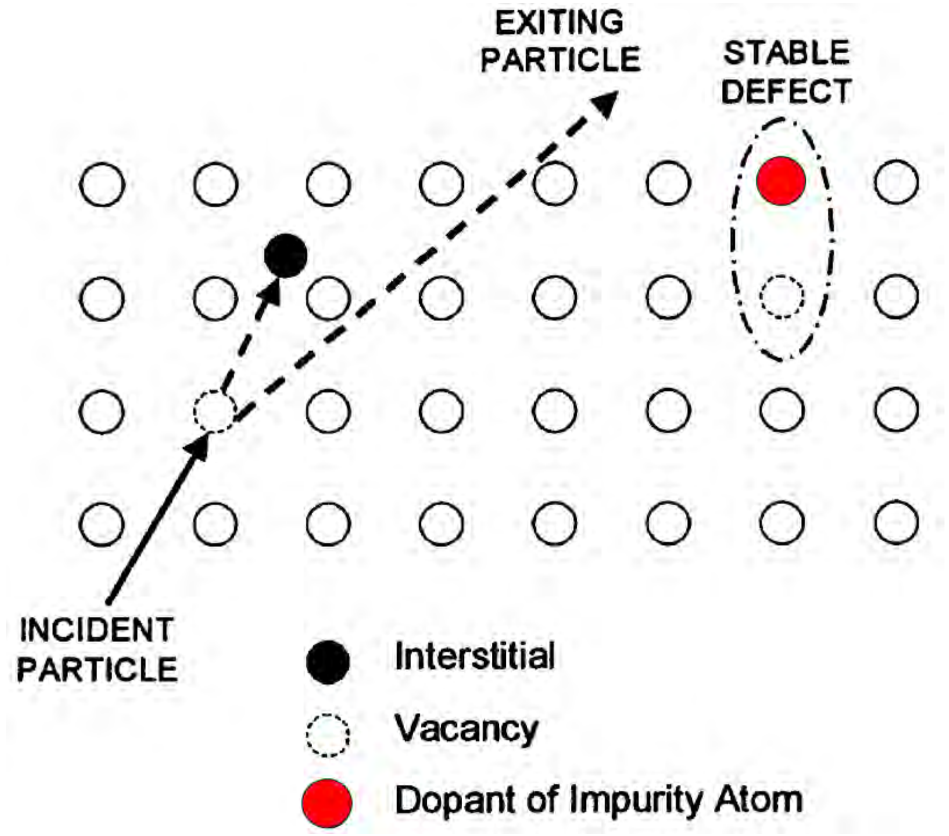


Figure 2.6: Displacement Damage occurs by an incoming radiation particle

- **Total Ionising Dose (TID)** effects occur when electrons and protons create an excess charge in the dielectric layers used for electronic devices' insulation. Extended exposure of a device to TID radiation can shift the threshold voltages, making transistors easier or harder to alternate. It can also increase leakage current, causing the on and off states of the transistors to become less distinguishable.
- **Displacement Damage (DD)** is the result of nuclear interactions, typically scattering, which cause lattice defects. The collision between an incoming radiative particle and a lattice atom subsequently displaces the latter from its original lattice position creating the same time a vacancy, as shown in Fig. 2.6. Displacement Damage is caused by long-term non-ionising damage from protons of all energies, high-energy electrons (above 150KeV) and neutrons. However, Displacement Damage is not such a major effect as Total Ionising Dose or Single-Event Effects.

Both of the above cumulative radiation effects are hard errors since they cause permanent damage to the electronic device, but they cause it long term.

Other significant effects caused by radiative particles are the Single-Event Effects (SEEs). As an event is called a particle strike caused in an electronic device. SEEs cause instantaneous changes or transient behaviour in circuits, leading to erroneous performance. They are classified into two types of errors: Hard Errors and Soft Errors. Hard errors are a type of errors causing permanent damage to the device, and most of the time to memory chips.

On the other hand, soft errors are events in which data are corrupted, but the device itself is not permanently damaged. Soft errors can affect a device in different ways. In some cases, they may result in data corruption, detectable or not, at the system level, while in other cases, they can cause a circuit's malfunctioning or even a system crash.

Soft errors can be classified into the following categories:

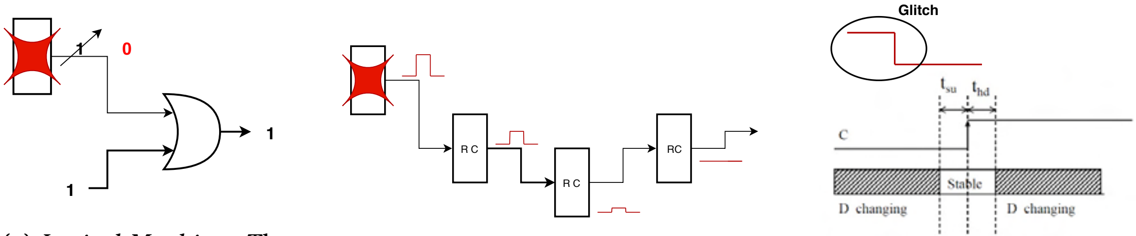
- **Single-Event Transient (SET)**: The event causes a voltage in the circuit's logic, which becomes a bit error when captured in a storage element. In case a particle strike hit more than one combinational elements of the circuit, the fault caused is called **Single-Event Multiple-Transient (SEMT)**.
- **Single-Event Upset (SEU)**: The particle strike affects node's charge and causes a logical upset in sequential elements of the circuit. When the event causes a bit-flip (upset) in a memory cell or a latch, it is called **Single-Bit Upset (SBU)**. Mainly, using the term SEU, we are referring to SBU. When the particle strike causes the upset of two or more bit in the same word, the upset is called **Multiple-Bit Upset (MBU)**. Finally, when the event causes the upset of two or more sequential elements, it is called **Multiple-Cell Upset (MCU)**.
- **Single-Event Functional Interrupt (SEFI)**: The event causes loss of functionality due to the perturbation of control circuits, like state machines, placing the device into an undefined state.
- **Single-Event Latchup (SEL)**: The particle strike causes loss of device functionality due to a single-event induced current state. A SEL may cause permanent damage to the device, in which case the result is a hard error. The SEL results in high operating current, above device specifications. The latched condition can destroy the device, drag down the bus voltage, or damage the power supply.
- **Single-Event induced Burnout (SEB)**: It can cause the device's destruction due to a high current state in a power transistor, resulting in a permanent device failure (hard error).

The failure rate of a device induced by soft errors is called Soft Error Rate (SER). SER usually is measured for a given environment in FIT units (Failures In Time), where 1FIT denotes one failure per billion device operation hours. Typical SER values for electronic devices can range between 100 and 100000FIT, *i.e.* one soft error per year. The FIT value is either predicted by simulation or is the result of a series of experimental error measurements. SER monitoring shows that the hard error failure rate, due to external events, such as electrical latchup, is at maximum 10FIT, but commonly is much less. For the modern process technologies, the SER of 1Mbit of SRAM, one of the most susceptible components to soft errors, is in the order of 1000FIT. Depending on the number of Mbits of SRAM in a device, the above SER can be escalated enough, in a way, that for a device containing multiple Mbits of SRAM it can be higher than the total failure rate due to all other mechanisms.

The occurrences of random particle strikes are distributed fairly uniformly in space and time. The probability of a particle strike in a circuit node is thus roughly proportional to its active area. Each particle strike is not able to cause a transient fault (TF) in a circuit node. The particle strike affects a circuit's node only in the case it causes a charge which is greater than a critical charge (Q_{crit}), specific for each circuit node. The characteristics of a transient pulse at a node strongly depend on the incident particle's energy distribution and node's Q_{crit}. However, various masking mechanisms determine whether a transient pulse will propagate to primary outputs, latches or flip-flops (*fault catching points*) and cause a soft error or not. The three major masking mechanisms, Fig. 2.7, are the following:

- **Logical Masking**: occurs when there is no sensitised path from the incident node to any of the *fault catching points*. This phenomenon appears when a transient fault arrives at an input of a subsequent cell, which prevents its propagation because at least one of the other cell's input is at a controlling logic value. For example, the controlling value of an AND gate is logic 0, whereas logic 1 is the controlling value of an OR gate.
- **Electrical Masking**: occurs due to transient pulse's width attenuation in a sensitised path from its occurrence node to any of the *fault catching points* until it is eliminated. Thus, electrical masking strongly depends on the electrical properties of the gates in the sensitised path.
- **Timing-window Masking**: occurs when the transient pulse, even though logical and electrical masking did not prevent its propagation to fault catching points, it does not cause a soft error because its arrival point is sufficiently separated in time from the arrival of clock edge. Parameters which determine the occurrence of the timing masking include the arrival time of the pulse at a sequential element, as well as its SETUP and HOLD times. The last two variables define its latching window, during which the sequential element samples its inputs. As the transient pulse is momentary, if the fault arrives outside the latching window, it is masked and does not cause a soft error.

The above masking mechanisms thus lead various circuit nodes to be quite different in their soft errors susceptibility.



(a) *Logical Masking*: The OR gate does not let the fault to propagate since it is in its controlling value

(b) *Electrical Masking*: The parasitic capacitance and resistance of the following gates in the sensitised path attenuate the particle strike's glitch

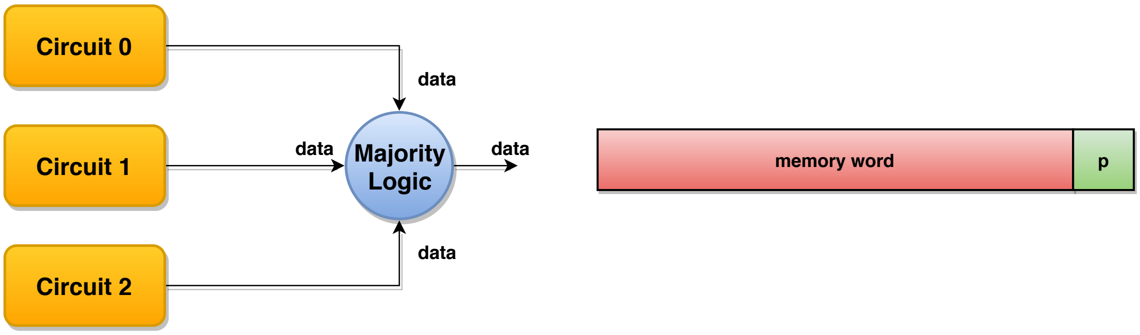
(c) *Timing-window Masking*: The fault arrives at the FF outside the [SETUP HOLD] window, during which the FF catches the input data

Figure 2.7: The different masking mechanisms that can affect the propagation of a transient fault

2.2.3 Radiation Hardening Techniques

Radiation Hardening is the process of making electronic components and circuits resistant to damage or malfunction caused by ionising radiation [8]. The radiation hardening techniques can be applied at different levels and categorised as

- System-level techniques
- Device-level techniques
- Circuit-level techniques

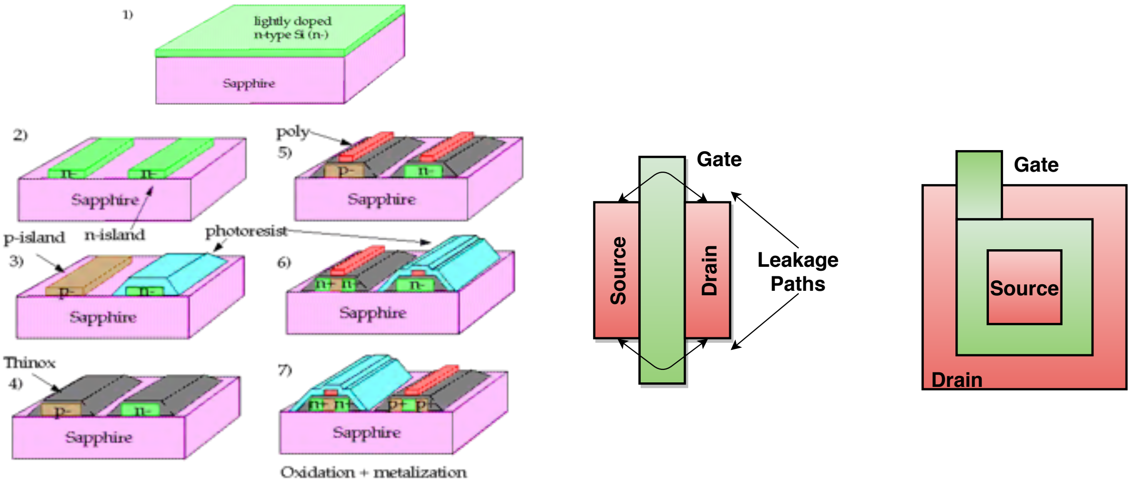


(a) TMR applied to the whole circuit

(b) Adding parity bit for a memory word

Figure 2.8: Two common system-level radiation hardening techniques

System-level hardening techniques aim to achieve error detection/tolerance ability in the design. To accomplish that, system designers mainly apply redundancy techniques, with the most common one, the Triple-Modular Redundancy (TMR). In TMR technique the whole circuit is replicated three times, and a majority voter is added to filter the corrupted value propagated by one of the TMR replicas, Fig. 2.8a. For memory circuits, the system-level hardening technique is applied by adding a parity bit to the memory word, Fig. 2.8b. Each time a word is written to the memory, a parity bit is generated and appended to the data. Upon data retrieval, the parity of the obtained data is calculated, and a check compares the later one with the stored parity bit. In case a single error has occurred, the data parity won't match with the parity bit. In that case, an additional circuit is needed to correct the data. Although there are various methods to correct the errors, like hamming codes, their use may result in significant area and power overhead. Even though a single error can be detected, the parity check is not able to reveal a double fault because the data parity will match with the parity bit.

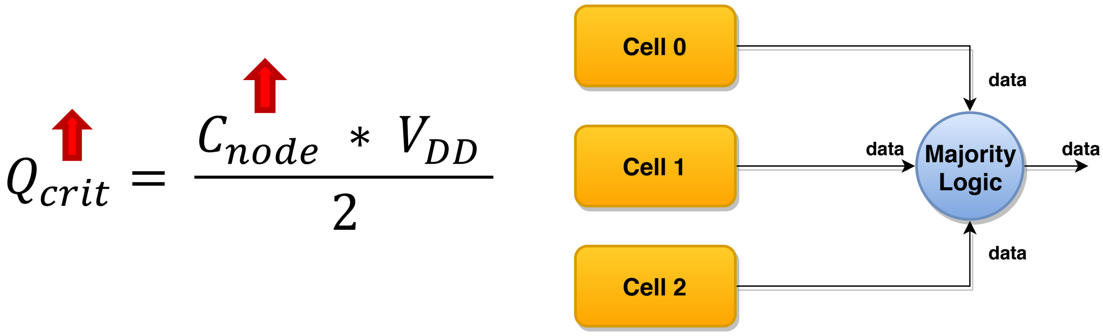


(a) Silicon On Sapphire (SOS)

(b) Shielding Method

Figure 2.9: Two widely device-level radiation hardening techniques

Device-level hardening techniques aim to reduce and mitigate the charge collection at the region of the particle strike. These methods require a change to the fabrication process. One technique is to manufacture the chips on insulating substrates instead of the known semiconductor wafers. Silicon-On-Insulator (SOI) and Silicon-On-Sapphire (SOS), Fig. 2.9a, are assumed to provide radiation-hardened chips. Another device-level hardening technique is either the shielding of the entire package against radiation, to reduce the exposure of the device, Fig. 2.9b. Although these methods protect the devices against radiation sufficiently, they increase a lot the fabrication cost.



(a) Qcrit is proportional to node's capacitance

(b) Apply TMR to critical cells of the circuit

Figure 2.10: Common circuit-level radiation hardening techniques

Circuit-level hardening techniques focus on changing the circuit design to achieve SEEs mitigation. Typically, circuit-level mitigation techniques must either filter or dissipate the collected charge or provide some form of redundancy to prevent the corrupted data propagation. A method to reduce the soft error rate is to increase the node capacitance, Fig. 2.10a. A particle strike can cause a fault only if the produced charge is larger than the node's critical-charge, which is defined as $Q_{crit} = C_{node} * V_{DD}/2$. From the above equation, we can see than increasing the node capacitance can make the specific node more resistant to radiation. To achieve the above, we can either add a capacitor or increase the wire length of the node connectivity. Although, this method imposes a significant area and power penalty due to the added capacitor. Spatial redundancy techniques such as TMR triplicate the critical circuit/cell and add a majority voter to filter the transient fault, as shown in Fig. 2.10b. TMR can mask a single error happening among the three logic replicas. This fault can be masked thanks to the existence of the other two TMR instances. However, this method results in a large area overhead due to triplication.

Chapter 3

Existing Works

3.1 Existing Works in the Literature

The IC design flow for harsh environment differs from standard digital design flow. The latter one is shown in Fig. 3.1. Mainly, space applications impose additional requirements on the designs and tools. The main challenges are related to the resistance to the ageing and radiation effects, where the latter includes Total Ionising Dose (TID) and Single-Event Effects (SEEs).

The ageing of transistors and TID effects result in similar outer effects, such as increased leakage and reduced performance. As mentioned in Chapter 2.2.2, TID effects are caused after prolonged exposure of a device to electrons and protons, creating an excess charge in its dielectric insulation layers. On the other hand, SEEs are generally induced by a particle hit on the die surface. They can be observed as Single-Event Transients (SETs), in case of temporary value change of the Boolean gates, Single-Event Upsets (SEUs), in case of a bit-flip of the register, or even destructive, such as Single-Event Latchup (SEL), in case of induced short in the circuit due to the temporary forming of parasitic bipolar structures or Single-Event Burnout (SEB), in case a power transistor is affected.

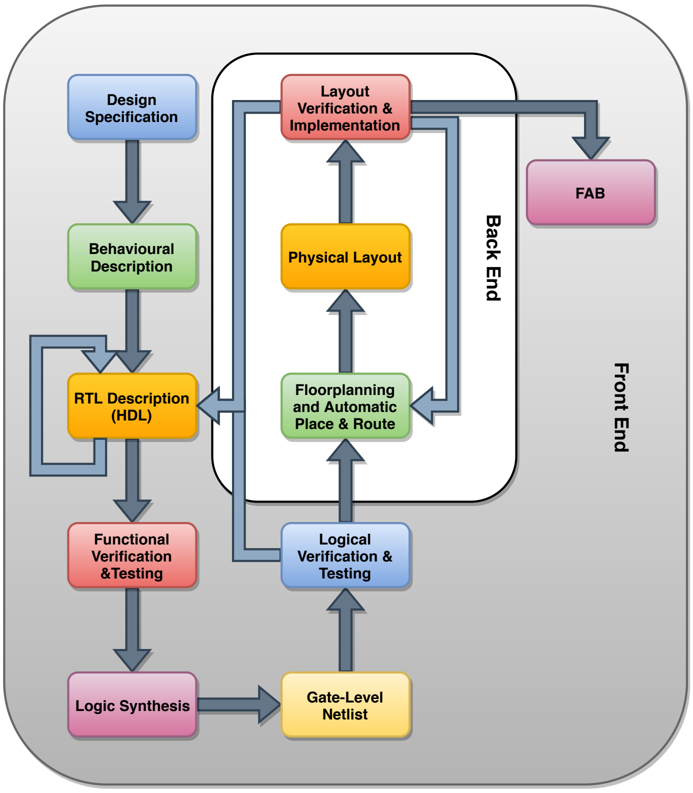

Design Specification
Behavioural Description
RTL Description (HDL)
Functional Verification &Testing
Logic Synthesis
Gate-Level Netlist
Logical Verification & Testing
Floorplanning and Automatic Place & Route
Physical Layout
Layout Verification & Implementation
FAB
Back End
Front End

Figure 3.1: Standard Digital Design Flow

Various radiation hardening methods have been proposed in the literature to mitigate those effects. Most of them take place in physical and device level, while others in the circuit level, and more specifically in placement stage. On the other hand, in industrial EDA tools, there is no significant contribution to radiation hardening. Although, one of the most robust EDA tools, supports a radiation hardening flow during the placement phase.

3.1.1 Radiation Hardening in the Literature

Many works exist in the literature proposing methods to mitigate radiating effects. The methods to address those effects are mainly on the layout level, such as the use of Enclosed Layout Transistors (ELT) against TID effects [9]. Considering SEEs, the SELs could be addressed by the technology, *i.e.* Silicon-On-Insulator (SOI) or Silicon-On-Sapphire (SOS), which are immune in this effect, or careful contacting and introduction of guard rings [10].

For the radiation hardening against SETs and SEUs, different methods could be used. One significant method includes the transistors up-sizing, *i.e.* increase in the relation between width and length of used transistors [11]. Nevertheless, this method is also quite expensive in terms of area, performance and power consumption. The alternative is the use of specific flip-flop architectures that are tolerant against SEUs. An example of such architecture includes the use of Dual-Interlocked CEll (DICE) architecture, where the storing structure of the latch has been made as a ring of 4 inverter pairs [12]. In this case, if one node is affected by the particle hit the other three nodes will keep the latch stable. This structure has also been further improved to LEAP-DICE [13] by taking into consideration the layout effects. Similar approaches have been provided in the know DARE library with Heavy Ion Tolerant storage cell (HIT) architecture [14]. Such radiation-tolerant latches are usually extended with some glitch filter at the data input, to accommodate the SETs efficiently.

With transistor scaling it is very difficult to implement the capable radiation-tolerant architecture of the flip-flop. The small transistor dimensions, reduced critical charges (Q_{CR}) and high integration lead to the increased susceptibility to the SEUs. As a consequence, at the higher Linear Energy Transfer (LETs) there is a high probability that more than one nodes of the latch will be affected by the particle hit. In this case, increasing the distance between the transistors is not such an efficient measure since it is leading to performance drop and unnecessary cell size increase.

