INTEGRATION, the VLSI journal 47 (2014) 115–122

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Planar CMOS to multi-gate layout conversion for maximal fin utilization

Shmuel Wimer [a,b,*]

[a] Technion—Israel Institute of Technology, EE Department, Haifa, Israel
[b] Bar-Ilan University, Engineering Faculty, Rama-Gan, Israel

ARTICLE INFO

Article history:
Received 1 October 2012
Received in revised form
27 February 2013
Accepted 6 March 2013
Available online 21 March 2013

Keywords:
Multi-gate transistors layout
TriGate
FinFET
Hard-IP reuse

ABSTRACT

Multi-gate transistors enable the pace of Moore's Law for another decade. In its 22 nm technology node Intel switched to multi-gate transistors called TriGate, whereas IBM, TSMC, Samsung and others will do so in their 20 nm and 14 nm nodes with multi-gate transistors called FinFET. Several recent publications studied the drawing of multi-gate transistors layout. Designing new VLSI cell libraries and blocks requires massive re-drawing of layout. Hard-IP reuse is an alternative method taking advantage of existing source layout by automatically mapping it into new target technology, which was used in Intel's Tick-Tock marketing strategy for several product generations. This paper presents a cell-level hard-IP reuse algorithm, converting planar transistors to multi-gate ones. We show an automatic, robust transformation of bulk diffusion polygons into fins, while addressing the key requirements of cell libraries, as maximizing performance and interface compatibility across a variety of driving strength. We present a layout conversion flow comprising time-efficient geometric manipulations and discrete optimization algorithms, while generating manually drawn layout quality. Those can easily be used in composing larger functional blocks.

1. Introduction and motivation

Intel has lately delivered to market its Ivy Bridge microprocessor which includes new 22 nm multi-gate transistors, called *TriGate* [1]. Those are fundamentally different than the traditional planar bulk transistors used since the early days of MOS technology. TriGate enables the VLSI industry to continue the pace of Moore's Law for 14 nm, 10 nm and smaller feature-size technologies. TriGate transistor uses three gates wrapped around the silicon channel in a 3-D structure, delivering significant performance and energy efficiency improvement over planar transistors, while the transistors density remains intact. We subsequently use the terms transistor and device interchangeably. Fig. 1(a) shows the structure of a traditional planar transistor, while Fig. 1(b) shows the structure of a TriGate transistor comprising a single fin. Fig. 1(c) shows that TriGate transistors can have multiple fins connected together to increase total drive strength [1].

Other companies like IBM [2], TSMC [3], GlobalFoundries [4], Samsung [5] and STMicroelectronics [6] are also reporting on their progress in 3-D multi-gate transistors, known as *FinFET*. Those include a variety of technologies, some are similar to Tri-Gate, others comprise four terminal devices, and some are bulk while others are of Silicon on Insulator (SOI). All those technologies are expected to be used for products manufactured in 20 nm and 14 nm. We subsequently use the term multi-gate to mean both TriGate and FinFET.

While the physical structure of the various multi-gate transistor types are somewhat different of each other, they all share a common property, where the drain and sources of the underlying CMOS transistor are implemented by fins. From layout and mask design perspective both TriGate and FinFET look similar, where their fins must align to a uniform grid imposed on the entire chip to enable lithography. Fig. 2 is a photograph of an ensemble of TriGate transistors connected together.

The increasing interest in multi-gate transistors yielded lately few works related to their drawing. In [7] the drawing of a single transistor with multiple fins is described, while in [8,9] the layout drawing of complete cells is described. Drawing multi-gate devices must obey strict design rules enforcing the fins to align to a predefined grid common to the entire layout. In [10] the layout density obtained by using TriGate (3T) devices versus FinFET devices having four terminals (4T) was compared, concluding that 3T devices offer higher layout density. (3T devices are also known as connected-gate (CG) FinFET, while 4T are called isolated-gate (IG) FinFET.)

Former works dealt with the construction of multi-gate layouts from scratch, which is the common VLSI design scenario. There, cell library families comprising logic, sequential, memory, IO and other circuits are designed first and then used to build larger

* Corresponding author at: Technion—Israel Institute of Technology, EE Department, Haifa, Israel. Tel.: +972 4 8295744.
E-mail address: wimes@biu.ac.il

