TherMOS: A Thermal Model for Self-heating in Advanced MOSFETs

Vidya A. Chhabria, Arvind K. Sharma, Meghna G. Mankalale, and Sachin S. Sapatnekar
University of Minnesota

***Abstract*—Modern transistors suffer from excessive heat-confinement due to confined structures and limited paths to thermal ambient. This results in device self-heating which can degrade performance and reliability. This article describes TherMOS, a publicly available software that builds thermal models to estimate the temperature rise as a consequence of self-heating in 14nm FDSOI and 7nm FinFET technologies.**

I. Introduction

To enable efficient scaling, designs at the 14nm/16nm technologies are built on silicon-on-insulator (SOI) substrates and have three dimensional architectures. SOI improves performance by reducing leakage and parasitics and the three-dimensional nature provides superior electrostatic control. However, this comes at the cost of increased self-heating. In SOI devices, a 14nm FDSOI FET for example (Fig. 1(a)), the thick buried oxide layer with low thermal conductivity degrades the effectiveness of heat-conduction through the substrate, resulting in a significant portion of heat to flow along the channel into the source and drain terminals. Moreover, in devices with 3-D architectures such as the Bulk/SOI-FinFETs (Fig. 1(b)), a thin layer of gate oxide surrounds the fin on three sides further trapping the heat. The low fin and gate pitches in these technologies result in high heat flux with restrictive paths to the ambient. The consequent rise in temperature degrades device performance and reliability, making analyzing self-heating effects critical part of the design cycle.

TherMOS [1], an open-source software, is a Python-based framework that builds thermal models to measure the effect of self-heating on these device structures. TherMOS solves the 3-D Fourier heat equations in steady-state by using finite difference method and thermal-electrical equivalence. The power dissipation (or heat) in each element is modeled as a current source. If there are n finite elements, then solving for temperature in each element amounts to solving a system of n linear equations of the form:

$$G\mathbf{T} = \mathbf{P} \tag{1}$$

Here, $G \in R^{n \times n}$ is the thermal conductance matrix, $T, P \in R^n$, are vectors of the unknown temperatures, and the power dissipation within each element, respectively. The G matrix can be formed in O(n) time using standard nodal analysis, and the values in G depend on thermal conductivities between two neighboring elements and the boundary conditions. TherMOS can accurately simulate the temperature rise within a transistor or an array of transistors by detailed modeling of power distribution both within and across the transistors. It also accounts for submicron scattering effects, such as phonon boundary scattering by using scaled values of thermal conductivity within the channel/fin. The key contributions of TherMOS are the following:

- Simulation for different technologies (Bulk/SOI MOSFET/FinFET) at the change of a button
- Improved run-times due to the non-uniform meshing, with larger finite elements in non-critical regions
- Fast simulation of multigate and multifin arrays.
- One-stop-shop for device parameters and thermal conductivity values compiled from various literature sources for 7nm FinFET and 14nm FDSOI devices
- Detailed models for power distribution across multiple gates, multiple fins, and within the channel of devices.

The architectures of the devices TherMOS supports are shown in Fig. 1. The primary paths for heat conduction are towards the substrate and along the channel into the source drain terminals. As a temperature solver which models for these thermal paths, TherMOS takes the following inputs:

- a device parameter file which specifies the thermal conductivities of all the materials and device dimensions.
- a list of various arguments including device types (NMOS/PMOS), power dissipated by the device(s), width, and technology.
- a thermal model configuration file which specifies the resolution of the finite elements.

and generates the following outputs:

- a temperature report which states the summary (minimum, maximum, and mean) of the temperature rise within the device(s).
- a Python object which stores the temperature vector for every node in device(s).
- an image of the temperature profile image depicting the detailed distribution across simulated device(s).

In the following sections, we explain the steps to use TherMOS with details on the input and output specification. A detailed explanation on the algorithms employed in TherMOS can be found in [2].

II. TherMOS User Guide

TherMOS is publicly available on GitHub [1] and is capable of building thermal models for Bulk/SOI FinFETs and FDSOI devices. It is implemented in Python3.6 using Numpy, Scipy, and TensorFlow packages, and has the following prerequisites:

- Python 3.6
- Pip 18.1
- Python3-venv

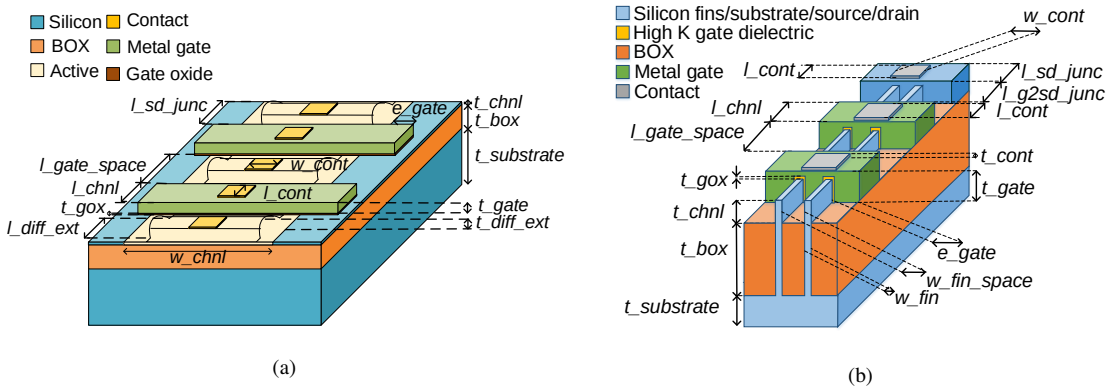


Figure 1: Device architecture of (a) 14nm FDSOI MOSFET and (b) 7nm Bulk FinFET.

TherMOS runs best on Red Hat Enterprise, Ubuntu or CentOS Linux distributions. Additionally, please refer to requirements.txt file in the repository for other dependencies. The packages in requirements.txt will be automatically installed in a virtual environment during build process.

A. Download and install

TherMOS can be downloaded and installed from the GitHub repository using the instructions in Listing 1.

```
1 git clone
      https://github.com/VidyaChhabria/TherMOS.git
2 cd TherMOS
3 source install.sh
4 pytest
```

Listing 1: Download and install TherMOS.

The install script (install.sh), shown on line 3 of Listing 1, creates a virtual python environment and installs all the necessary packages, mentioned in the requirements.txt file, using pip. Pytest (line 4) is used to runs a set of unit tests to check if the installation has been successful.

B. Tool Usage

TherMOS relies on a device parameter file that provides the various device dimensions and thermal conductivities of the various materials used. An example of this file for 7nm FinFET technology (model_parameters_FinFET.json) is shown in Listing 2. The repository contains two such files for both 7nm FinFET and 14nm FDSOI devices. The values in these files have been compiled from various literature sources. The references to these values are shown in Table I. These values can also be found on the GitHub repository as default parameter values to use TherMOS.

```
1  {
2      "dimensions": {
3          "l_chnl": 20,
4          "t_box": 25,
5          "t_substrate": 25,
6          "t_gate": 44,
7          "t_chnl": 32,
8          "t_gox": 1.6,
9          "t_cont": 10,
10         "t_diff_ext": 20,
11         "l_gate_space": 35,
12         "l_diff_ext": 25 ,
13         "l_cont": 18,
14         "l_sd_junc" : 20,
15         "l_g2sd_junc" : 20,
16         "w_fin" : 7,
17         "w_fin_space" : 20,
18         "w_cont": 18,
19         "e_gate": 10
20     },
21     "thermal_conductivity": {
22         "gate": 197e-9,
23         "Si NMOS SD": 148e-09,
24         "Si PMOS SD": 114e-9,
25         "Si substrate": 148e-9,
26         "SiGe PMOS channel": 12.75e-9,
27         "Si NMOS channel": 14.7e-9,
28         "Si PMOS fin": 12.75e-9,
29         "Si NMOS fin": 14.7e-9,
30         "SiO2": 0.8e-9,
31         "spacer": 30e-9,
32         "contact": 385e-9
33     }
34 }
```

Listing 2: Sample model parameter file for FinFET technology.

Table I: Default 7nm FinFET and 14nm FDSOI parameters in TherMOS

Parameter	14nm FDSOI	Source	7nm FinFET	Source
l_chnl	20	[3]	20	[6]
t_box	25	[3]	25	[7]
t_substrate	500	[3]	500	[8]
t_gate	44	[3]	44	[7]
t_chnl	32	[3]	32	[7]
t_gox	1.6	[3]	1.6	[7]
t_cont	10	[4]	80	[8]
t_diff_ext	20	[3]	32	[9]
l_gate_space	34	[3]	34	[6]
l_diff_ext	25	[3]	25	[6]
l_cont	18	[4]	18	[6]
l_sd_junc	20	[3]	20	[6]
l_g2sd_junc	30	[3]	30	[6]
w_fin	-	-	7	[6]
w_fin_space	-	-	20	[6]
w_cont	18	[4]	18	[6]
e_gate	10	[5]	10	[6]

After defining the parameters, TherMOS can be run using the command in Listing 3. TherMOS takes arguments listed in Table II.

```
python3 <process_type> -device_type <str> -n_gate
    <int> -power <float> [-n_fin <int> | -width
    <float>]  -active "<int_list>" -percent
    "<float_list>"
```

Listing 3: TherMOS run command.