Meanwhile, the reduction of Soft Error Rate (SER) can also be achieved by selectively optimising wire length for soft error critical nets. The larger the wire lengths for nets can act as larger RC ladders and can efficiently filter out the transient glitches cause by particle strikes. Based on the above, in [15] a simulated annealing placement algorithm is proposed, aiming to reduce the SER of logic circuits. The SER capture is based on the *logical observability* metric, which is inversely proportional to each node's logical masking effect. So, nodes with lower logical masking ability have high logical observability, since a fault caused in these node has a substantial probability of being captured at a sequential element. As mentioned in Chapter 2.1.1 simulated annealing is a placement algorithm inspired by the metal cooling process, and the acceptance of a move depends on the current temperature. In this work, the cost of simulated annealing is the summation of the logical observability weighted with the wire length of each net, while taking into account at the same time total area and wire length constraints. Considering that, soft error critical nets are assigned longer wire length for glitches filtering, while delay critical nets are assigned shorter wire length for minimising circuit delay. Although, simulated annealing placement algorithms are not sufficient because they strongly depend on their global variables, *i.e.* temperature, initialisation.

As a consequence, it would be better to apply some radiation hardening method at the RTL level. The principles of N-Modular Redundancy (NMR), and especially Triple-Modular Redundancy (TMR) have been used in the space applications for years. In TMR, the fault tolerance is achieved by triplication of the original module, and the single fault can be filtered out through a majority voting [16]. As for addressing SEUs, TMR method could be applied at the level of flip-flops, where single unprotected flip-flop can be replaced by a fault-tolerant structure, including three replicas of the original flip-flop and a majority voter. Regarding addressing SETs, two strategies are possible:

1. generation of three clock trees with a timing offset between them, or
2. adding a temporal filter at the input of TMR-structure

The latter is of significantly lower complexity and area cost. TMR approach has higher potential for the radiation hardness compared to the flip-flop architectural approaches. It is more unlikely that a single particle can influence at the same time two flip-flops that are in some distance to each other. As for the TMR approach to be reliable, it is imperative to guarantee that the FFs of each TMR structure are at a minimum distance from each other so that they are not going to be affected by a single particle hit. This strategy has been shown to provide excellent radiation tolerance. In case a SET is caused in combinational logic near active clock edge, the fault can be propagated to the TMR FF triplet resulting in SEU. In this case, the SETs can be masked using different delay elements in data input among the TMR triplicated FFs [17]. Thus, it is more worthwhile to aim to the SEUs mitigation instead of SETs.

3.1.2 Industry Approach

As mentioned in the previous section, to achieve radiation hardness in the circuit, we have to modify the standard design flow. Modern standard CAD tools are trying to optimise and reduce the routing distance between the connected standard cells aiming to achieve better area, power and performance. To accomplish that, for most of the industrial tools is impossible to add custom constraints into their cost function and thus are unable to achieve fault mitigation. However, an existed robust industrial CAD tool started to support spacing constraints among cells in its newer version, which can be utilised for radiation hardening. Specifically, it supports commands to create spacing groups and set the spacing constraints to be applied both in x-axis and y-axis.

After creating the spacing groups and setting their spacing constraints, the tool's placer is enforced to take into consideration any specified spacing constraints. To ensure that a particle strike will not affect more than one of the specified instances, it places each member of a group into a position with a minimum distance from the other members, at least as much the spacing constraint specifies, both horizontally and vertically. Nevertheless, this approach leads to spread the critical cells further apart, possibly affecting the area, power and circuit delay. On the other hand, ensuring that the Euclidean distance among the TMR members is greater than or equal to the specified spacing constraint is a suitable alternative approach. The latter not only guarantees that a single particle strike will not affect multiple critical cells but also doesn't affect the Power-Performance-Area (PPA) results significantly.

3.2 Thesis' Used Tool

The freedom of using existing standard cell libraries make the radiation hardening during the placement stage of ICs' design flow very appealing. The radiation hardening process can be applied during the legalisation process. As mentioned before, legalisation aims to eliminate any circuit's violations removing cells' overlaps and aligning them to grids. Thus, in addition to these legalisation rules, we can introduce a constraint which aims to radiation hardening.

Our goal is to render the designs resistant to radiation maintaining as much as possible the original design's solution. Various legalisation algorithms are proposed in the literature, aiming to minimise the impact of legalisation as much as possible. The existing algorithms can be categorised to (i) global approaches, where multiple cells are legalised simultaneously, and (ii) local, where cells are legalised one at a time [18]. In the former case, legalisation approaches aim to exploit the global view of the cells' positions and legalise them into positions finding the global optima. In the latter case, cells are legalised trying to place each one into its best available location. Local Legalisers, such as Tetris [19] and Abacus [20], legalise cells one-by-one trying to find their optimal position in a greedy way. Mainly, Tetris handles mixed cells, *i.e.* standard cells and macroblocks, and greedily assigns each cell to its nearest legal position, respecting at the same time the row capacity, and fixes it before continuing with the next cell. However, Tetris has several drawbacks [1], with the most crucial being that (i) it doesn't maintain the cells' relative GP order leading to higher total displacement and wire length, and (ii) fixing each legalised cell into its new positions it subsequently preserves large amounts of whitespace.

These main drawbacks are solved by other improved approaches based on Tetris, like Abacus. The latter legalises only standard cells with the same height and different width, in contrast with Tetris, trying to minimise their displacement from their GP positions. In contrast to Tetris, Abacus, to achieve the minimum displacement, allows already legalised cells to be shifted through its placement row trying to minimise the total cells' displacement maintaining at the same time their GP cell order. A drawback of Abacus legaliser is the fact that supports only standard same-height cells. A variation of Abacus legaliser, called Abax [21, 22], extended Abacus to support mixed height cells and also handles blockages. Thus, Abax used in our work, since it is a suitable legaliser trying to minimise the impact of legalisation to GP solution. In the next section, Abax is presented in more details.

3.2.1 Abax Legaliser

Abax is a modification of the classical Abacus Legaliser. It is a sequential legaliser, legalising one cell at a time in legal positions in a specified order. Cell order, according to the goal of the legalisation, depends on various factors, like the cell's initial GP position, cell's area or even the influence of the cells on the circuit's timing. Abax is a greedy displacement-driven legaliser aiming to minimise the influence of legalisation on the GP solution. To achieve this goal, Abax assigns each cell to its nearest row, that fits in, achieving minimum displacement from its GP position. In this way, cells are sorted either by the cell's GP x-coordinate or the cell's area. Thus, Abax supports three cell orders, *i.e.* (i) increasing, (ii) decreasing, and (iii) centre-outwards orders.

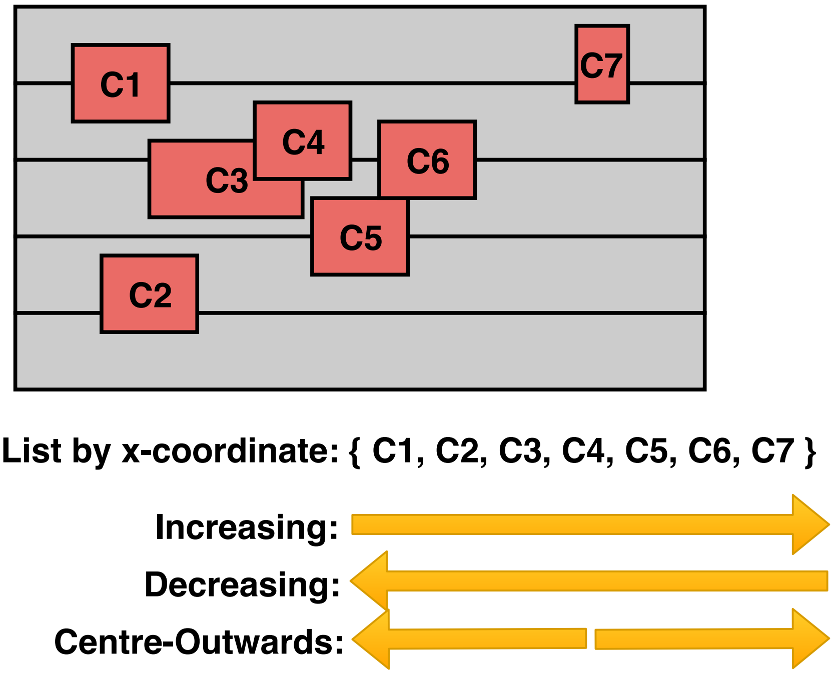
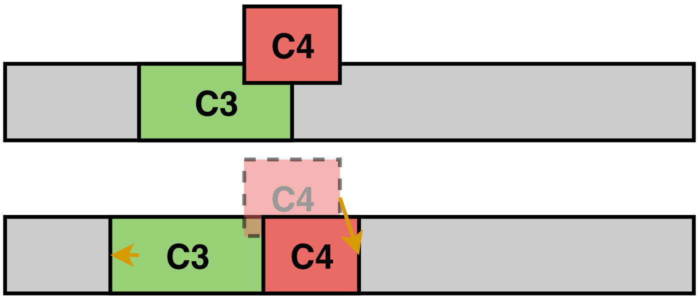


Figure 3.2: Abax's supported cell orderings

As mentioned, Abacus, and subsequently Abax, tries to minimise the cells' displacement from their initial GP positions. Thus, the chosen legal position of each cell is based on its displacement cost. One modification of Abax Legaliser is associated with the cost function evaluating each cell's tentative move. Abax supports Abacus' `single-cell displacement` cost function. However, this cost function has a local overview of the legalisation effect on GP's solution, since it considers only the displacement of the current legalising cell. Thus, Abax introduces two additional cost functions, the (i) `multi-cell total displacement` cost function, and (ii) `multi-cell mean displacement` cost function. The legalisation of a cell can cause the displacement of already legalised adjacent cells. Thus, `multi-cell total displacement` takes into account the total displacement of all moved cells during this tentative move, while the `multi-cell mean displacement` cost function takes into account the average displacement of the moved cells. Thus, these two cost functions have a global overview of the legalisation influence in GP solution.



$$C_{single} = D_{C4} \qquad C_{total} = D_{C4} + D_{C3}$$

$$C_{mean} = \frac{D_{C4}+D_{C3}}{2}$$

Figure 3.3: Abax's supported cost functions

Abax legalises each cell, by checking first the displacement cost assigning it to its nearest row and then examines a set of rows above and below the nearest row, which reside inside a row search bound. The latter is decreased each time a new best cost is found. Although the row search bound reduces the number of examined legal positions avoiding legaliser to check all the core rows exhaustively, it ensures that it does not influence the quality of the solution.

Another important feature of Abax legaliser is the handling of hard macros and blockages. The placement of the latter divides the core rows into a set of row segments, called subrows. Its starting x-coordinate and its width determine each subrow. Thus, the legalisation of each cell depends on the available subrows that the corresponding cell fits in. Abax supports two approaches for handling hard macros and blockages:

- **Sub-Row Assignment (SRA)**: This approach, Fig. 3.4 as originally proposed in the Abacus, preserves the initial GP's cell order only within the boundaries of each subrow. After a cell is legalised in a subrow, it can be shifted inside the current subrow. However, moving cells to adjacent subrows are not allowed.
- **Sub-Row Re-assignment (SRR)**: This is an alternative approach of handling macros and blockages. In contrast to SRA, SRR allows moving cells to adjacent subrows, Fig. 3.5. When placing a cell into a subrow causes a subrow overflow, some cells are shifted to adjacent subrows to be able to place the current cell into this subrow. This way, SRR preserves GP's global cell order.

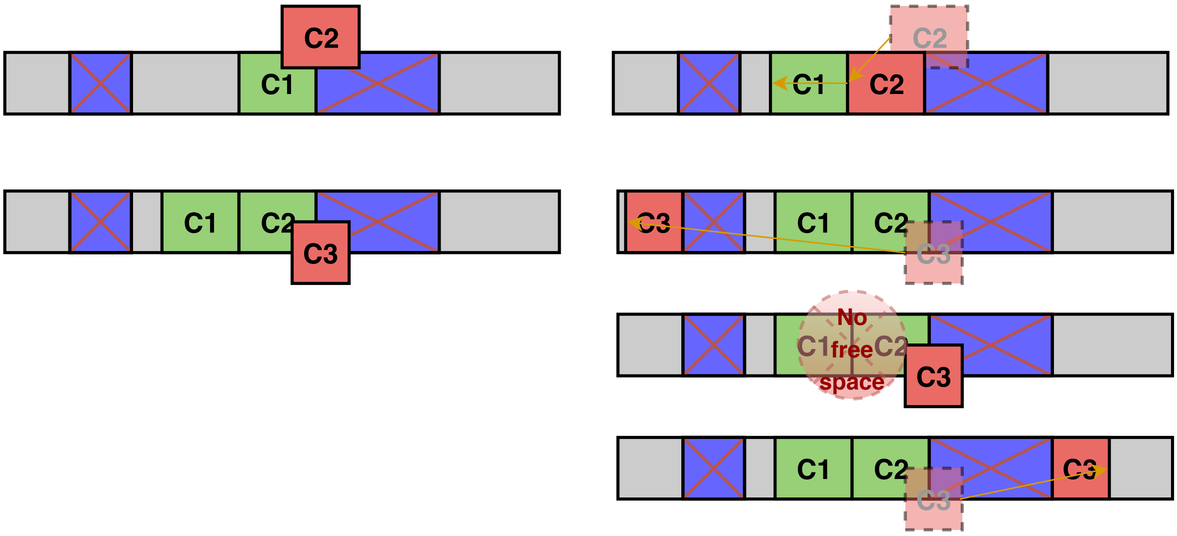


Figure 3.4: *SRA*: Allows already legalised cells to be shifted into the subrows they are assigned to

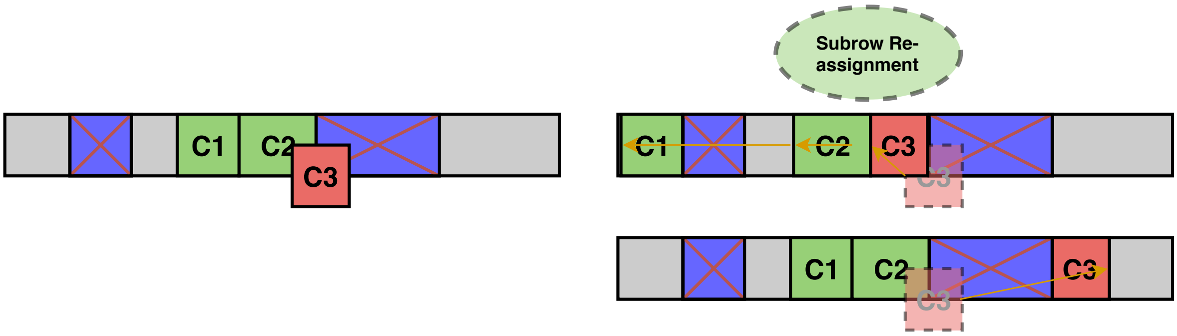


Figure 3.5: *SRR*: The legalisation of a cell can cause a cell wave to previous subrows in order to maintain GP's cell order

Moreover, Abax extended Abacus also to support cells with different heights, *i.e.* Multi-Row-Height-Cells (MRHCs). The height of MRHCs must be an integral multiple of row height. Since the size of MRHCs is larger than the size of Single-Row-Height-Cells (SRHCs), the impact of their legalisation to metrics, such as Total Wirelength (TWL), is much larger. So the MRHCs must be legalised in a way the influence to TWL to be minimised. Abax supports MRHCs legalisation implementing a Tetris-like algorithm, by fixing MRHCs to the legal position with the minimum displacement from the GP position and they can not move again during the legalisation procedure. The MRHC legalisation approach will be presented with a contrived example.

The algorithm starts with an initial placement produced during Global Placement with blockages already placed, as shown in Fig. 3.6. The algorithm finds for each cell the legal position with the minimum displacement finding the corresponding subrows by scanning the core area into two opposite directions, *i.e.* bottom-up and top-down.

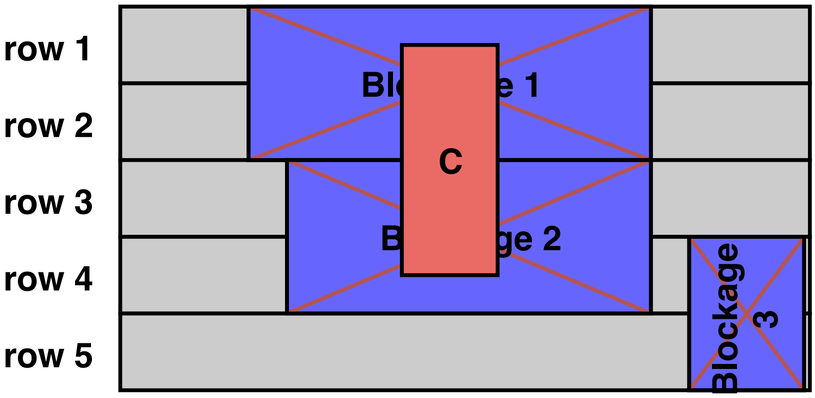


Figure 3.6: MRHC Legalisation Example: Initial Global Placement solution with pre-placed blockages

First, Abax searches the subrows in the bottom-up direction. In this phase, the algorithm finds the nearest subrow considering the lowest y-coordinate of the MRHC. Then, considering the new x-coordinate of the cell, the algorithm checks from the current subrow to the height of MRHC, if the cell doesn't cause any overlap with pre-placed blockages or MRHCs. For example, in Fig. 3.7, the nearest subrow from the bottom of cell *C* is in row 4 and on the left of *Blockage 2*. Starting from this subrow, the algorithm checks if placing *C* into this subrow causes any overlap to the above rows. In case a legal position is found, the displacement cost is determined.

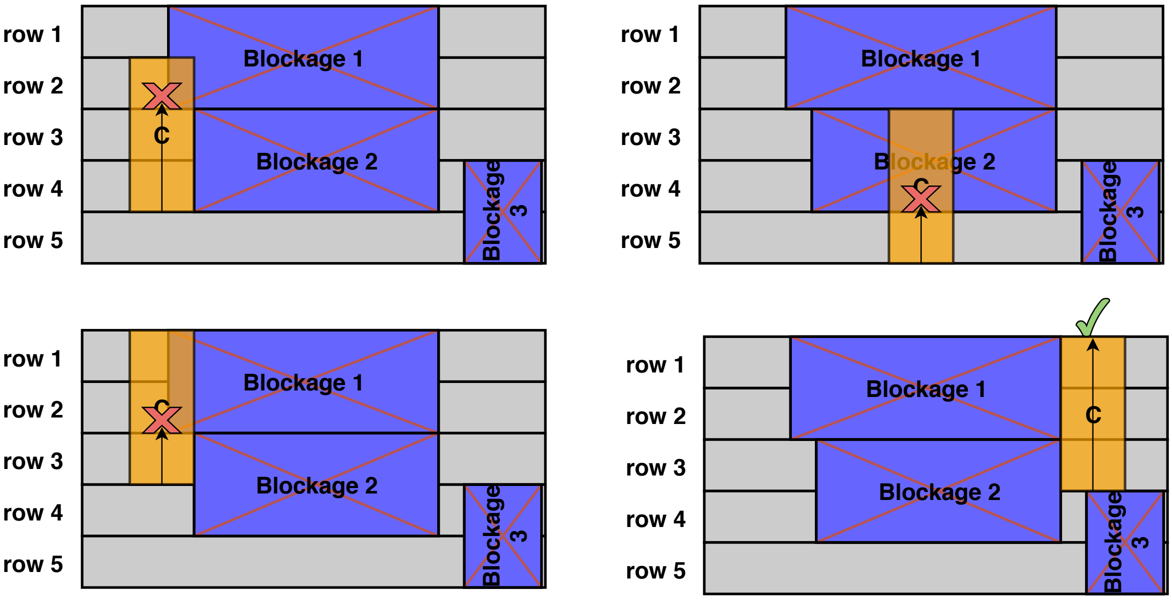


Figure 3.7: MRHC Legalisation Example: Bottom-Up Subrow Scan

However, the bottom-up scan is not enough to find the position with the minimum displacement from the original location. So, then the algorithm continues with the top-down scanning direction. The combination of bottom-up and top-down scans gives the whole set of legal positions. In Fig. 3.8 the top-down scanning phase is presented, starting from the nearest subrow of the upper y-coordinate and scans the rest subrows the same way as in bottom-up, but in the opposite direction.