0167-9260/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.vlsi.2013.03.004

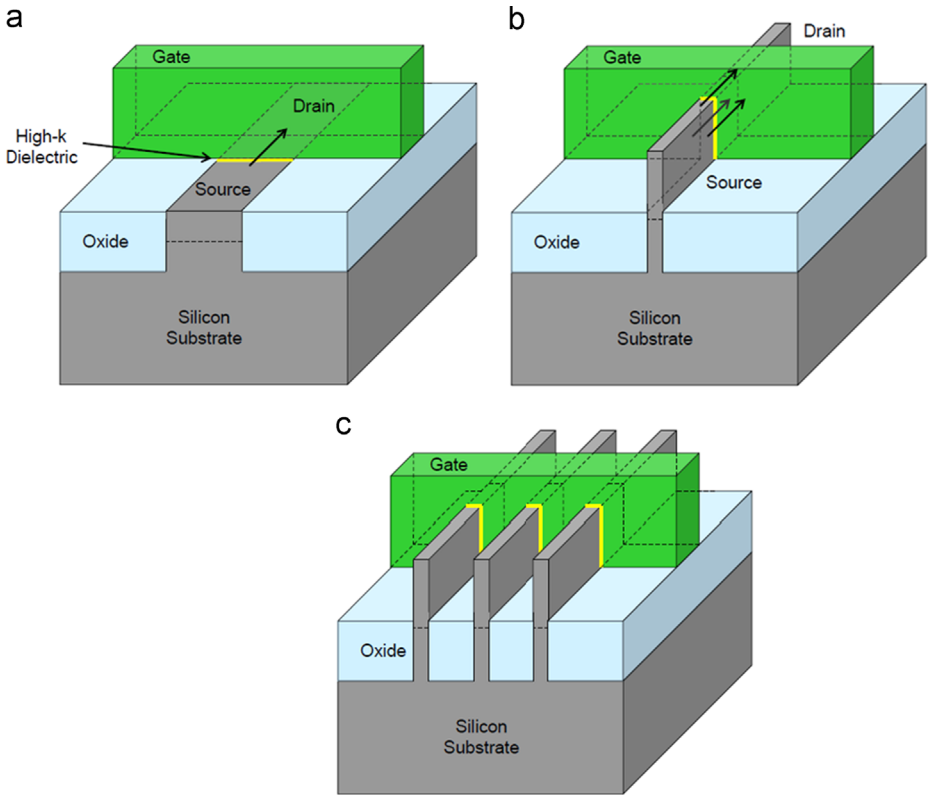

Fig. 1. Planar device in (a), single fin TriGate device in (b) and multiple fin device in (c) [1].

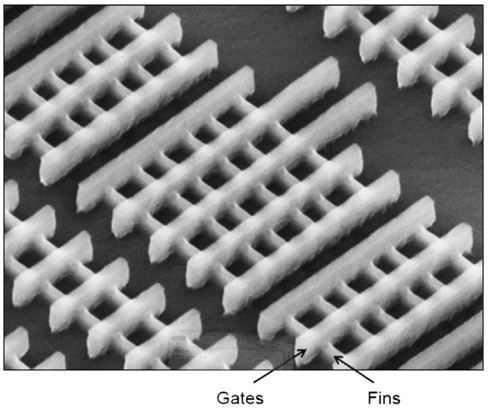

Fig. 2. TriGate transistors connected in a functional circuit [1].

functional blocks and systems. In a different scenario, known as *hard-IP reuse* [11,16], the physical layout of an existing chip, called source layout, is migrated into new target technology in an attempt to deliver the same functionality with improved energy efficiency and performance, but in lower cost. Such conversions were used at Intel as a part of its well-known Tick-Tock marketing strategy [11,12].

In a hard-IP reuse methodology called *cell-based*, the layout of the cells in the libraries are converted first and then are being placed and routed in an attempt to preserve their relative positions as in the source layout [13]. Converting the layout of cell libraries has the advantage of re-using the IP invested in their optimization, which mainly deals with the relative position of the transistors within the cells, their relative sizes and cell's interface. Such optimization is primarily affected by cell functionality and its underlying connectivity, while the specific technology in use has secondary impact on the optimization. Since cell's functionality and connectivity do not change across technology migration, layout topology is not suspected to significant changes. It is therefore beneficial to develop layout conversion algorithm supporting the transition from planar to multi-gate technology, and also from multi-gate to multi-gate for future technology migrations.

An early attempt for automatic conversion of planar CMOS SOI microprocessor into FinFET technology at 100 nm was reported in [14], but details were not provided. In [15] the authors describe an automatic process to replace the active diffusion regions of devices by appropriate fins to generate a legal and functional cell layout. Some oversizing of the active area takes place in order to legalize a fin in case it is not fully contained within the source active area. Though increasing the probability of legal and functional FinFET cell, it is a local transformation which does not take into account the entire cell devices simultaneously and their “competition” on the area. Furthermore, oversizing the active area as proposed in [15] still does not guarantee that all the required fins can indeed be materialized.

This work presents a planar to multi-gate layout conversion flow maximizing the fin utilization in the target cell. We shall address the main requirements from standard cell libraries while minimizing the amount of additional manual artwork required by a mask designer in case that conversion goals have not been met. The proposed flow is also applicable for migration of multi-gate source layout to multi-gate target layout across progression in process technologies, a situation expected for years to come. The rest of the paper is organized as follows. Next section presents the conversion problem and its tradeoffs. Section 3 outlines the conversion flow and briefly explains its various steps. Section 4 elaborates on the step of polygon rectangular decomposition, while Section 5 discusses the grid imposition algorithm and shows its optimality. Experimental results are discussed in Section 6.