Table II: List of arguments as input to TherMOS.

Argument	Comments
-h, –help	Prints out the usage
<process_type>	Process and technology specification (str, required). Choice: {Bulk_MOSFET, SOI_MOSFET, Bulk_FinFET, SOI_FinFET}
-device_type <int>	Type of the device (str, required). Choice: {NMOS, PMOS}
-n_gate <int>	Specifies the number of transistors in the simulation (int, required)
-power <float>	Specifies the total power dissipated by the array of transistors (s)
-n_fin <int>	Number of fins in FinFET (required, int, use only when process_type is FinFET)
-width <float>	Width of the MOSFET (required, float, use only when process is MOSFET)
-active “int_list”	States the list of gate ids that are dissipating power (list of int, optional)
-percent “float_list”	Percentage of power distributed between the active gates (optional, list_of_float)

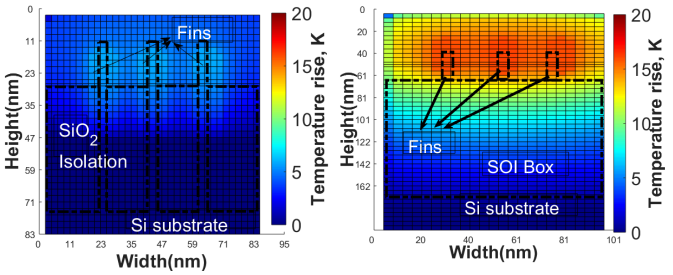
The python script generates (i) a temperature report, which specifies the average and maximum temperature rise, and (ii) a python object which stores the temperature for each finite element of the simulated devices. Optionally, the temperature distribution plot can be viewed using GNU Octave [10]. A script that loads the generated Python object into Octave and plots the 3-dimensional temperature contours is available in the repository.

C. Sample Results

Sample TherMOS-generated temperature contour plot for 7nm bulk FinFET and SOI FinFET are shown in Fig. 2(a) and Fig. 2(b) respectively. This can be viewed using GNU Octave.

III. Future Directions

In its current state, TherMOS is an open-source temperature analyzer for advanced FDSOI FETs and FinFET technologies. Near-term efforts are focused on developing models for 5nm nanowire FETs. We aim to eventually integrate TherMOS with reliability models for bias temperature instability, hot carrier injection and electromigration. While the tool is found to provide convincing results in its current state, continuous effort is being focused on addressing certain limitations, involving scalability issues and calibration of the analyzer with commercial tools.



(a) (b)

Figure 2: Sample temperature contour plots generated by TherMOS for (a) 7nm bulk FinFET and (b) 7nm SOI FinFET [2].

IV. License

The repository [1] is under a permissive BSD-3 Clause license.

V. Acknowledgement

This work was supported in part by the National Science Foundation (NSF) under Award CCF-1714805.

References

[1] TherMOS, https://github.com/VidyaChhabria/TherMOS
[2] V. A. Chhabria and S. S. Sapatnekar, “Impact of Self-heating on Performance and Reliability in FinFET and GAAFET Designs,” *Proc. ISQED*, pp. 235-240, 2019.
[3] Q. Liu *et al.*, “High performance UTBB FDSOI devices featuring 20nm gate length for 14nm node and beyond,” *Proc. IEDM*, pp. 9.2.1-9.2.4, 2013.
[4] O. Weber *et al.*, “14nm FDSOI technology for high speed and energy efficient applications,” *Symp. on VLSI Tech.*, pp. 1-2, 2014.
[5] ”NCSU 45nm FreePDK”, Available at https://www.eda.ncsu.edu/wiki/FreePDK45:PolyRules
[6] Lawrence T. Clark, *et al.*, “ASAP7: A 7-nm finFET predictive process design kit,” *Microelectronics Journal*, Volume 53, 2016.
[7] Vinay Vashishtha, *et al.*, “ASAP7 Predictive Design Kit Development and Cell Design Technology Co-optimization,” ICCAD Embedded Tutorial, 2017.
[8] M. Shrivastava, *et al.*, “Physical insight toward heat transport and an improved electrothermal modeling framework for FinFET architectures” *IEEE Trans. Electron Devices*, vol. 59, no. 5, pp. 1353-1363, 2012.
[9] B. Swahn and S. Hassoun, “Electro-thermal analysis of multi-fin devices,” *IEEE Trans. VLSI Systems*, vol. 16, no. 7, pp. 816-829, 2008.
[10] “GNU Octave” https://www.gnu.org/software/octave