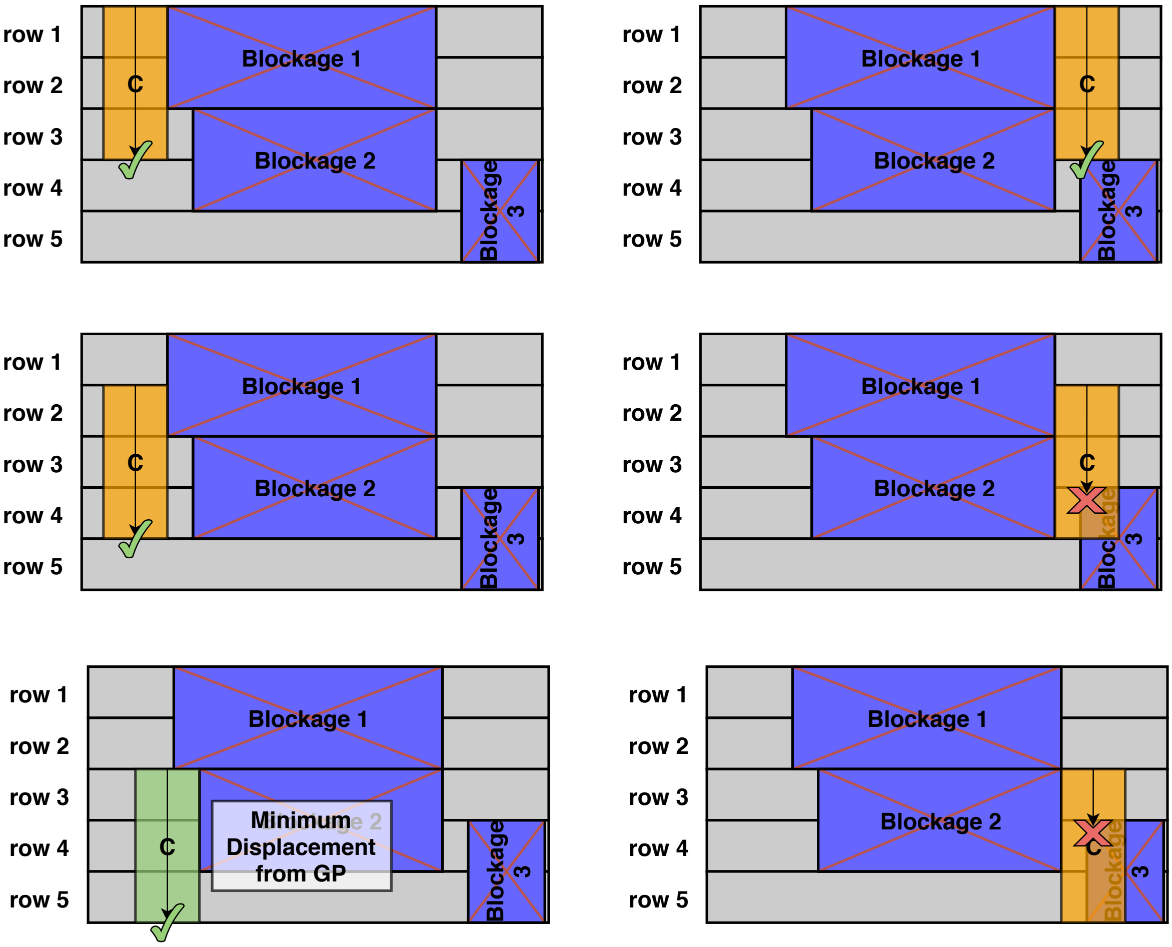


Figure 3.8: MRHC Legalisation Example: Top-Down Subrow Scan

Finally, the entire set of the legal positions for the MRHC is retrieved with their displacement from the initial GP position. Since the algorithm legalises the cell into the position with the minimum displacement, it places the cell into the subrow with the minimum displacement cost from the above set of legal positions. The lowest left figure in Fig. 3.8 highlights the position the cell *C* is finally legalised.

After the legalisation of MRHCs, the latter become fixed into their updated legal positions, to avoid any overlaps being caused during the next step, *i.e.* SRHCs legalisation.

All the above characteristics of Abax Legaliser, make the latter a suitable legalisation algorithm able to be modified to support RADHARD cells, by applying a minimum spacing constraint among the TMR triplet members. Mainly, MRHCs legalisation and blockages handling are two Abax's features that are utilised to achieve the spacing constraints satisfaction. In the next chapter, we are presenting the way the above two features are used to satisfy the spacing constraints.

Chapter 4

RADHARD Legalisation

Modern electronic circuits due to technology shrinking became very sensitive to radiation, leading to apply radiation hardening techniques to mitigation radiation effects. In our work, we focus on the mitigation of SEUs, converting each sequential element to TMR structures and applying a spacing constraint to reduce the probability a particle strike affecting more than one members of the TMR triplet. An efficient way to satisfy the spacing constraint is during the placement stage of the ICs design flow and specifically during the legalisation process. Thus, in our work, we implemented a Radiation-Hardening (RADHARD) Legalisation algorithm, developing four approaches to achieve better PPA results. The first approach was to extend Abax to support RADHARD cells satisfying their minimum spacing constraint. However, legalisation strongly depends on the GP solution, which can place cells of the same TMR triplet far away. To improve the results, we also supported a maximum spacing constraint forcing legaliser to place the triplet cells inside a region. However, a more efficient way to achieve better results is by modifying Abax to consider the total wire length during the evaluation of each cell's move. Finally, another approach which can have a significant impact on PPA results is taking into account the slack during the legalisation of each cell. The rest of this chapter describes in more details the above approaches we followed during the implementation of our RADHARD Legalisation Algorithm.

4.1 Min-bounded RADHARD Legalisation

The technology shrinking the last few years leads electronic devices to be more and more susceptible to SEUs. The small transistor dimensions and the reduced critical charges result in a higher probability that a sequential element will be affected by a particle strike. To alleviate this problem, we used a widely used technique called N-Modular Redundancy (NMR). NMR is a hardware redundancy method which uses N replicas of the same module and a majority voter which compares the outputs of the N replicas and outputs the correct value.

In general, NMR technique can mask at most $(N-1)/2$ faults happening at the same time to some of the replicas, since the remaining unaffected NMR instances can mask them. The higher the grade of NMR, the higher the module is protected against transient faults, although the higher negative impact on the area, power and timing of the circuit. There is a tradeoff between Quality-of-Results (QoR) and the circuit resistance to transient faults. So, in most practical applications, a commonly used grade of redundancy is Triple-Modular-Redundancy (TMR). TMR is a redundancy method with the lowest impact on PPA and can mask a single fault happening among the triplet members.

In TMR, fault tolerance is achieved by triplication of the original module and usage of a two out of three majority voting, referred to as *TMR structure* for the rest of the thesis. For Single-Event Upsets (SEUs), this method may be applied at the FF level, replacing the original ones by the triplet and the voter. For SETs, two approaches are possible, using three, separate clock trees, with timing offset between them, or adding a temporal filter at the TMR circuit section. The latter is of significantly lower complexity and area cost. As for the TMR approach to be reliable, it is imperative to guarantee that the FFs of each TMR structure are placed at a minimum distance from each other so that they are not going to be affected by a particle strike.

In this work, we propose a TMR Radiation Hardening (RADHARD) methodology, including a custom Post-Placement Legalisation algorithm, able to satisfy user-specific, minimum distances among the FFs of each TMR structure. Our algorithm is fully compatible with industrial EDA flow and can easily be integrated into the standard design flows.

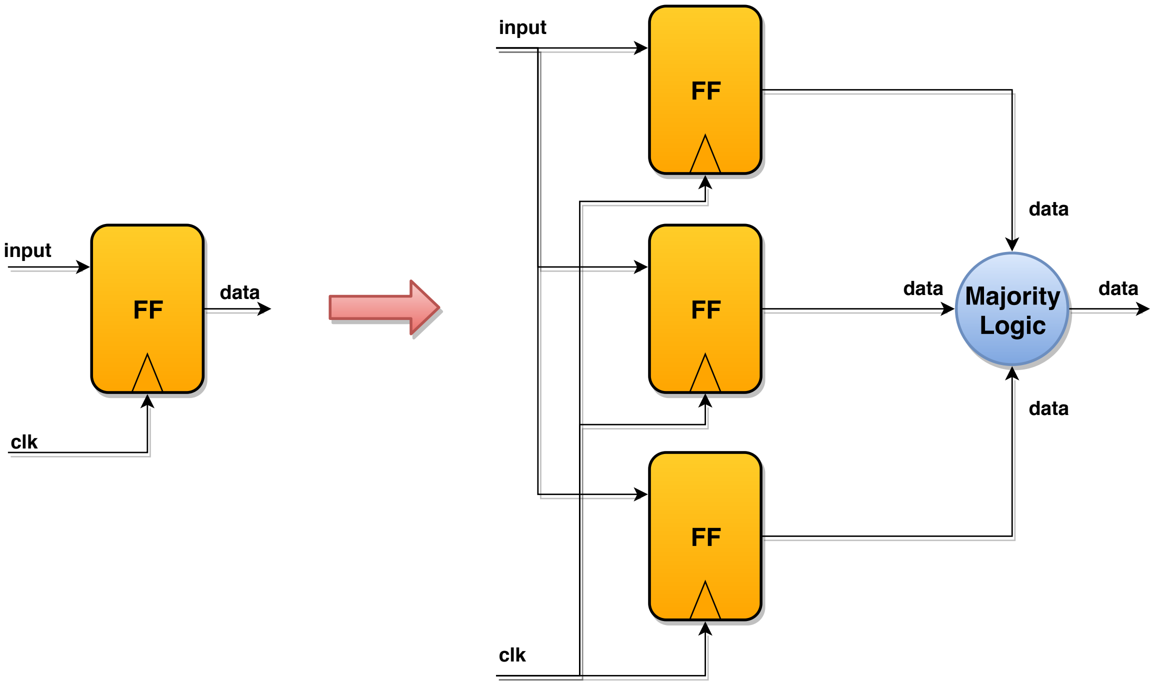


Figure 4.1: TMR Conversion of a sequential element

4.1.1 Create TMR Netlist

As mentioned before, TMR is a redundancy method which uses three identical instances of the desired circuit logic and a voter which outputs the majority of the three replicas. TMR can mask a single fault happening among the triplet members at the same time. This fault can be masked thanks to the existence of the other two TMR instances.

Our spatial TMR RADHARD flow starts by converting the original netlist into a TMR one. Fig. 4.1 presents the TMR structure conversion in our flow, which exploits the TMR technique only for FFs, and not for the entire circuit. Each FF of the original design is converted into a TMR structure. This structure consists of three FFs replicas and a majority voter. FFs' inputs are tied together, as they must receive the same input and the clock signal, while their outputs are connected to the voter. The latter (Fig. 4.2) compares their result and returns the majority of them to the rest of the circuit logic.

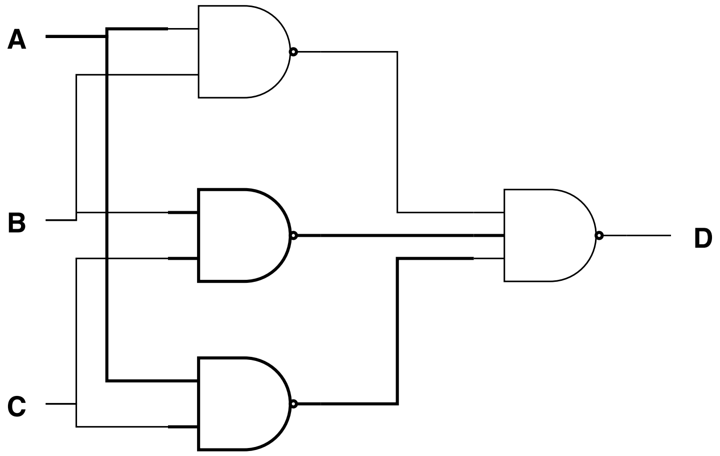
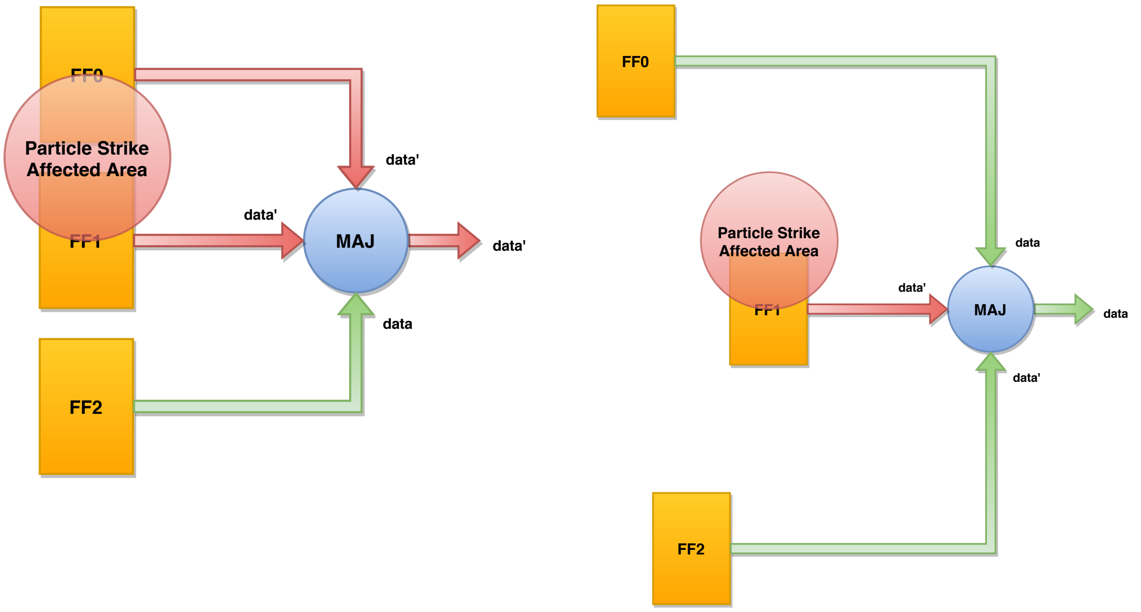


Figure 4.2: Schematic of majority gate using NAND2 gates

4.1.2 TMR Groups and Minimum Spacing Constraints

A key element in the RADHARD Legalisation process is the notion of the TMR group. A TMR group consists of the FFs of a TMR structure. When a radiation particle strikes the integrated circuit (IC), it creates a charge which can affect an area containing multiple circuit components. In recent technologies, this phenomenon has been inflamed by the components shrinking. Moreover, state-of-the-art industrial EDA tools tend to place interconnected cells near each other to achieve better PPA results. However, for a TMR group, this would be disastrous. Placing the members of a TMR group near each other will significantly increase the probability of a fault being induced, as more than one members being affected by the same particle strike, as shown in Fig. 4.3a.

In order to make the TMR group resistant to particle hits, the individual FFs of a TMR group must be spatially distributed, based on a user-specified spacing constraint, signifying that all group members should be spaced from each other by a given minimum distance, as shown in Fig. 4.3b. The value of the spacing depends on the charged area range, which is highly related to the radiation particle expected diameter, its Linear Energy Transfer (LET) and the speed and angle under which it hits the chips.



(a) Particle strike affecting multiple members of a TMR triplet

(b) Particle strike affecting only one TMR member since they are distributed

Figure 4.3: Applying Minimum Spacing Constraint reduces the probability a particle strike affect more than one TMR members

4.1.3 Abax Modification

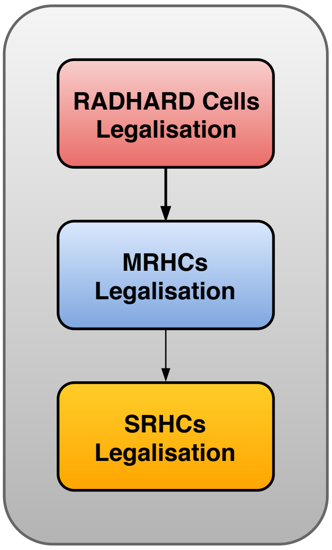


Figure 4.4: Radiation Hardening Legalisation Stages

As mentioned in Section 3.2.1, Abax is a greedy, sequential Displacement-driven Legaliser based on Abacus, performing multiple legalisation runs for multiple cells orders, based on cells' x-coordinate, *i.e.* increasing, decreasing and centre-outwards, resulting in a legal placement with the best cost across the orders. Moreover, Abax supports both SRHC and MRHCs and works in stages, legalising first the MRHCs, whose legalisation is harder compared to SRHCs, and continues with the legalisation of SRHCs.

The Abax legaliser also support blockages, the positions of which create core row cuts and divide core area rows into subrows, which represent the legal placement cell positions. Thus, each core area row contains a set of subrows, represented by their leftmost x-coordinate and width.

In our RADHARD version of Abax, we have added one more stage prior to the MRHCs and SRHCs legalisation stages, the RADHARD cells legalisation stage, which aims to place triplet FFs in positions which satisfy their spacing constraints. Therefore, the RADHARD Legalisation algorithm consists of a total of three stages, as shown in Fig. 4.4:

1. **RADHARD Cells Legalisation**, finding the best, in terms of displacement, legal position for each RH cell, satisfying user-specified minimum spacing constraints among the TMR FFs triplet, and fixing it to this position to guarantee no spacing violations,
2. **MRHCs legalisation**, and fixing them after its completion,
3. **SRHCs legalisation**

4.1.4 Satisfy Spacing Constraints

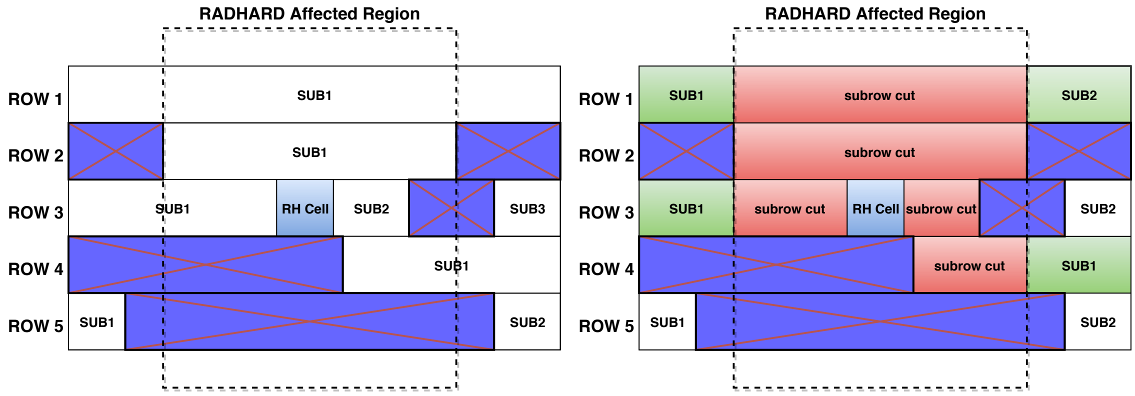
As mentioned before, the goal of Min-bounded RADHARD Legalisation is to satisfy a minimum spacing constraint among the TMR replicas. Each member of a TMR group must be placed with a distance, at least equal to the specified minimum spacing constraint, among the other members of this group.

One way to ensure the minimum distance among the TMR FFs triplet is to "bloat" or "inflate" the already legalised member of a TMR group. Cell Inflation is a technique used to reduce whitespace or even routing congestions. During inflation, we temporarily increase the size of the cell, making it occupy more area than before (cell upscaling). In our context, cell inflation can be used to inflate the already legalised members of a TMR group that way their new width and height will be increased by current TMR group's minimum spacing constraint in both directions (left and right, top and bottom respectively).

However, cell inflation for RADHARD presents several drawbacks. It has to be performed prior to legalisation leading to space all FFs, instead of only the FFs consisting each TMR group. Hence, a lot of area would be wasted, leading to worse placement and thus worse PPA results. Further on, before the next step, *i.e.* MRHC legalisation, the previously inflated TMR group cells would have to be deflated. Due to these difficulties, instead of directly bloating TMR group cells, we implemented an indirect cell inflation approach.

The indirect way we achieve cell inflation exploits the Abax's feature of diving the core area into a set of rows and subrows, where their x-coordinates and width represent the latter. Thus, instead of changing cell size, we perform cell inflation by changing the x-coordinate and width of the subrows affected by the inflated cell. This method fills the parts of the affected subrows, which lie inside a region around the inflated cell, called RADHARD Affected Region. The latter spreads around the inflated cell in a distance equal to the minimum spacing constraint specified for the TMR group that the inflated cell belongs. The update of the affected subrows creates white space, where other cells within the same group cannot be placed. This process is performed during the legalisation of each RADHARD cell to the already legalised members of the same TMR group, preventing the legaliser from placing another cell of the same group within the spacing constraint range, *i.e.* within the RADHARD Affected Region, as these positions are presented as not available. When a RADHARD cell is placed, the subrow fills are reverted, to place the next one.

Fig. 4.5 presents a mock-up example, including all the possible cases, of updating the subrows of the affected rows. In Fig. 4.5a is presented the subrows state before updating the subrows affected by the RADHARD inflation, while their up-



(a) Before updating affected subrows

(b) After updating affected subrows

Figure 4.5: Subrows Fill Cases

dated state is presented in Fig. 4.5b. The black rectangle represents the RADHARD Affected Region, *i.e.* the area influenced by the cell inflation defining the minimum spacing constraint and the purplish crossed rectangles represent already placed blockages or fixed cells. With blue colour is the cell that must be inflated and in pinkish is the white space area which will be affected by the inflation, while in green colour are the subrows that their x-coordinate or width have been updated due to the inflation.

There are five cases to update the subrows, *i.e.*

1. A subrow may be divided into smaller subrows (ROW1 - SUB1 and SUB2)
2. A subrow may be removed, as it is occupied entirely by the RADHARD Affected Region (ROW2 - SUB1)
3. The width of a subrow may be decreased (ROW3 - SUB1)
4. The width and the x-coordinate of a subrow may be changed (ROW4 - SUB1)
5. The inflation may not perform any change to the subrows (ROW5 - SUB1 and SUB2)

In general, there may be combinations of these five cases. For example, in ROW3 case 3, case 2 and case 5 occur at the same time, *i.e.* case 3 occurs for SUB1, case 2 occurs for SUB2, and case 5 occurs for SUB3.

4.1.5 Create the list of the RADHARD Components

The array consisted of the RADHARD components is created by iterating through the TMR groups the user created. However, this depends strongly on the order the user created them. Thus, to reduce any randomness, we need to sort the TMR groups based on some criteria. As mentioned, after the completion of legalisation of the RADHARD cells, the latter are fixed into their new legal positions before proceeding to the legalisation of the non-RADHARD cells, to guarantee that the spacing constraints among the TMR FFs triplet will not be violated. Thus, for the RADHARD legalisation process, a reasonable ordering is based on the minimum spacing constraint of each RH group, since the larger the spacing constraint among a TMR FFs triplet is, the less the available legal positions and thus the harder for legaliser to place them. In general case, each TMR group can have different minimum spacing constraints. However, in most cases, the TMR groups have the same spacing constraint. So to resolve the sorting equations, we sort them based on the group's x-coordinate. As a group's x-coordinate can be determined:

- Leftmost group member x-coordinate (Group's Min X-Coordinate)
- Rightmost group member x-coordinate (Group's Max X-Coordinate)
- X-coordinate of group's centre of mass (Group's Mean X-Coordinate)

Two mutually exclusive sorting methods of TMR groups have been investigated:

1. Based on Group's Mean X-Coordinate, and
2. Based on group's ordered x-coordinate, *i.e.*:
    - Group's Min X-Coordinate in increasing order,
    - Group's Max X-Coordinate in decreasing order,
    - Group's Mean X-Coordinate in centre-outwards order

The selection of the sorting method is performed by the user. Note that the cells within the groups are also sorted based on their x-coordinate, creating the list of the RADHARD components.

4.1.6 RADHARD Legalisation Algorithm

As mentioned in Section 4.1.3, our RADHARD Legalisation algorithm is based on Abax, which supports both MRHCs and SRHCs. As described in Section 3.2.1, Abax is sequential, performs multiple orders legalisation, *i.e.* increasing, decreasing and centre-outwards, and selects the one with the best cost. Our RADHARD Legaliser takes place legalising one cell at a time from the list of RADHARD components, applying first a sorting method to the existed TMR groups and then sorting the cells within each group, as described in Section 4.1.5. Hence, after sorting the TMR groups, the notion of Abax orders has extended to the list of the RADHARD cells. For each order, RADHARD legalisation is performed, and the best result is obtained.

Algorithm 1 presents the top-level RADHARD Legalisation algorithm. RADHARD Legalisation legalises cells one at a time. It takes as input parameters:

- The user-defined sorting flag *SRT*, which selects between group's mean x-coordinate and group's order-based x-coordinate,
- The set of legalisation orders *O*, typically increasing, decreasing and centre-outwards order, and
- The set of placement subrows S_R.

If *SRT* is set to mean x-coordinate order, only one sorting of the groups is necessary, lines 3-5, otherwise sorting is per order, lines 7-9. For each cell in C_{RH}, its group is identified, line 11, the current S_R is stored, line 12, and the group's already legal members are indirectly inflated, by the call to function *RH_inflation*(),

Algorithm 1 RADHARD Legalisation Algorithm

```
Input: Sorting Method Flag (SRT),
    Legalisation Orders (O), Placement Subrows ($S_R$).
Output: Minimum Displacement Cost, RADHARD Legal
    Placement, satisfying TMR FFs triplet minimum spacing constraints.
 1: best_cost = $\infty$;
 2: $S_{RBEST}$ = $S_R$;
 3: if (SRT == group mean x-coordinate) then
 4:     $C_{RH}$ = sort_RADHARD_cells_by_mean_x();
 5: end if
 6: for each order in O do
 7:     if (SRT == order-based x-coordinate) then
 8:         $C_{RH}$ = sort_RADHARD_cells_by_order(O);
 9:     end if
10:     for all (cells c in $C_{RH}$) do
11:         group = get_RADHARD_group(c);
12:         $S_{ROLD}$ = $S_R$; // store subrows //
            // indirect cell inflate, filling affected subrows //
            // inside the RADHARD Affected Region //
13:         $S_R$ = RH_inflation(group, $S_R$);
14:         total_cost = total_cost + legalise_RH_cell(c, $S_R$);
15:         if (total_cost < best_cost) then
16:             best_cost = total_cost;
17:             $S_{RBEST}$ = $S_R$;
18:         end if
19:         $S_R$ = $S_{ROLD}$; // revert subrows //
20:     end for
21:     best_legal_placement = set_best_placement($S_{RBEST}$);
22: end for
23: return best_legal_placement;
```

line 13. The latter creates white space around the legalised members of the specified group, as described in Section 4.1.4. Then, cell c is legalised, line 14, and subrows are restored from S_{ROLD}, line 19. For each order explored, the order producing best legalisation cost, *i.e.* minimum total displacement, is stored and returned, line 21.

Algorithm 2 legalise_RH_cell()

```
Input: Current RADHARD Cell (C), Placement Subrows ($S_R$).
Output: Minimum Displacement, RADHARD Legal Position.
B = placement rows;
best_cost = $\infty$;
best_position = $\emptyset$;
nearest_row = get_nearest_row(C, B);
(best_cost, best_position) = check_neighbouring_row (C, B, nearest_row, up, $S_R$);
(best_cost, best_position) = check_neighbouring_row (C, B, nearest_row, down, $S_R$);
place_cell_at_best_position(C, best_position);
```

Our RADHARD Legalisation algorithm is sequential finding for each cell the best, in terms of displacement, legal position. This process is presented in Algorithm 2. It first identifies the nearest row, line 4, and then checks a set of adjacent rows above and below, based on the Abax row search bound B [21], lines 5-6. Finally, the cell is placed and fixed in the position with the best legalisation cost, line 7.

Algorithm 3 check_neighbouring_row()

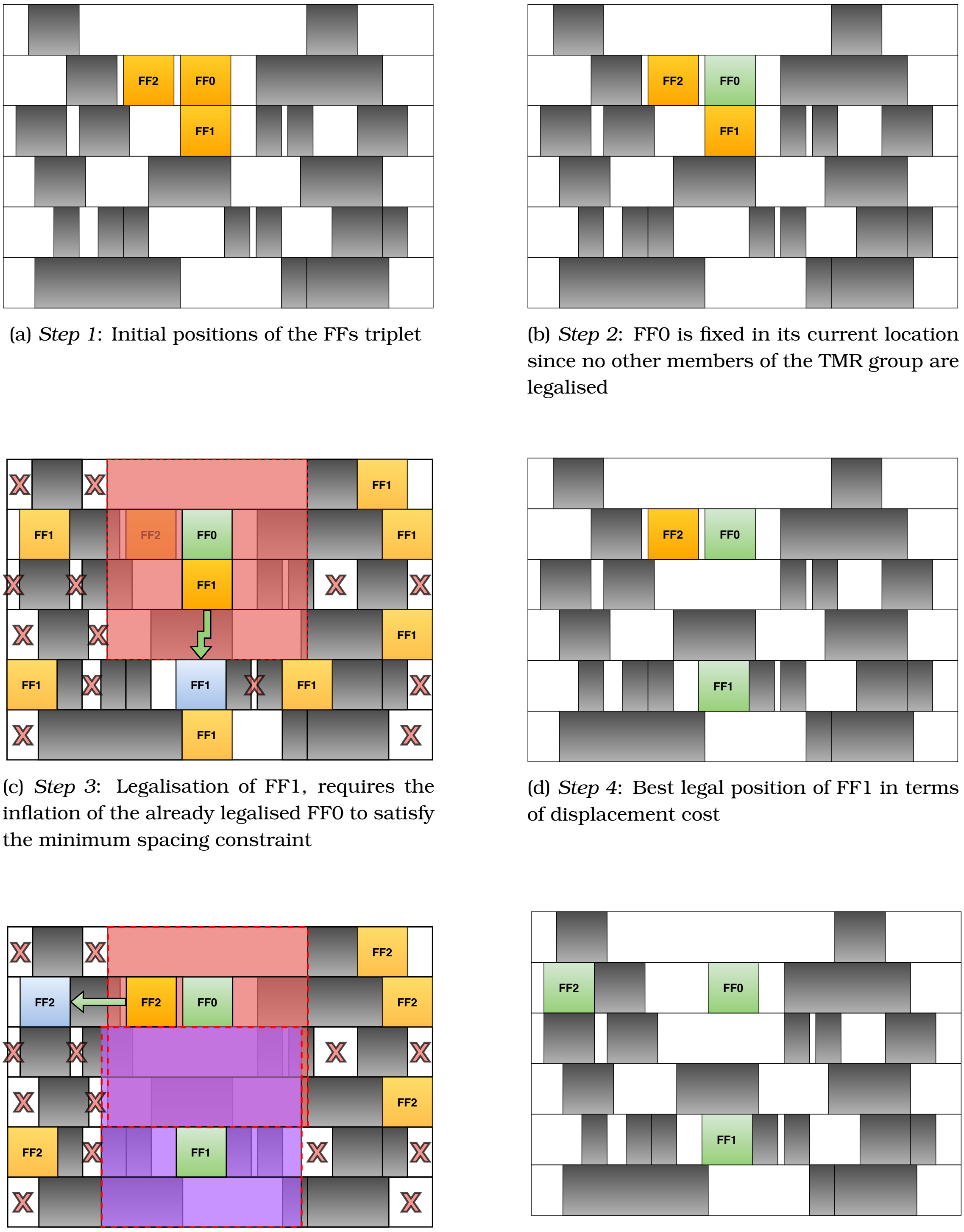
Input: Current RADHARD Cell (C), Row Bound (B),
Nearest Row (NR), Up/Down Flag for Row Checking (Dir),
Placement Subrows (S_R).

Output: Minimum Displacement Cost for RADHARD Legal Cell
Placement and Position.

```
 1: dist_from_nearest_row = 0;
 2: check_row = NR;
 3: while (check_row within bound B) do
 4:    subrows_fit = get_subrows_fit_cell(C, check_row, $S_R$);
 5:    for all (subrows s in subrows_fit) do
 6:       cell_new_position = place_cell_at_subrow(C, s, cell_height);
 7:       if (cell_new_position != legal position) then
 8:          continue;
 9:       end if
10:       cost = compute_placement_cost(C, cell_new_position);
11:       if (cost $<$ best_cost) then
12:          best_cost = cost;
13:          best_position = cell_new_position;
14:          B = update_row_bound(B);
15:       end if
16:    end for
17:    dist_from_nearest_row + +;
18:    if (Dir == up) then
19:       check_row = NR - dist_from_nearest_row;
20:    else
21:       check_row = NR + dist_from_nearest_row;
22:    end if
23: end while
24: return best_cost, best_position;
```

Algorithm 3 describes function *check_neighbouring_row()*. Its inputs include the cell C, the row bound B, the nearest row NR and the direction Dir, which is either up or down. Nearest row distance, *dist_from_nearest_row* is initialised to 0 and the current row to be checked, *check_row* is set to NR, lines 1-2. In the main while loop, we examine the complete set of neighbouring rows one at a time. For row *check_row*, we obtain the candidate subrows, where the cell fits, *i.e.* *subrows_fit*, line 4. Then, for each candidate subrow, we perform tentative placement of the cell C, and compute the placement cost, if it is legal, lines 6-10. According to [22] a tentative placement may be illegal, in case a RADHARD cell is MRHC. In this case, the cell can fit in a subrow of the current row, but cause overlaps in adjacent rows. Next, if the cost of the current placement is better than the already found best cost, we accept it, updating the cell's position and the row search bound B as described in [22], lines 11-15. After exploring all candidate subrows, we update the checking row, until the search bound B is reached in the main while loop, lines 17-23. At the end of the neighbouring rows exploration, the best cost and position of the cell are returned, line 24.

Fig. 4.6 presents a contrived example of legalising the FFs triplet of a TMR group. Initially, the three FFs violate their spacing constraints since they are in close proximity, Fig. 4.6a. So, let suppose that we start with the legalisation of FF0. Since no other members of the same TMR group are legalised, there is no need for inflation, and we fix FF0 in its nearest legal position, Fig. 4.6b. Next, we proceed to the legalisation of FF1. Since FF0 is already legalised it must be inflated by updating the affected subrows' x-coordinate and width, as described in Section 4.1.4, in order to ensure that FF1 will no be placed inside the red rectangle, Fig. 4.6c. Next, legaliser finds all possible legal positions that FF1 fits and finally fix it in the position with the best cost in terms of displacement, Fig. 4.6d. Lastly, in Fig. 4.6e we continue with the legalisation of FF2 in the same way the two previous cells were legalised by inflating first the already legalised cells, *i.e.* FF0 and FF1. The set of subrows which will be updated is the *union* of the subrows that are affected by the inflation of each cell individually. After this process the TMR FFs triplet is spread away, as shown in Fig. 4.6f, satisfying the minimum spacing constraint and reducing the probability that multiple members of the TMR group will be affected by the same particle strike.



(a) *Step 1*: Initial positions of the FFs triplet

(b) *Step 2*: FF0 is fixed in its current location since no other members of the TMR group are legalised

(c) *Step 3*: Legalisation of FF1, requires the inflation of the already legalised FF0 to satisfy the minimum spacing constraint

(d) *Step 4*: Best legal position of FF1 in terms of displacement cost

(e) *Step 5*: Legalisation of FF2, inflating first cells FF0 and FF1

(f) *Step 6*: Final legal placement of the FFs triplet of a TMR group satisfying the minimum spacing constraint among them

Figure 4.6: TMR group legalisation mock-up example

4.2 MinMax-bounded RADHARD Legalisation

Legalisation process takes as input the solution of GP and resolves any cells overlaps satisfying at the same time any specified constraints. Thus, the quality of the legalisation solution strongly depends on the solution produced by GP. GP, assigning positions to each cell, may place triplet cells far away, leading to worse PPA results. Hence, legalising the TMR groups satisfying only their minimum spacing constraint is not enough. To achieve better PPA results, we investigated the satisfaction of another spacing constraint, which does not allow legaliser to place cells far way, *i.e.* a maximum spacing constraint.

4.2.1 Maximum Spacing Constraint Handling

As mentioned before, max spacing constraint aims to force the Legaliser to avoid placing cells of the same TMR triplet far away affecting the power, performance and area of the circuit. So supporting a maximum spacing constraint for each TMR group creates a boundary outside of which members of the same TMR triplet can not be placed.

Supporting both minimum and maximum spacing constraints creates two boundaries around the inflated cell, i.e. the minimum one forcing Legaliser to place cells beyond it and the maximum one forcing cells of the same TMR triplet to be placed inside it. So, these two boundaries create a ring signifying the legal region Legaliser is permitted to place another member of the same TMR group, as shown in Fig. 4.7.

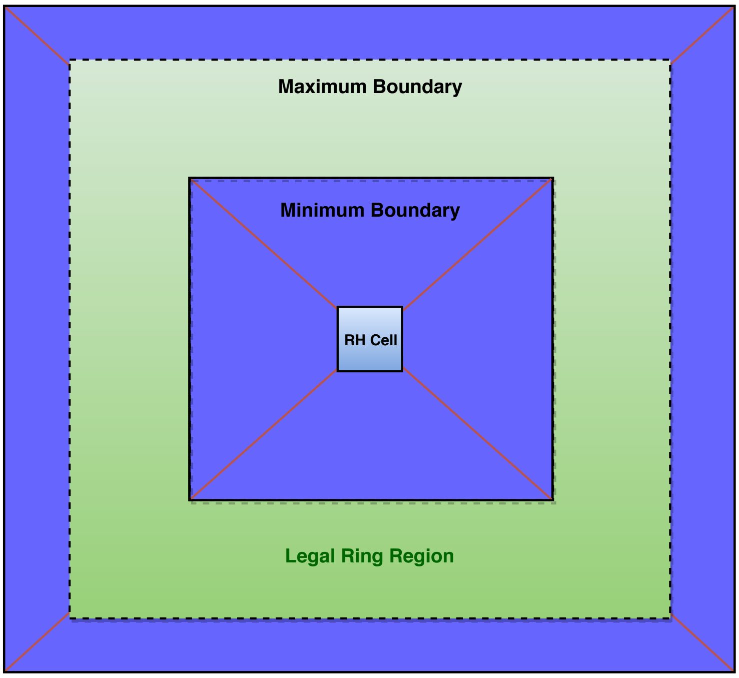


Figure 4.7: Minimum and Maximum Spacing Constraints create a ring region legal for the legalisation of next member

To satisfy both minimum and maximum spacing constraints for a TMR group, we modified the *RH_inflation* procedure we described in Section 4.1.4. The maximum boundary forces Legaliser not to place any other cells of the same triplet outside this region. Thus, the *RH_inflation* procedure is performed, filling all the subrows of the core that are outside this boundary. Since all the core rows are updated during the inflation, there is no need to find the affected rows. So we store the core state before the inflation to be able to restore the state of all core subrows after the RH inflation.

Multiple Cell Inflation supporting both Min and Max Boundaries

Similar to supporting only the minimum spacing constraint, the RH inflation of multiple cells requires the combination of their spacing constraints. The legal region to place a member of a TMR group, created after the RH inflation, is determined by the union of the minimum boundaries and the intersection of the maximum boundaries created for each legal member of the TMR group, *i.e.* the intersection of the individual legal regions. For example, in Fig. 4.8, we have two already legalised members of a TMR group. Thus, before we legalise the third one, we need to inflate these two cells. The *Minimum Boundary 1* and *Maximum Boundary 1* signify the two boundaries due to the inflation of *RH Cell 1*, while the *Minimum Boundary 2* and *Maximum Boundary 2* the boundaries due to the inflation of *RH Cell 2*. As we can see, the final legal region is the intersection of the legal regions determined after the inflation of each cell.

Maximum Spacing Constraint is a Soft Constraint

In contrast to the minimum spacing constraint, the maximum one is a soft constraint, meaning that its satisfaction is not critical for the functionality of the circuit but aims to the optimisation of the solution's quality. Thus, the max spacing constraint is initialised to 2 × *minimum_spacing_constraint*, to ensure that there is enough space for the next cell to be placed. However, as mentioned before, the final maximum boundary inflating more that one cell is the intersection of the separate maximum boundaries. Thus, the resulting legal region is reduced drastically, as we can see in Fig. 4.8. So, it is quite possible that Legaliser will not be able to find a legal position to place the next cell. Since the maximum spacing constraint is a soft one, in order to resolve this situation the maximum spacing constraint of the corresponding TMR group is iteratively increased until Legaliser finds a legal position for the current cell of the TMR triplet.

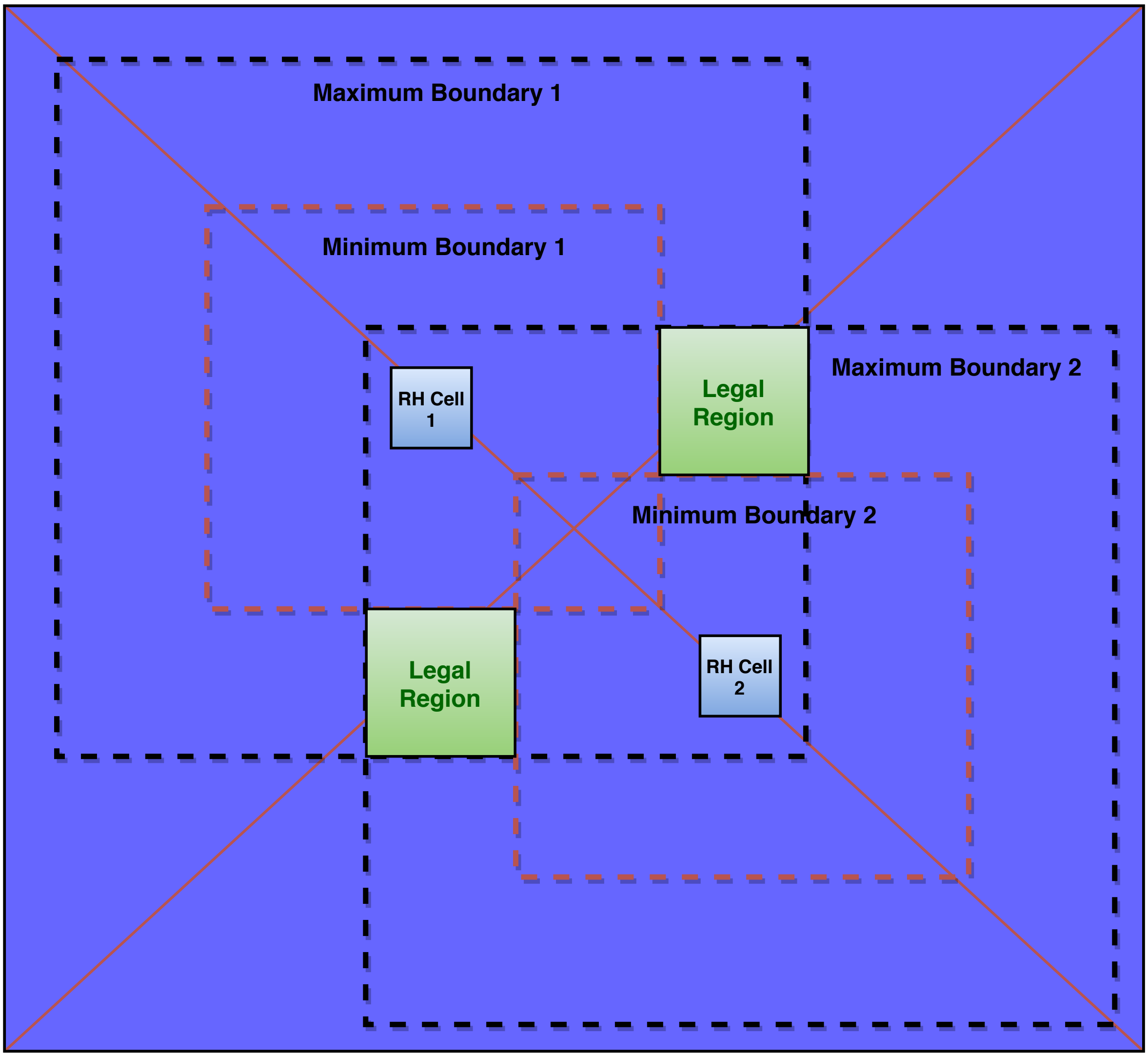


Figure 4.8: Multiple Cells Inflation: The final boundary is the union of the minimum spacing constraints and the intersection of the maximum spacing constraints

4.3 HPWL-driven Legalisation

The previous section presented a method to place cells of the same TMR group close enough, satisfying at the same time the minimum spacing constraint, by adding another one, *i.e.* the maximum spacing constraint. However, the latter is applied only among each TMR group members and not for non-RADHARD components. As described in Section 4.1.3, the RADHARD cells legalisation is before the legalisation of the MRHCs and SRHCs, and they are fixed into their new positions. Thus, placing TMR triplet members close enough may cause non-RADHARD components to be placed far away from their original position leading to worse PPA results.

An efficient way to improve the circuit's performance is to take into account during the evaluation of a cell move the total wire length (TWL) of the circuit. The TWL in cost function leads Legaliser to place cells closer, aiming to reduce the entire length of the wires across the circuit. This approach can result in better PPA results since it is applied during the legalisation of all cells.

4.3.1 Total Wirelength Calculation

As mentioned before, inserting TWL into legalisation's cost function can have a significant effect on the solution's quality. Minimising total wire length implies a less congested, or timing-satisfied placement. The TWL is measured by the total rectilinear minimum Steiner tree (RMST) wire length.

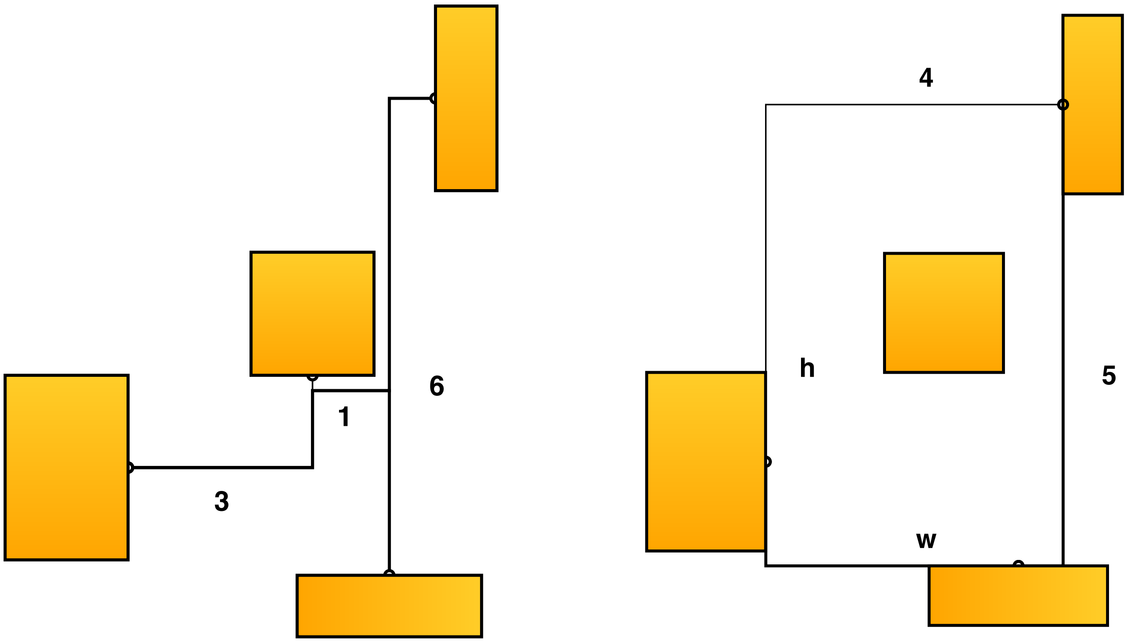
However, the latter is very expensive to compute. Since a cell's move cost needs to be evaluated frequently during the legalisation process, it is not affordable any costly computation on the cost function. Thus, an estimation can be used for the calculation of total wire length without affecting the legalisation result if there is a good correlation between the actual wire length and the estimated one. The half perimeter of the bounding box of a net, called Half-Perimeter WireLength (HPWL), is a reasonable estimation for actual wire length In Fig. 4.9 is shown that HPWL is a very close estimation for wire length compared to RMST. Specifically, for two-terminal and three-terminal nets, HPWL is exactly equal to the RMST wire length. HPWL is so far the most commonly used cost function in the placement procedure.

4.3.2 Updated Cost Function

Adding HPWL into cost function leads Legaliser to place cells closer, aiming to minimise the total circuit WL. As mentioned in Section 3.2.1, Abax is a displacement-driven Legaliser, and as cost function is the displacement cost of the current move. Since Abax allows already legalised cells to shift during the legalisation of current cell the displacement cost can be: (i) the displacement of the current legalising cell (single-cell displacement), (ii) the displacement of all moving cells (multi-cell total displacement), or (iii) the mean total displacement (multi-cell mean displacement).

In this approach, we modified Abax's cost function to consider both displacement and total HPWL (THPWL) of the circuit. Thus, the updated cost function is:

$$\boxed{cost = w \times C_{disp} + (1 - w)) \times THWPL} \tag{4.1}$$



(a) RMST = 10

(b) HPWL = w + h = 9

Figure 4.9: Comparison of Rectilinear Minimum Steiner Tree (RMST) and Half-Perimeter WireLength (HPWL)

, where w is a user specified weight factor determining the impact of C_{disp} and $THPWL$ in total cost of a tentative move.

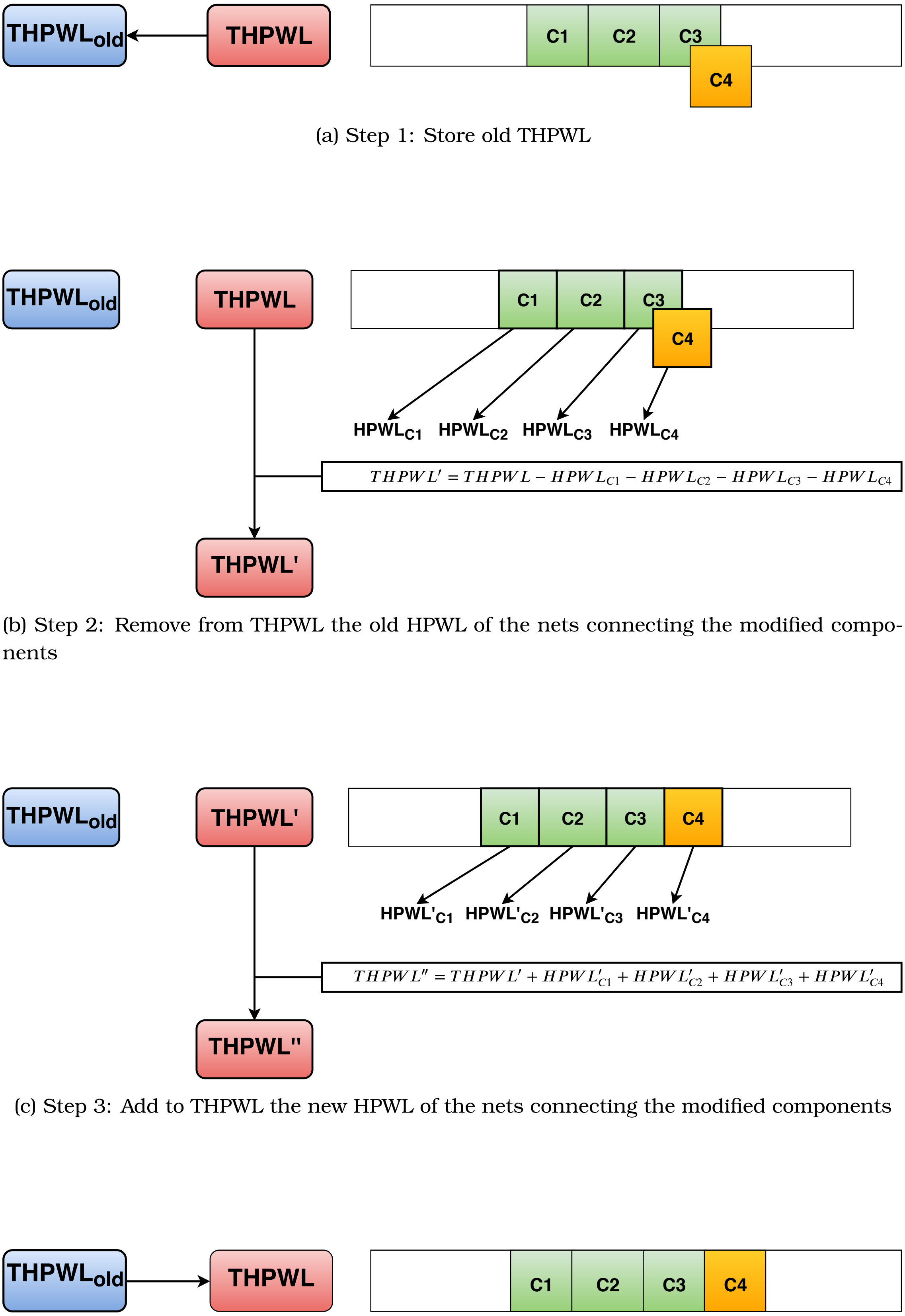
Respectively the bound cost function has to be modified to take into consideration the HPWL. The bound cost is quantised to core rows and, thus, it is not feasible to add the THPWL in its cost function, since in this case, legaliser is going to check all the core rows exhaustively. So, in the bound cost function instead of THPWL, we used the change in THPWL, The bound cost function is updated as follows:

$$\boxed{bound\_cost = p \times C_{disp} + (1 - p)) \times \Delta_{THWPL}} \tag{4.2}$$

, where p is also a user specified weight factor determining the effect of C_{disp} and Δ_{THPWL} in bound cost.

4.3.3 Update of THPWL

As mentioned before, the computation of the THPWL is performed for each tentative cell move. Considering that, updating the new THPWL by computing it again, taking into account all the nets of the circuit is not feasible. Thus, its update must be performed incrementally. Since, during a tentative move, only a few cells are moved, we can calculate the new THPWL by updating the HPWL only for the nets connecting moved cells. The following example explains the incremental update of THPWL.


THPWL_old
THPWL
C1
C2
C3
C4
HPWL_C1
HPWL_C2
HPWL_C3
HPWL_C4
THPWL' = THPWL − HPWL_C1 − HPWL_C2 − HPWL_C3 − HPWL_C4
THPWL'
HPWL'_C1
HPWL'_C2
HPWL'_C3
HPWL'_C4
THPWL'' = THPWL' + HPWL'_C1 + HPWL'_C2 + HPWL'_C3 + HPWL'_C4
THPWL''

(a) Step 1: Store old THPWL

(b) Step 2: Remove from THPWL the old HPWL of the nets connecting the modified components

(c) Step 3: Add to THPWL the new HPWL of the nets connecting the modified components

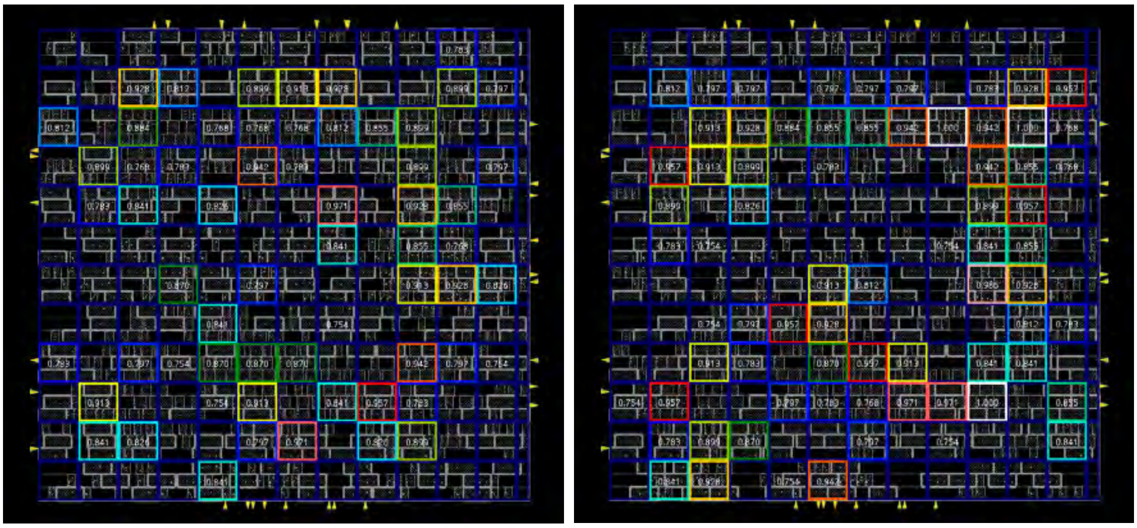
(d) Step 4: Restore the old THPWL before continuing with cell's next tentative move

Figure 4.10: Steps of incremental THPWL update

Fig. 4.10 presents with an example the steps followed to update the *THPWL* during the evaluation of cell's tentative move. Suppose the current cell being legalised is *C4* and evaluate the cost of placing it into a row. Let also suppose that in this row three cells are already legalised and the placing *C4* into this row is going to shift them. Before placing *C4* into this row we store the current value of *THPWL*, Fig. 4.10a. This is needed in order to restore it later. Following, for each moved cell during this tentative move, *i.e. C1*, *C2*, *C3* and *C4*, we compute their connected nets HPWL and subtract them from *THPWL*, resulting in THPWL', Fig. 4.10b. Next, we place *C4* into the row, shifting at the same time the cells C1, C2 and C3. For each one, we compute their nets' updated HPWL and add it into the total, resulting in *THPWL"*, Fig. 4.10c. Considering the new value of total HPWL, *i.e. THPWL"*, the cost of this tentative move is computed. In case the latter is better than the already best one, we keep it in order to restore to it after exploring all the row search bound. Finally, the old value of THPWL, *i.e.* the one stored in Step 1, is restored, in order to evaluate next tentative move for *C4*.

4.4 Timing-driven Legalisation

As presented in the previous section, HPWL seems a suitable metric to optimise the legalisation result. It reduces the total wire length (TWL) of the circuit, resulting in the reduction of wire delays. *i.e.* the delay spent during the transition of a signal across the wires. However, reducing the total wire length of the circuit may lead to an increase in its density. As shown in Fig. 4.11, performing HPWL-driven legalisation leads to a denser placement compared to Displacement-driven. This density overhead can make the routing of the design, *i.e.* the connection of the circuit instances, more challenging.



(a) Displacement-driven RADHARD Legalisation Density Map

(b) HPWL-driven RADHARD Legalisation Density Map

Figure 4.11: Density comparison between Displacement-driven and HPWL-driven RADHARD Legalisation

An alternative way to improve the quality of results (QoR) is to legalise cells concerning the circuit performance. The IC placement layout must not only guarantee the absence of cell overlaps and routability. It also has to meet the design's timing-constraints, *i.e.* *setup* (long-path) and *hold* (short-path) constraints. The optimisation process to meet these constraints is often called *timing closure* [1].

The circuit delay consists of the propagation delay in logic gates, called gate delay, and the delay across the wires called wire delay. For many years, the gate delay was the most significant contributor to circuit delay, while the wire delay was negligible. Therefore, the cell placement did not have a vital impact on circuit performance. However, the technology downscaling the past few decades overturned this situation. Nowadays, small nanometer technologies made the impact of wire delay critical on circuit performance, creating the need for timing-optimised placement and routing.

Timing optimisations adjust propagation delays across the circuit gates, aiming to satisfy timing constraints. The latter, include the setup constraints, which specify the time duration a data input signal should be stable before the clock edge for each sequential element, *i.e.* FF or latch, and hold constraints, indicating the amount of time a data input signal should be stable after the clock edge at each storage element, as shown in Fig. 4.12.

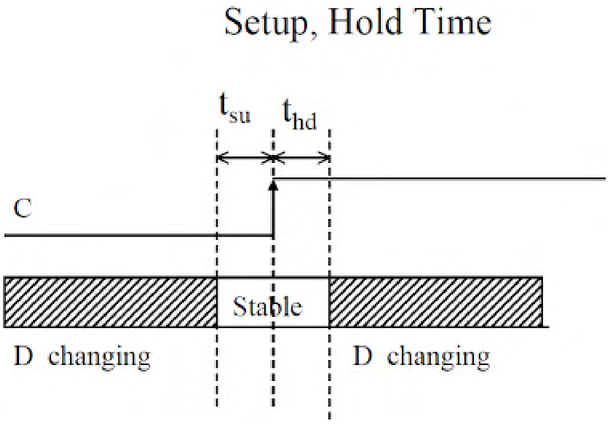


Figure 4.12: Setup and Hold latching window. The data should be stable at least t_{su} and t_{hd} before and after the clock edge.

The satisfaction of hold constraints, ensure that signal transitions do not occur too early. Thus, a violation of them, occurring when a signal path is too short, allows a receiving FF to capture the data signal at the current cycle instead of the next one.

On the other hand, setup constraints ensure that no signal transitions occur too late. Checking whether a circuit meets the setup constraints requires estimating how long signal transitions will take to propagate from one sequential element to the next one. Such delay estimation is based on Static Timing Analysis (STA). STA estimates the delay by propagating actual arrival times (AATs) and required arrival times (RATs) to the pins of each gate. It investigates timing violations by tracing out critical paths in the circuit, responsible for these timing failures.

STA results are used to estimate the importance of each cell and each net in a particular layout. A key metric for a pin of a gate g, which is a timing point, is the timing slack. Timing slack is the difference between AAT and RAT in g:

$$\boxed{slack(g) = RAT(g) - AAT(g)} \tag{4.3}$$

Positive slack indicates that timing is met, *i.e.* the signal arrives before it is required, while a negative slack indicates a timing violation, *i.e.* the data arrives after its required time. Thus, algorithms for timing-driven placement focus on timing slack values.

4.4.1 Static Time Analysis Background

In STA, a combinational logic circuit is represented as a *directed acyclic graph* (DAG). Fig. 4.13 illustrates a logic circuit consisted of four logic gates - x, y, z, w - three inputs - a, b, c - and one output f. Each input is annotated with a time, at which the signal transitions occur relative to the start of the clock cycle. In Fig. 4.13 are also illustrated the gate and wire delays. For example, the gate delay, *i.e.* the delay from the input to the output, of inverter x is 1 unit, and the wire delay from input b to the input of x is 0.1 units.

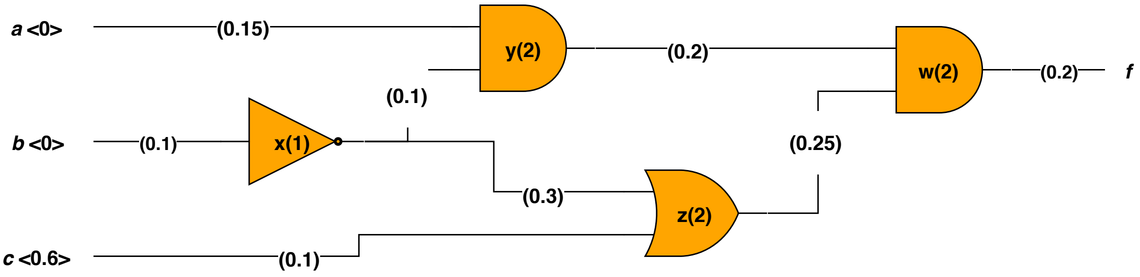


Figure 4.13: Combinational logic circuit with three inputs a, b, c, annotated with the times at which the signal transition occurs in brackets, and four logic gates, whose delays are presented in parentheses. Wire delays are also given in parentheses

As mentioned before, in STA, each circuit is represented by its corresponding DAG. A common representation follows the *gate node convention*. This convention introduces one node for each input and output, as well as for each logic gate. Also, a source node is introduced with a directed edge to each input. Each node representing a logic gate is labelled with its gate delay. For example, the node x has the label 1. The directed edges from the source node to the inputs are labelled with transition times, while the directed edges between the nodes of logic gates are labelled with the corresponding wire delays.

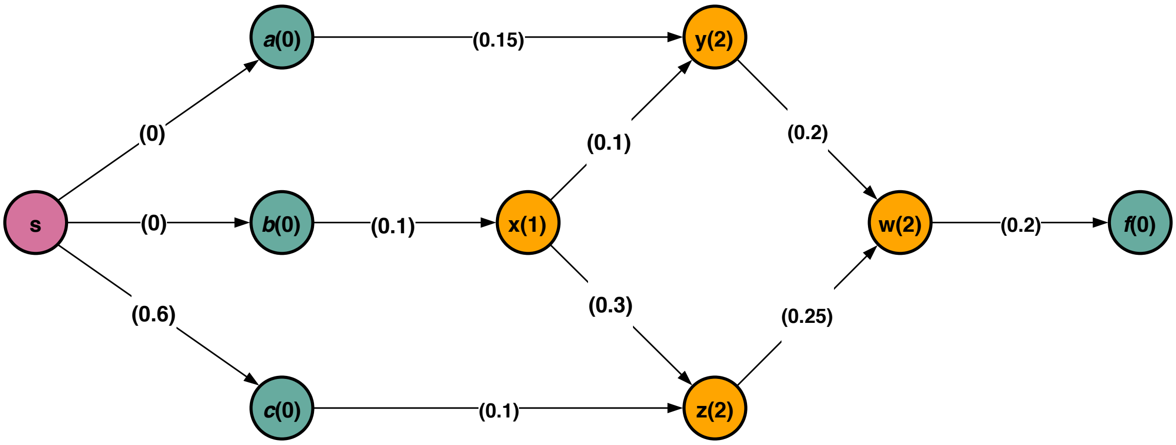


Figure 4.14: DAG representation of circuit in Fig. 4.13 using node convention

In a circuit, the transition time at a given node v of its DAG measured from the beginning of the clock cycle is called actual arrival time (AAT). By convention, this is the arrival time at the output of the corresponding node v. In the case in a node converge two paths, the AAT of this node is the largest between the two AATs. Formally, the AAT of a node v, denoted as $AAT(v)$ is:

$$AAT(v) = \max_{u \in FI(v)} (AAT(u) + t(u, v))$$

where $FI(v)$ is the *Fan In* of node v, *i.e.* the set of all nodes from which there exists a directed edge to v, and $t(u, v)$ is the delay on the edge (u,v). For a given circuit, the AATs are determined for its inputs and FFs' outputs and they are propagated forward through the graph.

Fig. 4.15 presents the AAT for each node of the circuit in Fig. 4.13. The computation of the AAT for each node is started from the source node and applying the above equation, is propagated through the DAG till it reaches the output f. For example, $AAT(x) = 1.1$, due to the wire delay from input b (0.1) and the gate delay of inverter x (1). In node y two paths converge, one from input a resulting in $AAT(y)_{a \to y} = 0.15$, and one from node x resulting in $AAT(y)_{x \to y} = 3.2$. As mentioned, for each node, we keep the latest arrival time. Thus, $AAT(y) = 3.2$.

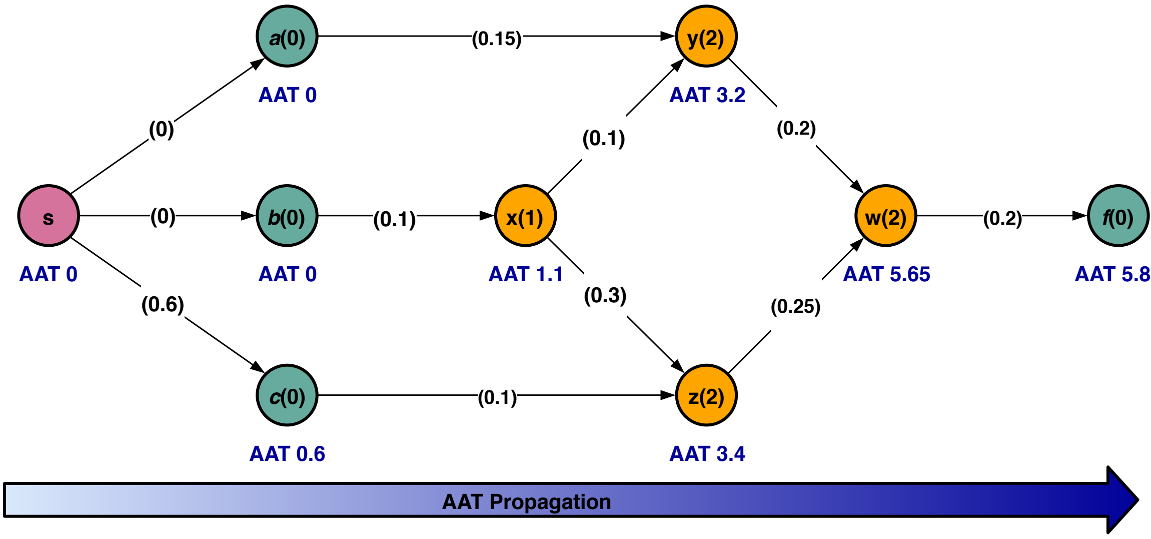


Figure 4.15: Forward propagation of actual arrival times (AATs) across the DAG for each node

Next step in STA analysis is the assignment for each node v of the DAG its required arrival time (RAT), denoted as $RAT(v)$. The $RAT(v)$ is the maximum delay by which the latest data signal transition at a given node v must occur to guarantee the proper operation of a circuit within a given clock cycle. The RATs, in contrast to AATs, are determined for the outputs and FF's inputs of a circuit and are propagated backwards, *i.e.* across the inverted graph. Thus, formally the RAT of a node v is:

$$RAT(v) = \max_{u \in FO(v)} (RAT(u) - t(u, v))$$

where $FO(v)$ is the *Fan Out* of node v, *i.e.* the set of all nodes with a directed edge from v, and $t(u, v)$ is the delay on the edge (u,v).

Fig. 4.16 presents the RAT for each node of the circuit in Fig. 4.13. As mentioned before, initially the RAT is specified for the output f, and its calculation is propagated backwards using the equation above till we reach the inputs. For example, supposing that $RAT(f) = 5.5$ the propagation of RAT resulting in $RAT(w) = 5.3$, $RAT(y) = 3.1$ and so on.

Having the AAT and RAT for each node, the computation of slack follows. The proper operation of the circuit concerning setup constraints requires that for each node $AAT(v) \leq RAT(v)$. Thus, the slack of a node v is defined as:

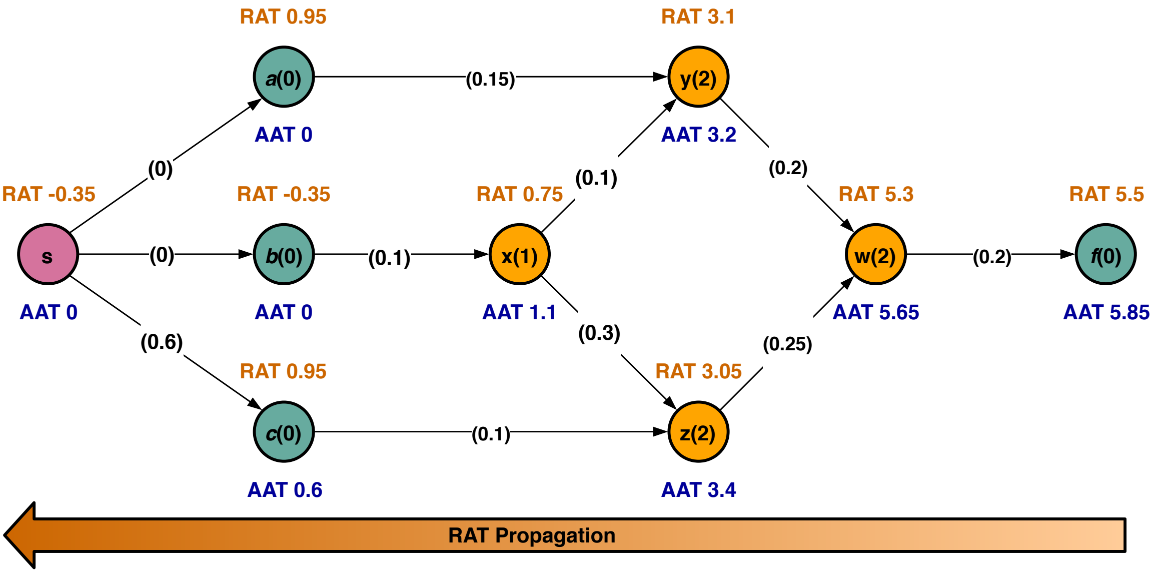


Figure 4.16: Backward propagation of required arrival times (RATs) across the DAG for each node

$$slack(v) = RAT(v) - AAT(v)$$

Slack is a metric indicating the satisfaction of setup constraint for a node v. Thus, a positive or zero slack for node v signifies that for the corresponding node the setup constraint is met. Otherwise a setup violation occurs for node v. The worst slack in the circuit is called *worst negative slack* (WNS) and the nodes with the WNS consist a path called *critical or longest path*.

In Fig.4.17 slack is annotated per node, where green colour indicates a constraint satisfaction while red colour a violation. In Fig. 4.18 the critical path of the circuit is illustrated with $WNS = -0.35$.

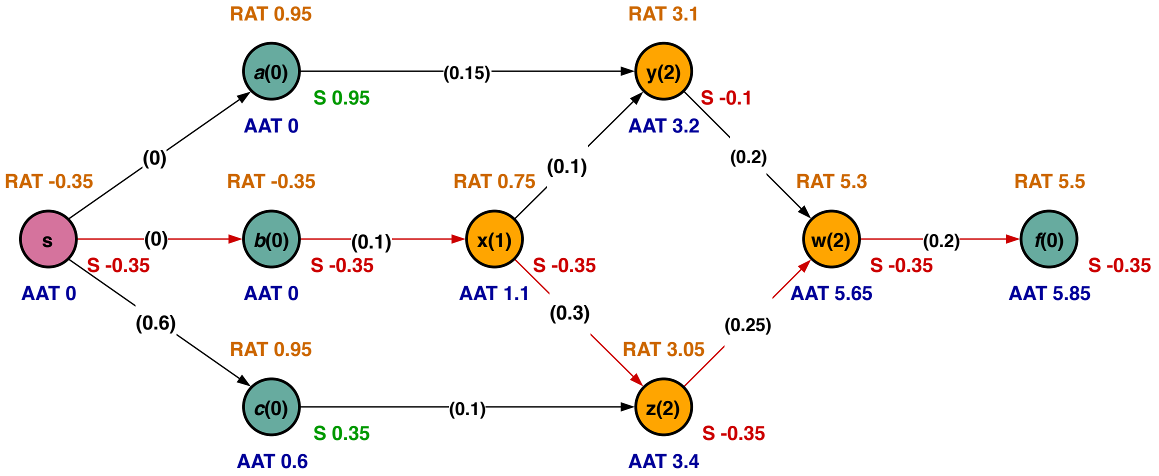


Figure 4.17: Slack computation for each node of the DAG using the Eq. 4.3

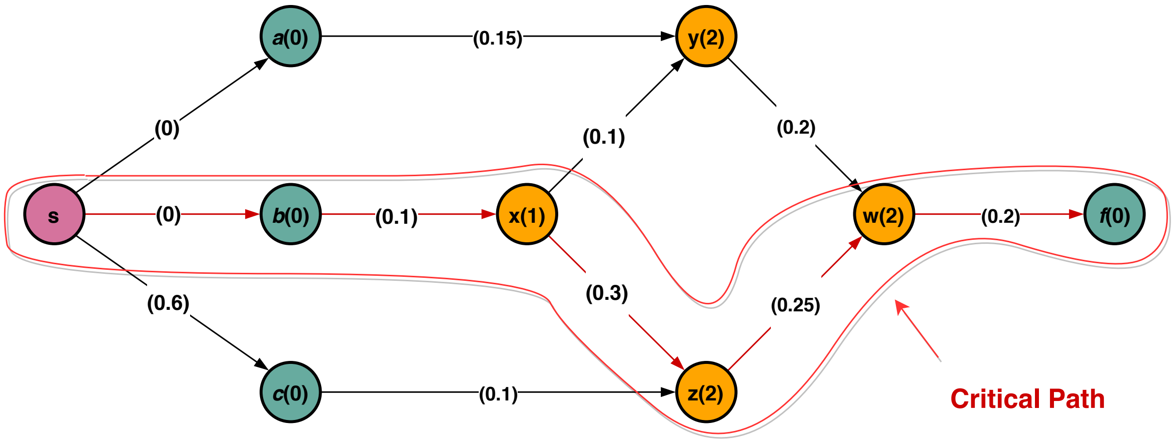


Figure 4.18: The nodes with the worst negative slack form the critical path

4.4.2 Cost Function

Timing-driven legalisation optimises circuit delay, either to satisfy the timing constraints or to achieve the highest possible clock frequency. To modify our legaliser to consider the timing closure is needed to adjust the cost function of a tentative move for each legalising cell. We tried three different cost function, trying to find the most suitable.

First, we used as cost function the worst negative slack (WNS) of the circuit. However, WNS determines the delay only for one path of the circuit, *i.e.* the critical/longest path. So, legalising a cell not included in the critical path the cost for all of its tentative moves will be the same, resulting in placing it into its nearest legal position. Nevertheless, placing a cell into its nearest position may affect the legalisation of the cells belonging in the critical path.

Thus, there is a need for a global view of the circuit's timing. Hence, we introduced as cost function the total negative slack (TNS), which is the sum of all negative slacks in the circuit. Although this is a global metric for the circuit timing, it is still not suitable. The reason is that in case a cell has positive slack, similar to the previous cost function, it will be placed into the position with the minimum displacement from its initial position.

The goal of timing-driven legalisation is (1) to increase negative slack to achieve design's correctness, and (2) to reduce positive slack to recover power and area. Thus, a promising cost function could be the total slack (TS) of the circuit, which is the sum of all positive and negative slacks in the circuit. However, further exploration is needed to choose the proper cost function, since depending on the design characteristics, the impact of the TS as cost function is ambiguous.

Chapter 5

Experimental Results

This chapter presents the results of this thesis. Our radiation-hardened legaliser is implemented in C and is integrated into an existed, under development, EDA tool called ASP [23]. As for the initial placement, an industrial EDA tool is used which performs all the placement phases, *i.e.* global placement, legalisation and detailed placement, but it ignores any spacing constraints. We tested our legalisation algorithm in 11 OpenCores benchmarks, whose characteristics are shown in table 5.1. In our approach, we convert each sequential element of the circuit into a TMR structure, by creating three instances, identical to the original cell, and adding a majority voter. Thus, the impact of our legaliser strongly depends on the percentage of sequential elements in each design, shown in chart 5.1. To evaluate the impact of the spacing constraint, we tested our algorithm for three spacing constraints - 5um, 7um and 10um spacing. For our experiments, we used a 130nm standard-cell library from IHP. Also, we ran them in a server with 8-core Intel Xeon(R) CPU E5-1620 v4 @ 3.50GHz, swap memory of 16GB, physical memory of 15GB, running CentOS 7 (4.20.12-1.el7.elrepo.x86_64).

Benchmarks	# Cells	# Sequential Elems	# Combinational Elems
lpffir	402	80	322
pid	3401	397	3004
openMSP	7348	771	6577
aes	7487	670	6817
aes192	9195	798	8397
aes_ip	9334	910	8424
ldpc	52506	2048	50458
netcard	312037	65965	246072
leon3mp	514411	108803	405608
jpegencode	524446	39583	484863
leon2	701850	149492	552358

Table 5.1: OpenCores Benchmarks Characteristics

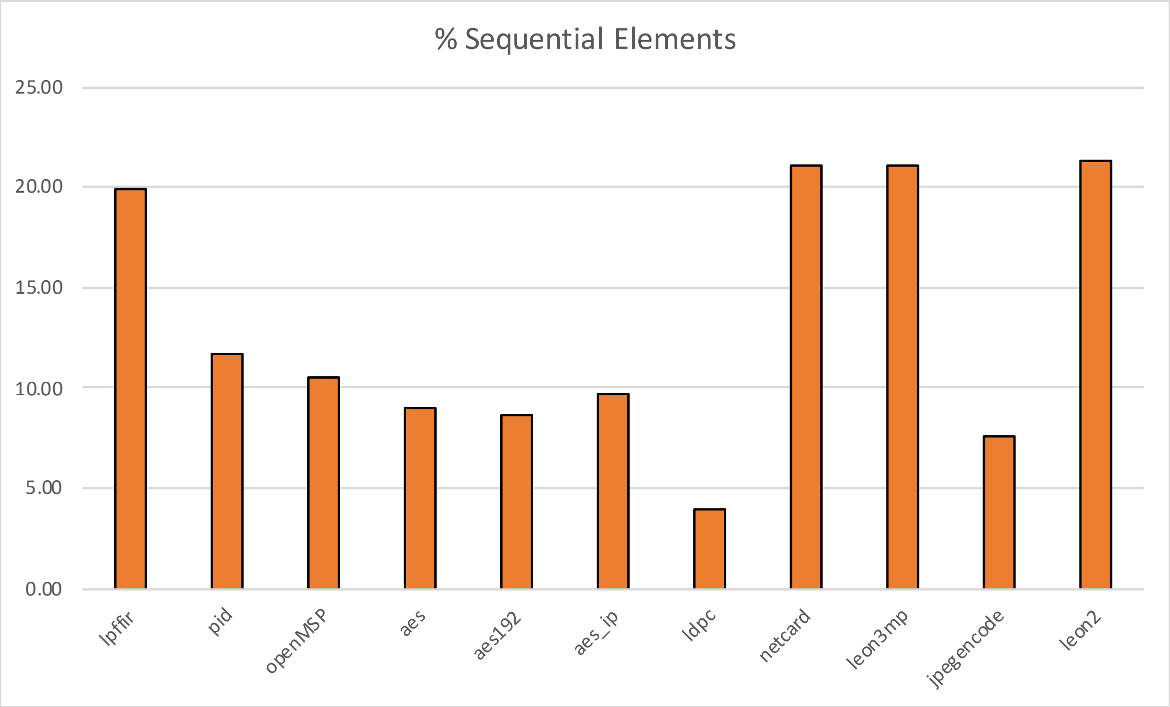


Figure 5.1: Percentage of sequential elements for each design

5.1 Flow

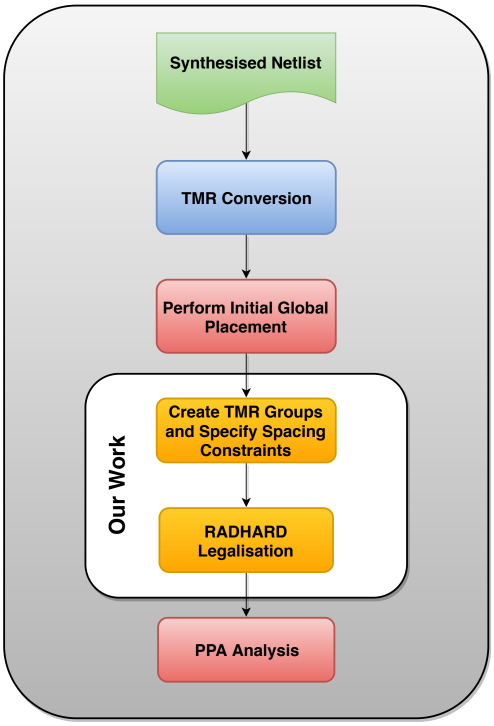

Synthesised Netlist
TMR Conversion
Perform Initial Global Placement
Our Work
Create TMR Groups and Specify Spacing Constraints
RADHARD Legalisation
PPA Analysis

Figure 5.2: Experiments Flow

During our testing, we followed the flow presented in Fig. 5.2. First, we synthesise the Verilog file, producing the corresponding netlist, which has the description of the circuit. Next, we modify the synthesised netlist by replacing each sequential element with a TMR structure, including the triplet and the majority voter. Since legalisation process requires an initial placement of the circuit, next step is to perform standard-cell placement, using an industrial EDA tool, which performs all three stages of placement, *i.e.* global placement, legalisation and detailed placement, producing a legal solution. However, in this solution, the specified constraints are ignored. Thus, we continue by extracting the generated placement, using the Design Exchange Format (DEF), and passing to our EDA tool. DEF is an open specification for representing physical layout of an IC in an ASCII format. Following, we create the TMR groups, containing the triplets of each TMR structure, and specify the spacing constraints for each one. Then, we perform RADHARD legalisation, which satisfies the spacing constraints among the TMR groups. Finally, we extract the updated legal DEF and perform PPA analysis.

5.2 Min-bounded RADHARD Legalisation

Tables 5.2 and 5.3 show the results of our displacement-driven Min-bounded RADHARD Legalisation. Specifically, table 5.2 presents the area, power and hpwl ratio compared to the non-RADHARD legalisation, while the table 5.3 presents the worst negative slack (WNS) and total negative slack (TNS) overhead over the non-RADHARD.

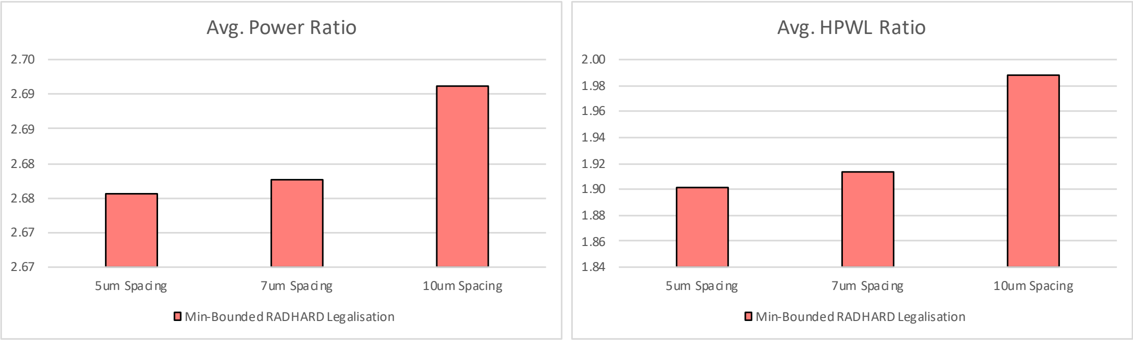
Benchmarks	Area Ratio	5 µm Spacing		7 µm Spacing		10 µm Spacing	
		Power Ratio	HPWL Ratio	Power Ratio	HPWL Ratio	Power Ratio	HPWL Ratio
lpffir	2.22	1.77	2.77	1.77	2.78	1.78	3.04
pid	1.94	2.18	1.97	2.18	1.99	2.19	2.08
openMSP	2.00	2.69	1.81	2.70	1.83	2.73	1.89
aes	1.89	1.43	1.84	1.43	1.85	1.44	1.90
aes192	1.85	1.29	1.55	1.29	1.55	1.30	1.60
aes_ip	1.96	2.45	1.90	2.45	1.91	2.46	1.98
ldpc	1.32	1.12	1.32	1.12	1.32	1.13	1.33
netcard	2.54	2.99	2.14	2.99	2.16	3.00	2.23
leon3mp	2.55	2.68	2.13	2.68	2.15	2.69	2.22
jpegencode	1.68	8.26	1.52	8.27	1.53	8.30	1.57
leon2	2.55	2.56	1.97	2.57	1.98	2.58	2.03

Table 5.2: Area, Power and HPWL ratio of our Min-bounded RADHARD Legalisation compared to non-RADHARD solution for the 11 tested benchmarks

Benchmarks	5 µm Spacing		7 µm Spacing		10 µm Spacing	
	WNS Overhead	TNS Overhead	WNS Overhead	TNS Overhead	WNS Overhead	TNS Overhead
lpffir	-0.34	-33.75	-0.34	-33.98	-0.35	-35.52
pid	-6.87	-7650.64	-6.75	-7511.34	-7.21	-7425.84
openMSP	-0.53	-35.19	-0.54	-35.30	-0.58	-36.09
aes	-0.79	-1526.84	-0.83	-1542.74	-0.82	-1568.64
aes192	-0.60	-2138.37	-0.60	-2139.97	-0.67	-2179.37
aes_ip	0.44	-1985.60	0.43	-1954.50	0.38	-2119.50
ldpc	-1.66	-36852.50	-1.73	-36669.30	-1.86	-37131.50
netcard	-5.66	-1115000.00	-5.52	-1125000.00	-5.52	-1135000.00
leon3mp	-9.26	-2548000.00	-9.41	-2558000.00	-9.51	-2578000.00
jpegencode	-4.84	-603000.00	-4.56	-2926000.00	-4.64	-2946000.00
leon2	-21.27	-13900000.00	-20.92	-13900000.00	-21.57	-14000000.00

Table 5.3: Worst Negative Slack (*WNS*) and Total Negative Slack (*TNS*) overheads in terms of non-RADHARD solution

Charts 5.3 and 5.4 show the average impact of our RADHARD legalisation. As shown in chart 5.3, increasing the spacing constraint, *i.e.* the minimum distance among the members of each TMR triplet, the power and HPWL worsen. This is expected, since increasing the spacing constraint, we increase the total wire length, and, thus, the power, which depends on the wire length. However, in terms of WNS and TNS, there is no monotonic behaviour depending on the value of spacing constraint, chart 5.4.

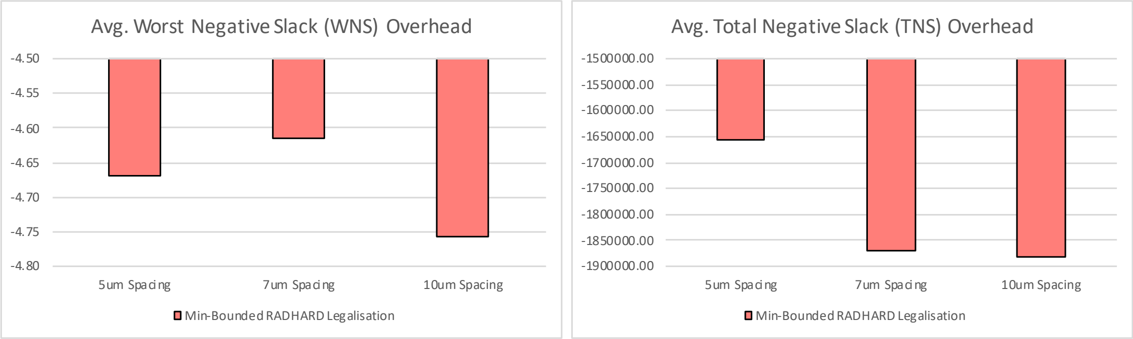


(a) Average Power Ratio for our Min-bounded RADHARD Legalisation

(b) Average HPWL Ratio for our Min-bounded RADHARD Legalisation

Figure 5.3: Impact of different spacing constraints in Power and HPWL: As the spacing constraint increases, both power and HPWL increase

In Appendix B is provided a list of layout for some of the tested benchmarks, showing the layout after performing non-RADHARD legalisation and after performing our Min-bounded RADHARD legalisation applying 5um, 7um and 10um spacing constraint.



(a) Average WNS Overhead of our Min-bounded RADHARD Legalisation compared to non-RADHARD Legalisation

(b) Average TNS Overhead of our Min-bounded RADHARD Legalisation compared to non-RADHARD Legalisation

Figure 5.4: Impact of different spacing constraints in WNS and TNS: No monotonic behaviour

5.3 Industrial RADHARD Legalisation

As mentioned previously, one industrial EDA tool supports the satisfaction of spacing constraints. Thus, we performed the same experiments using explicitly this industrial tool, in order to compare its results with the results produced by our legaliser. Tables 5.4 and 5.5 show the results of the industrial tool placement process satisfying at the same time the spacing constraints. Specifically, table 5.4 presents the area, power and HPWL ratio compared to the non-RADHARD legalisation, while the table 5.5 presents the worst negative slack (WNS) and total negative slack (TNS) overhead over the non-RADHARD.

Benchmarks	**Area Ratio**	**5 µm Spacing**		**7 µm Spacing**		**10 µm Spacing**	
		Power Ratio	**HPWL Ratio**	**Power Ratio**	**HPWL Ratio**	**Power Ratio**	**HPWL Ratio**
lpffir	2.22	1.76	2.60	1.76	2.60	1.77	2.83
pid	1.94	2.16	1.86	2.16	1.87	2.18	1.96
openMSP	2.00	2.67	1.75	2.67	1.76	2.71	1.83
aes	1.89	1.42	1.75	1.42	1.76	1.42	1.81
aes192	1.85	1.27	1.45	1.27	1.46	1.27	1.50
aes_ip	1.96	2.42	1.80	2.42	1.81	2.44	1.88
ldpc	1.32	1.11	1.30	1.11	1.30	1.12	1.31
netcard	2.54	2.97	2.06	2.97	2.06	2.98	2.15
leon3mp	-	-	-	-	-	-	-
jpegencode	1.68	8.18	1.45	8.18	1.46	8.21	1.51
leon2	-	-	-	-	-	-	-

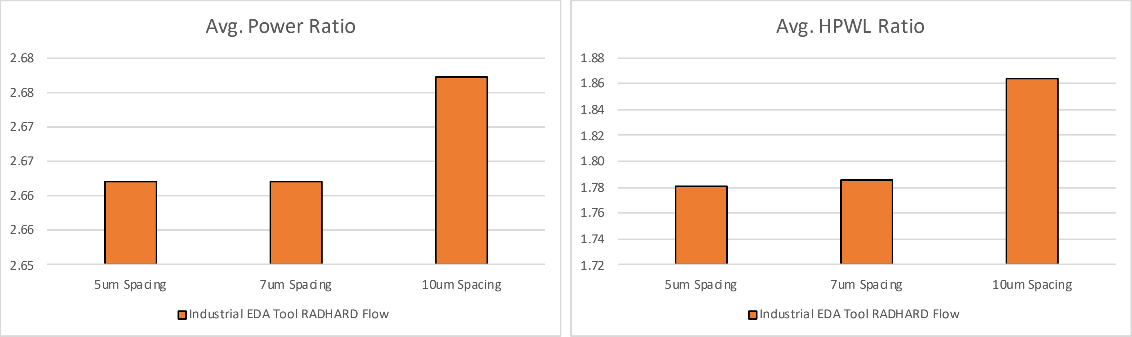
Table 5.4: Area, Power and HPWL ratio of Industrial RADHARD Legalisation compared to non-RADHARD solution for the 11 tested benchmarks

It is important to note that for two of our largest benchmarks, we could not get results and their cells in the tables are annotated by -, highlighted in red. This caused because the tool terminated unsuccessfully due to the lack of memory. As mentioned before, we tested our experiments in a server with 16GB SWAP Memory. This is a significant drawback of the industrial tool, and it is crucial to consider it for the rest of this chapter.

Benchmarks	**5 µm Spacing**		**7 µm Spacing**		**10 µm Spacing**	
	WNS Overhead	**TNS Overhead**	**WNS Overhead**	**TNS Overhead**	**WNS Overhead**	**TNS Overhead**
lpffir	-0.33	-33.16	-0.33	-32.90	-0.34	-34.31
pid	-8.31	-8996.24	-7.34	-7772.66	-7.07	-8463.84
openMSP	-0.54	-38.33	-0.55	-43.96	-0.60	-39.44
aes	-0.82	-1816.54	-0.80	-2143.73	-0.79	-1855.24
aes192	-0.49	-2022.87	-0.48	-2997.95	-0.48	-2104.17
aes_ip	0.39	-1832.80	0.38	-3346.79	0.38	-1851.10
ldpc	-0.68	-32589.90	-0.59	-54016.25	-0.71	-32687.00
netcard	-5.17	-1135000.00	-5.21	-1379986.48	-5.40	-1145000.00
leon3mp	-	-	-	-	-	-
jpegencode	-9.56	-737000.00	-9.40	-869978.55	-9.67	-737000.00
leon2	-	-	-	-	-	-

Table 5.5: Worst Negative Slack (*WNS*) and Total Negative Slack (*TNS*) overheads in terms of non-RADHARD solution

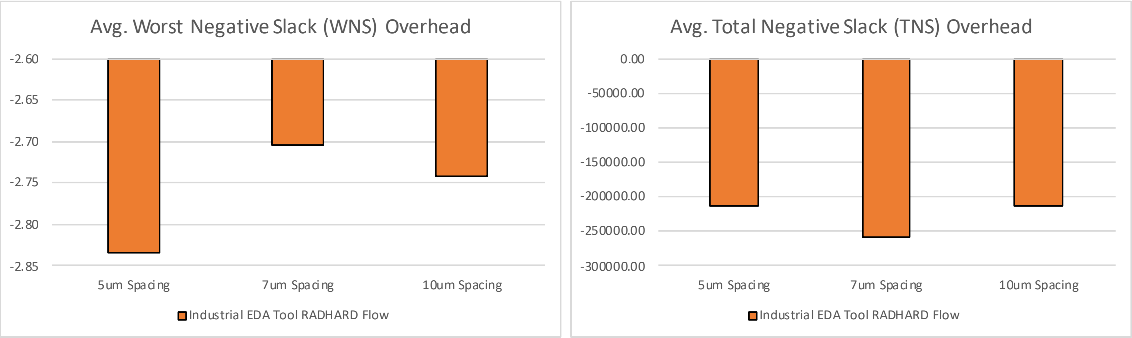
Charts 5.5 and 5.6 show the average power, HPWL ratio and WNS, TNS overhead compared to the non-RADHARD legalisation. As for power and HPWL, the Industrial RADHARD legalisation has the same behaviour as our Min-bounded RADHARD Legalisation, while for WNS and TNS we can not extract any outcome. Note that, in these charts are excluded the benchmarks for which the tool terminated unsuccessfully.



(a) Average Power Ratio for Industrial RADHARD Legalisation

(b) Average HPWL Ratio for Industrial RADHARD Legalisation

Figure 5.5: Impact of Industrial RADHARD Legalisation in Power and HPWL: As the spacing constraint increases, both power and HPWL increase



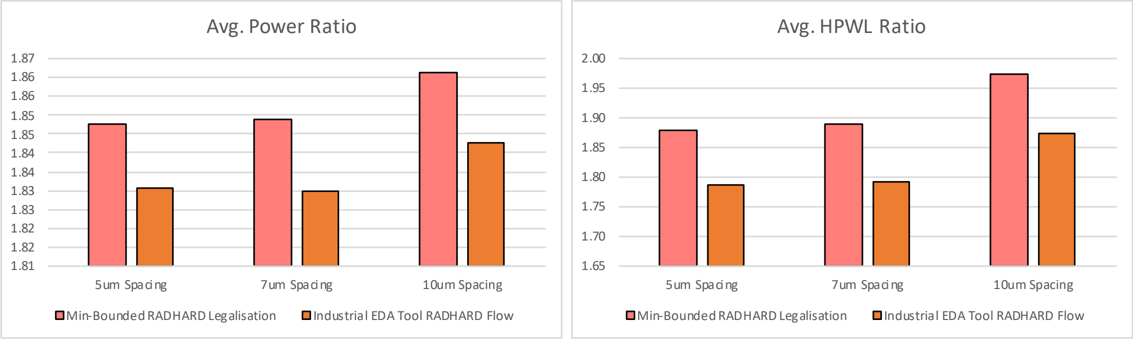
(a) Average WNS Overhead of Industrial RADHARD Legalisation compared to non-RADHARD Legalisation

(b) Average TNS Overhead of Industrial RADHARD Legalisation compared to non-RADHARD Legalisation

Figure 5.6: Impact of Industrial RADHARD Legalisation in WNS and TNS: No monotonic behaviour

5.4 Min-bounded *VS* Industrial RADHARD Legalisation

In order to evaluate the quality of our RADHARD Legaliser, we compare our results with those extracted by the industrial tool supporting spacing constraints. For this comparison, we used only the first seven benchmarks, *i.e.* `lpffir`, `pid`, `openMSP`, `aes`, `aes192`, `aes_ip` and `ldpc`, for which industrial tool has been terminated successfully. As for power, we can notice that industrial tool leads to slightly better results, by ~1%, as shown in chart 5.7a. So we can consider this difference negligible. Comparing our algorithm with the industrial tool, we can see that the latter results in better HPWL, by ~5%, chart 5.7b.

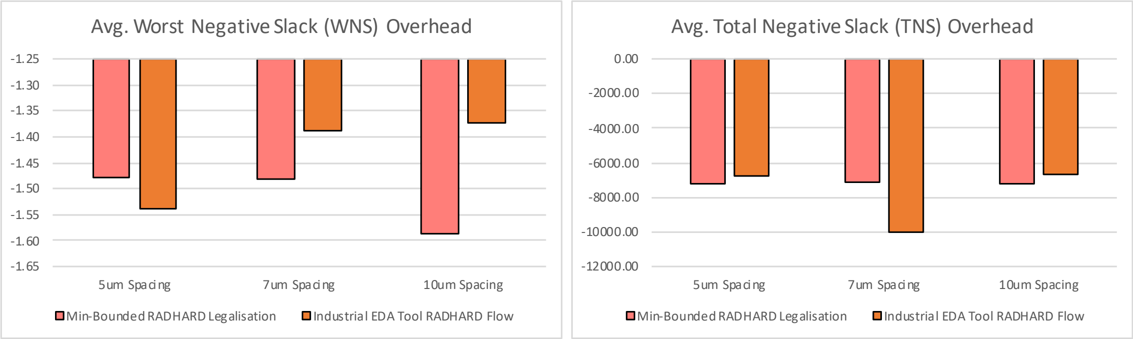


(a) Average Power Ratio

(b) Average HPWL Ratio

Figure 5.7: Min-bounded VS Industrial RADHARD Legalisation: Average Power and HPWL ratio

Considering WNS and TNS, we can not decide which one leads to better solution, since for some spacing constraints our approach is better while for others the industrial results in better solution, as we can see from chart 5.8.



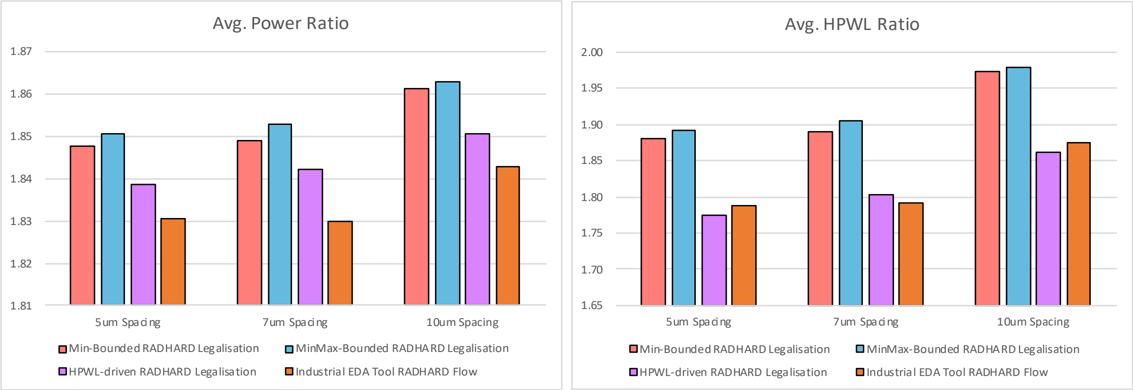
(a) Average WNS Overhead

(b) Average TNS Overhead

Figure 5.8: Min-bounded VS Industrial RADHARD Legalisation: Average WNS and TNS overhead

5.5 Impact of Optimisations

Concluding the previous results, our Min-bounded RADHARD Legalisation algorithm on average results in worse PPA results, compared to the industrial tool supporting the spacing constraints. This outcome urged us to apply a series of optimisations to our RADHARD Legalisation algorithm. First, we introduced except the minimum spacing constraint, a maximum one, to force the Legaliser to place the members of each TMR triplet closer together, *i.e.* MinMax-bounded RADHARD Legalisation. Another optimisation was the modification of Min-bounded RADHARD Legaliser to take into consideration not only the displacement of the cells but also the total HPWL by a weight factor, *i.e.* HPWL-driven RADHARD Legalisation. As for the latter, for the experiments, we used as weight factor 0, *i.e.* run Legaliser in full HPWL mode and ignore the cell displacement. We also tried to modify our RADHARD Legalisation algorithm to be timing-driven, *i.e.* Timing-driven RADHARD Legalisation, taking into consideration the Total Slack (TS), *i.e.* the sum of positive and negative slacks, of the design. However, it is needed to explore more cost functions before concluding into the best one. Thus, for this thesis, we will not present results for the Timing-driven RADHARD Legalisation approach.

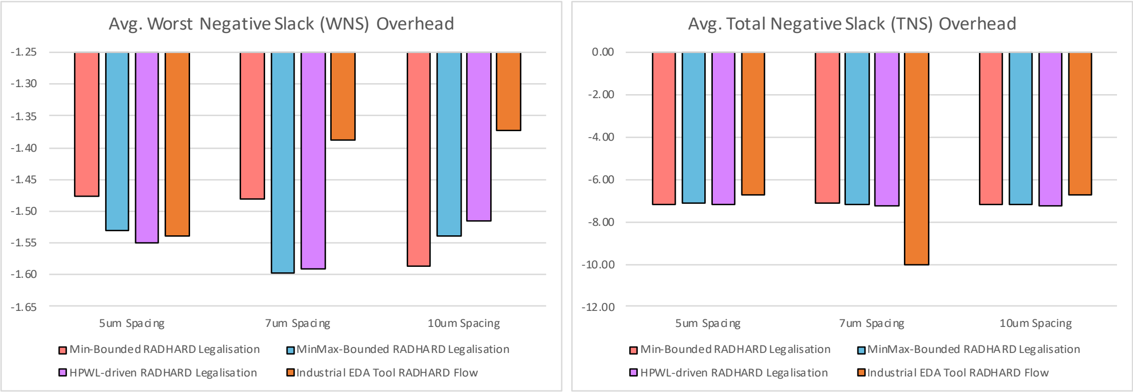


(a) Average Power Ratio

(b) Average HPWL Ratio

Figure 5.9: Impact of Optimisations on average Power and HPWL

The charts 5.9 and 5.10 presents the results of the various optimisations, *i.e.* MinMax-bounded RADHARD Legalisation and HPWL-driven RADHARD Legalisation, compared to the Min-bounded RADHARD Legalisation and Industrial RADHARD legalisation. As before, in terms of WNS and TNS, we cannot extract any outcome, since they do not have a monotonic behaviour. However, in the cases that our Min-bounded RADHARD Legaliser was worse than the industrial tool, after the optimisations still, our approach leads to worse results. Thus, the investigation of a Timing-driven RADHARD Legalisation may upset this behaviour and produce better results compared to the industrial tool. As for power and HPWL, our HPWL-driven Legalisation results in a better solution compared to Min-bounded and MinMax-bounded RADHARD Legalisation. In contrast to the industrial tool, our HPWL-driven Legaliser leads to worse power by ~0.5%, in which case we can suppose that their results are equivalent. Similar to power, the HPWL results for the HPWL-driven RADHARD Legaliser, are on average by ~1% better (5um spacing) or by ~0.5% worse compared to the Industrial tool. Thus, we can suppose again that the results for both approaches are equivalent.



(a) Average WNS Overhead

(b) Average TNS Overhead

Figure 5.10: Impact of Optimisations on average WNS and TNS overhead

Chapter 6

Conclusion and Future Work

In this thesis, we propose a radiation-hardening legalisation algorithm, based on existed legaliser called Abax. Our approach utilises the Triple-Modular Redundancy (TMR) methodology in the fields of circuit-level radiation-hardening techniques, which triplicates a critical cell of the circuit and adds a voter which receives the outputs of the three identical instances of the triplet and outputs their majority. However, this method is not efficient since a particle strike can affect more than one members of a TMR triplet leading to circuit malfunction. Our radiation-hardening legalisation algorithm solves this problem. In our work, we modified Abax to be able to satisfy a minimum spacing constraint among the members of each TMR triplet, reducing the probability a particle strike upset more than one TMR triplet members. Also, our radiation-hardening legaliser integrated into an under development EDA tool [23], which performs all the stages of the circuit design flow. Thus, our legaliser is fully compatible with the existed industrial EDA tools.

Our radiation-hardening legalisation must have as little impact as possible, compared the non-RADHARD legalisation process. Thus, we investigated a series of optimisations. First, we extended Abax to support not only a minimum spacing constraint for each TMR triplet but also a maximum one, not allowing members of the same triplet to be placed far away. Since this approach resulted in worse PPA results compared to Min-bounded legalisation, we performed another optimisation by modifying our legaliser to be HPWL-driven, taking into account the total wire length of the circuit. This optimisation leads to better PPA results compared to our original approach. Finally, as last optimisation, we tried to modify our legaliser to be timing-driven by using as cost function the total slack, *i.e.* the sum of both negative and positive slacks of the circuit.

Since the radiation-hardening design flow differs from the standard one, there is not sufficient support by industry, leading to the use of specialised radiation-hardened standard-cell libraries. However, one industrial EDA tool supports spacing constraints during the standard-cell placement. Depending on the design, our HPWL-driven RADHARD legalisation algorithm results in better PPA results compared to the industrial RADHARD flow or worse, but comparable. However, the industrial tool for large benchmarks terminated unsuccessfully due to lack of memory. Thus, our radiation-hardening legalisation approach seems very appealing.

As for future work, our first goal is to continue the implementation of our timing-driven legalisation. In this thesis, we experimented our algorithm using a 130nm standard-cell library. Thus, it is essential to investigate the scalability of our algorithm using smaller standard-cell libraries, e.g. 28nm. However, since the minimum spacing constraint will not scale, as it mostly depends on the particle strike characteristics, applying the same spacing, e.g. 10um, in smaller technologies, it will affect the circuit performance drastically. Thus, a possible solution would be to use N-Modular Redundancy (NMR), creating N instances of a cell, instead of using TMR and applying the spacing constraint. This way, the NMR can mask at most N-2 upsets happening at the same time in its members.

Appendix A

Acronyms

EDA Electronic Design Automation

IC Integrated Circuit

VLSI Very Large Scale Integration

GP Global Placement

TID Total Ionising Dose

DD Displacement Damage

SEE Single-Event Effect

SET Single-Event Transient

SEU Single-Event Upset

SBU Single-Bit Upset

MBU Multiple-Bit Upset

MCU Multiple-Cell Upset

SEFI Single-Event Functional Interrupt

SEL Single-Event Latchup

SEB Single-Event induced Burnout

SER Soft Error Rate

FIT Failures In Time

TF Transient Fault

FF Flip-Flop

TMR Triple-Modular Redundancy

SOI Silicon-On-Insulator

SOS Silicon-On-Sapphire

ELT Enclosed Layout Transistors

DICE Dual-Interlocked CEll

HIT Heavy Ion Tolerant storage cell

LET Linear Energy Transfer

PPA Power Performance Area

SRA Sub-Row Assignment

SRR Sub-Row Re-assignment

MRHC Multi-Row Height Cell

SRHC Single-Row Height Cell

TWL Total Wirelength

QoR Quality-of-Results

STA Static Timing Analysis

AAT Actual Arrival Time

RAT Required Arrival Time

DAG Directed Acyclic Graph

DEF Design Exchange Format

NMR N-Modular Redundancy

Appendix B

Designs Layouts

B.1 lpffir

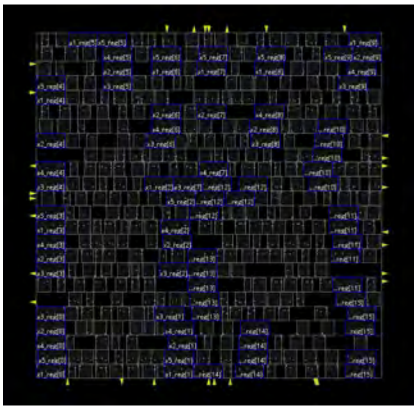
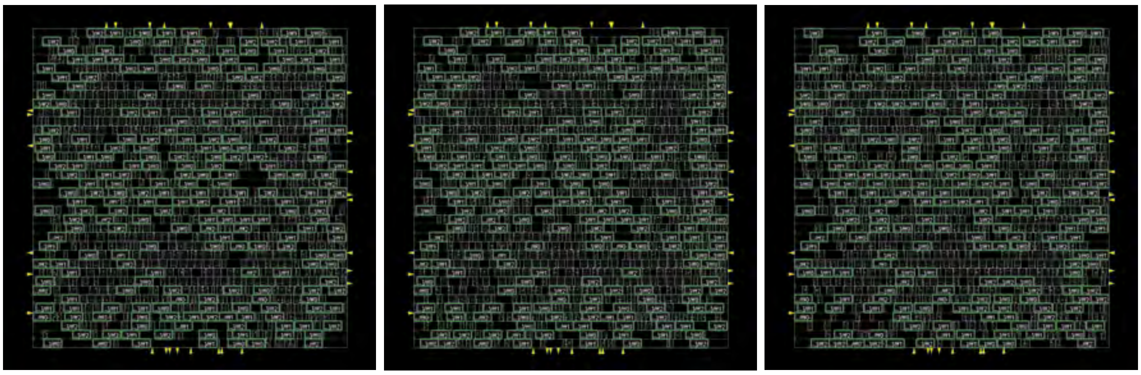


Figure B.1: `lpffir`: Non-RADHARD Legalisation Layout



(a) 5um Spacing

(b) 7um Spacing

(c) 10um Spacing

Figure B.2: `lpffir`: Min-bounded RADHARD Legalisation

B.2 pid

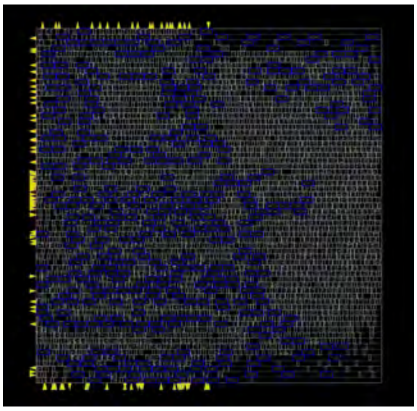
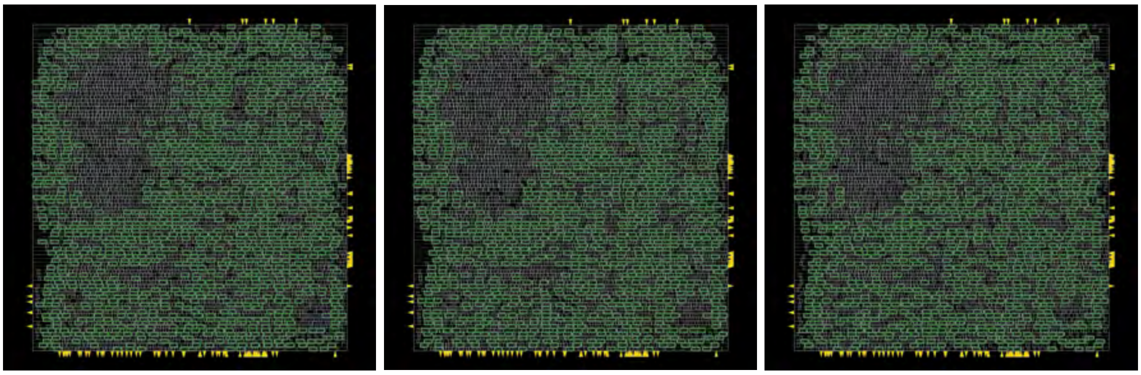


Figure B.3: `pid`: Non-RADHARD Legalisation Layout



(a) 5um Spacing

(b) 7um Spacing

(c) 10um Spacing

Figure B.4: `pid`: Min-bounded RADHARD Legalisation

B.3 aes

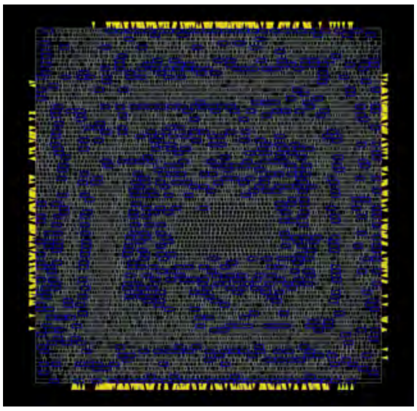
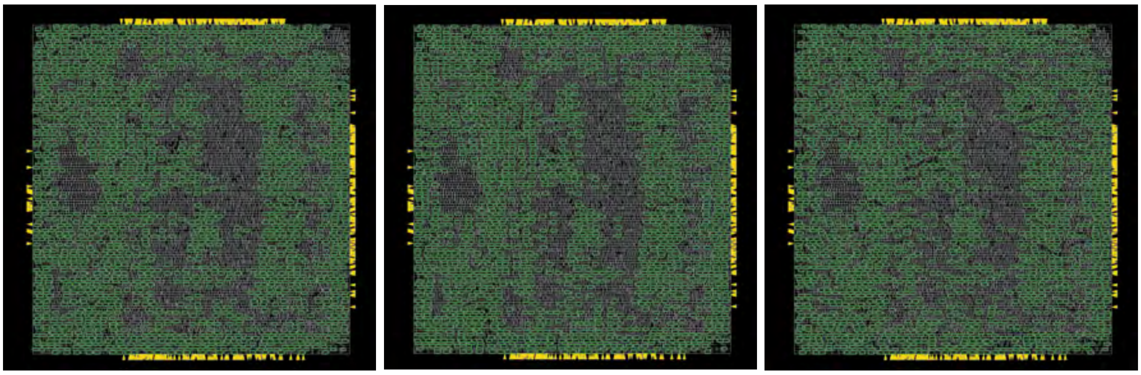


Figure B.5: `aes`: Non-RADHARD Legalisation Layout



(a) 5um Spacing

(b) 7um Spacing

(c) 10um Spacing

Figure B.6: `aes`: Min-bounded RADHARD Legalisation

B.4 aes192

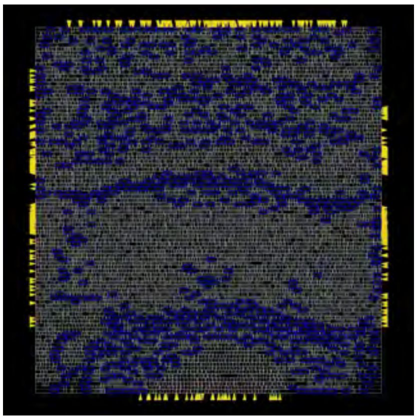
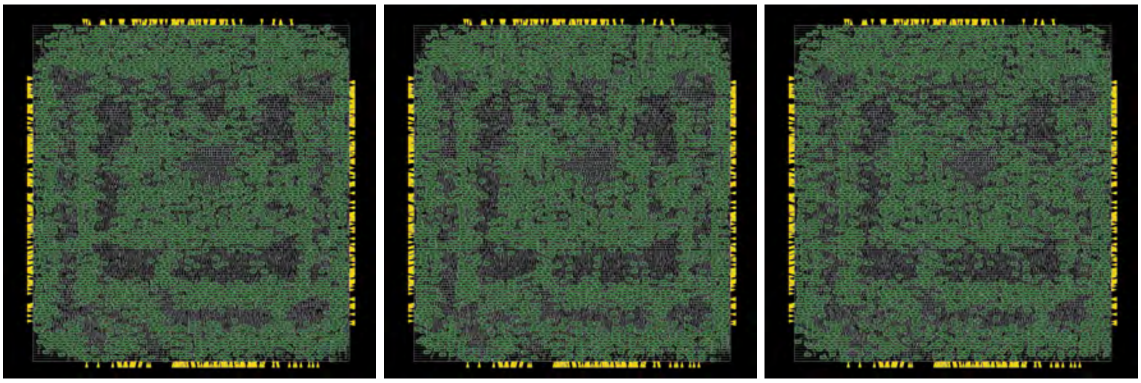


Figure B.7: `aes192`: Non-RADHARD Legalisation Layout



(a) 5um Spacing

(b) 7um Spacing

(c) 10um Spacing

Figure B.8: `aes192`: Min-bounded RADHARD Legalisation

B.5 aes_ip

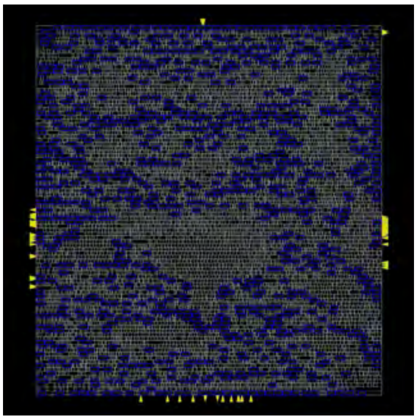
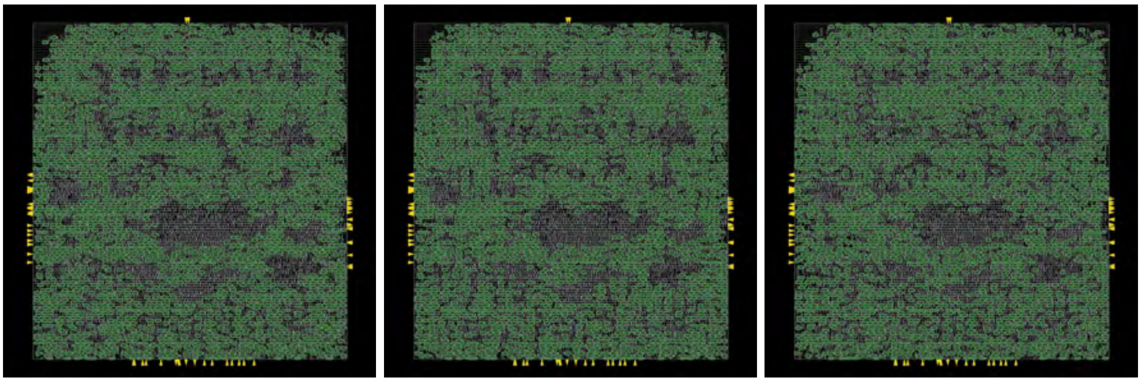


Figure B.9: aes_ip: Non-RADHARD Legalisation Layout



(a) 5um Spacing

(b) 7um Spacing

(c) 10um Spacing

Figure B.10: aes_ip: Min-bounded RADHARD Legalisation

B.6 ldpc

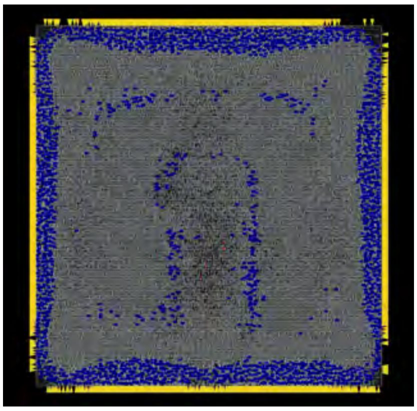
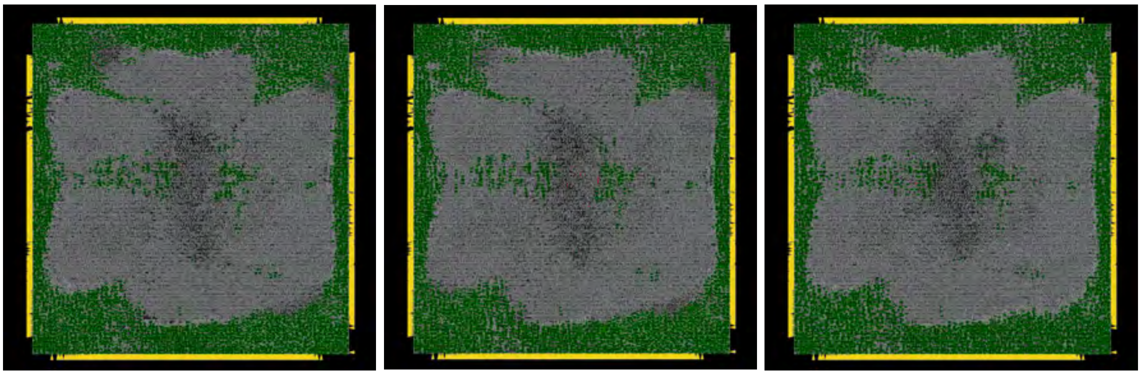


Figure B.11: ldpc: Non-RADHARD Legalisation Layout



(a) 5um Spacing (b) 7um Spacing (c) 10um Spacing

Figure B.12: `ldpc`: Min-bounded RADHARD Legalisation

B.7 jpegencode

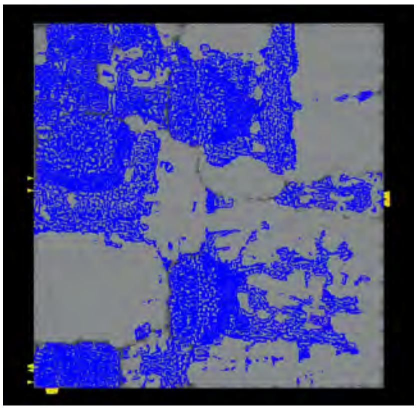
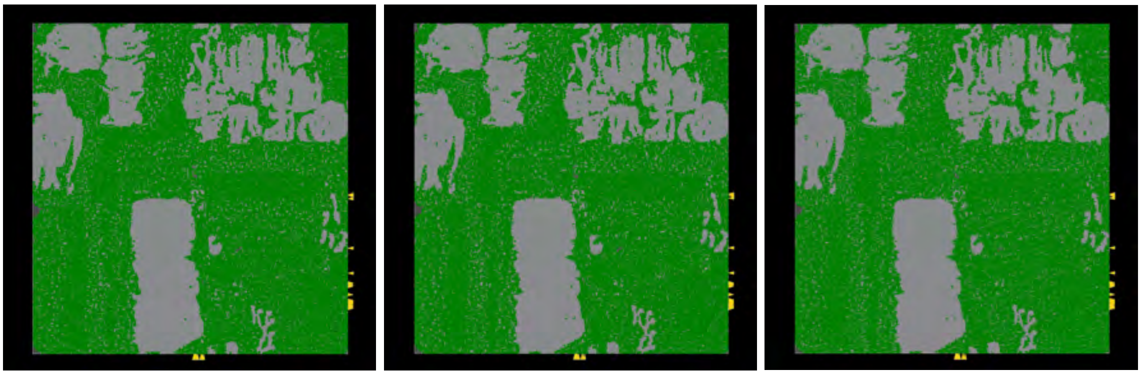


Figure B.13: `jpegencode`: Non-RADHARD Legalisation Layout



(a) 5um Spacing (b) 7um Spacing (c) 10um Spacing

Figure B.14: `jpegencode`: Min-bounded RADHARD Legalisation

Bibliography

[1] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media, 2011.

[2] M. Sarrafzadeh, M. Wang, and X. Yang, *Modern placement techniques*. Springer Science & Business Media, 2003.

[3] "NASA Facts: Understanding Space Radiation," https://spaceflight.nasa.gov/spacenews/factsheets/pdfs/radiation.pdf.

[4] "Cosmic rays: particles from outer space," https://home.cern/science/physics/cosmic-rays-particles-outer-space.

[5] "Solar Wind," https://en.wikipedia.org/wiki/Solar_wind#Properties_and_structure.

[6] "Studying the Van Allen Belts 60 Years After America's First Spacecraft," https://www.nasa.gov/feature/goddard/2018/studying-the-van-allen-belts-60-years-after-america-s-first-spacecraft.

[7] "Solar Flare," https://en.wikipedia.org/wiki/Solar_flare.

[8] S. Sayil, *Soft error mechanisms, modeling and mitigation*. Springer, 2016.

[9] W. J. Snoeys, T. P. Gutierrez, and G. Anelli, "A new nmos layout structure for radiation tolerance," in *2001 IEEE Nuclear Science Symposium Conference Record (Cat. No. 01CH37310)*, vol. 2. IEEE, 2001, pp. 822–826.

[10] R. Troutman, "Epitaxial layer enhancement of n-well guard rings for cmos circuits," *IEEE electron device letters*, vol. 4, no. 12, pp. 438–440, 1983.

[11] Q. Zhou and K. Mohanram, "Transistor sizing for radiation hardening," in *2004 IEEE International Reliability Physics Symposium. Proceedings*. IEEE, 2004, pp. 310–315.

[12] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron cmos technology," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 2874–2878, 1996.

[13] L. H.-H. Kelin, L. Klas, B. Mounaim, R. Prasanthi, I. R. Linscott, U. S. Inan, and M. Subhasish, "Leap: Layout design through error-aware transistor positioning for soft-error resilient sequential cell design," in *2010 IEEE International Reliability Physics Symposium*. IEEE, 2010, pp. 203–212.

[14] D. Bessot and R. Velazco, "Design of seu-hardened cmos memory cells: the hit cell," in *RADECS 93. Second European Conference on Radiation and its Effects on Components and Systems (Cat. No. 93TH0616-3).* IEEE, 1993, pp. 563–570.

[15] K. Bhattacharya and N. Ranganathan, "A new placement algorithm for reduction of soft errors in macrocell based design of nanometer circuits," in *2009 IEEE Computer Society Annual Symposium on VLSI.* IEEE, 2009, pp. 91–96.

[16] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata studies*, vol. 34, pp. 43–98, 1956.

[17] V. Petrovic and M. Krstic, "Design flow for radhard tmr flip-flops," in *2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems.* IEEE, 2015, pp. 203–208.

[18] I. L. Markov, J. Hu, and M.-C. Kim, "Progress and challenges in vlsi placement research," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 1985–2003, 2015.

[19] D. Hill, "Method and system for high speed detailed placement of cells within an integrated circuit design," Apr. 9 2002, uS Patent 6,370,673.

[20] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Abacus: fast legalization of standard cell circuits with minimal movement," in *Proceedings of the 2008 international symposium on Physical design.* ACM, 2008, pp. 47–53.

[21] N. Sketopoulos, C. Sotiriou, and S. Simoglou, "Abax: 2D/3D legaliser supporting look-ahead legalisation and blockage strategies," in *2018 Design, Automation and Test in Europe Conference and Exhibition (DATE).* IEEE, 2018, pp. 1469–1472.

[22] N. Sketopoulos, "Electronic design automation algorithms for standard cell legalization in microelectronic circuits," Master's thesis, Volos, Greece, 2016, http://ir.lib.uth.gr/handle/11615/45971.

[23] "ASP (Automated Structured Placement) Tool."