Chapter 2

Device structures and device simulation techniques

Siegfried Selberherr, Claus Fischer, Stefan Halama, Christoph Pichler, Gerhard Rieger, Gerhard Schrom, Thomas Simlinger

2.1 Introduction

Process and device simulation is commonly used for the design of new VLSI technologies. Simulation programs serve as exploratory tools in order to gain better understanding of process and device physics. On the other hand, simulations are also carried out after the design phase to optimise certain parameters of a technology, e.g., to improve device performance and reliability or to increase the yield [7].

For all these tasks the term TCAD, short for *technology computer-aided design*, was coined. TCAD includes both physically rigorous as well as simplified process and device simulation in one to three spatial dimensions. Furthermore, links to layout-oriented CAD and circuit simulation are required.

Depending on the particular application of TCAD tools, different demands arise. On the one hand, the development of new technologies and the prediction of the behaviour of new devices require both accuracy and robustness of the tool. In this case, very sophisticated physical models and numerical techniques must be used, usually at high computational costs. An example of such a model is the two-dimensional simulation of the transient-enhanced diffusion during rapid thermal annealing [28].

On the other hand, for statistical simulations [6] or post-design process optimisation [74] speed is the most crucial issue, as physical models can be calibrated to an existing manufacturing process and hence do not pose a reliability problem.

Independently of the progress in advanced physical modelling, the fast and simple "tuned" models will still remain in broad use; there is no unique "best model" for all simulation problems.

TCAD involves a number of scientific disciplines in addition to electrical engineering and computer science. This has also had an impact on the characteristics of the software which has been produced by that heterogeneous community during

Table 2.1 *List of TCAD systems and tool integration approaches, in alphabetical order of institutions.*

Name	Institution	Status	Data level	Task level	Presentation Level	Ref.
MECCA	*AT&T*	IP	awk/sed, C++	UNIX Shell, Tcl	Tk	[38] [39]
PREDITOR pdFab	*CMU*	E C	CDB/HCDB	Tcl	Tk, Motif	[70] [71]
"Superviser"	*Hitachi*	IP	Converter	"Superviser"		[43]
VATS	*IBM*	IP	VATS/DB	WIZARD (Tcl)	WIZARD (Tk)	[36]
EASE	*Intel*	IP	PIF derivative	UNIX Shell, FASST/TEL	Motif	[13] [41] [42]
CAFE	*MIT*	I	BPIF/Gestalt	*MIT* PFR		[5]
P&D Workbench	*NEC*	IP		MEDLEY + ESCORT	DAIJOBDA	[68]
UNISAS	*Oki*	IP	GCOS	UNICOL		[47]
IDDE	*Philips UK*	IP	ASCII PIF derivative	none	Apollo DIALOGUE	[23] [24]
PRIDE	*Philips USA*	IP	ASCII PIF derivative	none	SunView	[63]
SATURN	*Siemens*	IP	SATURN	UNIX Shell	none	[31] [32]
MASTER	*Silvaco*	C	SSF	none	DeckBuild	[29]
STORM	*ESPRIT*	P	DAMSEL	none	STORM UI / IDAS	[40]
Alladin, SEWB	*Stanford University*	R	SWR			[12]
CAESAR	*TMA*	C	TIF	Module/Step	CAESAR	[3]
ITS	*Texas Instruments*	IP	none	none	Motif	[22]
"System controller"	*Toshiba*	IP	ASCII topography	interface programs	none	[34]
VISTA	*TU Vienna*	P	PAI/PIF	XLISP	Xvw + VUI	[14] [25] [26] [56]
PROSE	*UC Berkeley*	I	BPIF, SWR	Tcl	Tk + VEM	[76] [77]
SIMPL-IPX	*UC Berkeley*	I	Converter (BTU)	none	SIMPL-DIX	[58] [72]

Status	Meaning	
I	Internal:	The system is not used outside the institution listed.
P	Production:	The system is known to be in use at all.
C	Commercial:	The system is commercially available.
E	Experimental:	The system is an experimental implementation.
R	Industrial Prototype:	The system is a prototype implementation.

the past 20 years. From a historical perspective, the evolution of TCAD started with single applications (so-called *point tools*) which were implemented independently by different developers. A typical, traditional point tool solves a specific TCAD problem, it has usually been designed without consideration of a (software) environment in which it will be used.

During the past years these tools have been integrated with other services (like visualisation, interactive design editing facilities, a tool control level) to form *Technology CAD systems.* To support the users, most of these TCAD systems are cur-

rently equipped with converter-based tool coupling, homogenising user interfaces and TCAD-specific task level environments which are often centred around typical design tasks like optimisation. This is very effective in facilitating the current use of the system, but it does not address the real software-related difficulties retarding the progress of Technology CAD.

As stated, TCAD systems which provide process simulation, device simulation, common services and miscellaneous task control facilities are nowadays used worldwide. Some of them were developed by extending existing integrated Electronic CAD (ECAD) systems or Computer Integrated Manufacturing (CIM) systems. Thus it is very hard to distinguish between advanced ECAD, CIM, and TCAD systems; a review and classification of ECAD systems is given in [35]. Furthermore, there are many TCAD approaches which emphasise certain aspects (like optimisation) and even integrate some simulators, but entirely lack to fulfil any of the requirements and architectural criteria presented here. Hence, despite their practical utility, these systems have been discounted. Other remarkable achievements which unfortunately lack a persistent implementation have also not been considered. For these reasons and because of the sparse published data, the overview of existing systems given in Table 2.1 is incomplete and may not accurately represent the current status in all cases. Only a minority of systems is publicly available, hence a detailed classification of the approaches taken for data level, task level, and presentation level is not possible within the scope of this text. Where in doubt, fields in the table have been left blank. Note that SSF is a derivative of the SUPREM-4 file format, designed by *Silvaco*, and TIF is a derivative of the SUPREM-4 file format, by *TMA*.

The systems listed exhibit an impressive variety of architectures and interesting features. Despite their shortcomings from a methodology point of view, each of them (and of many more unlisted TCAD components) is a remarkable achievement in terms of practical utility. An in depth overview of different existing TCAD systems is given in [15].

2.2 User Interfaces in TCAD Systems

The state-of-the-art interfaces to computing devices, i.e. workstations, are capable of displaying many virtual screens, which are usually termed windows, on a large physical screen. Therefore, several workstation-based TCAD systems can be found which address the issue of multi-tool integration into a unified user interface, mostly based on the X Window System [33]. PRIDE [63], based on SunView which is a vendor specific variation of the X Window System, exhibits a user interface and task level architecture which is strongly influenced by the preprocessing – computation – post-processing task model of TCAD. SIMPL-IPX [58], based directly on Xlib [50], features a central interactive graphical editor which has menu-oriented facilities for running simulators. In both cases, implicit or explicit assumptions about the design cycle have an impact on the (top-down) design of the software and restrict the design tasks which can be performed or implemented. A more flexible and

extension-oriented user interface architecture has been accomplished in PROSE [77], which is mainly due to the use of the generic Tcl interpreter [52] and Tk toolkit [53].

2.2.1 The VISTA solution

The structure of the VISTA user interface [27] is shown in Figure 2.1. The bottom layer is the X Toolkit [1], an object-oriented subroutine library, designed to simplify the development of X Window System applications. The X Toolkit defines meth-

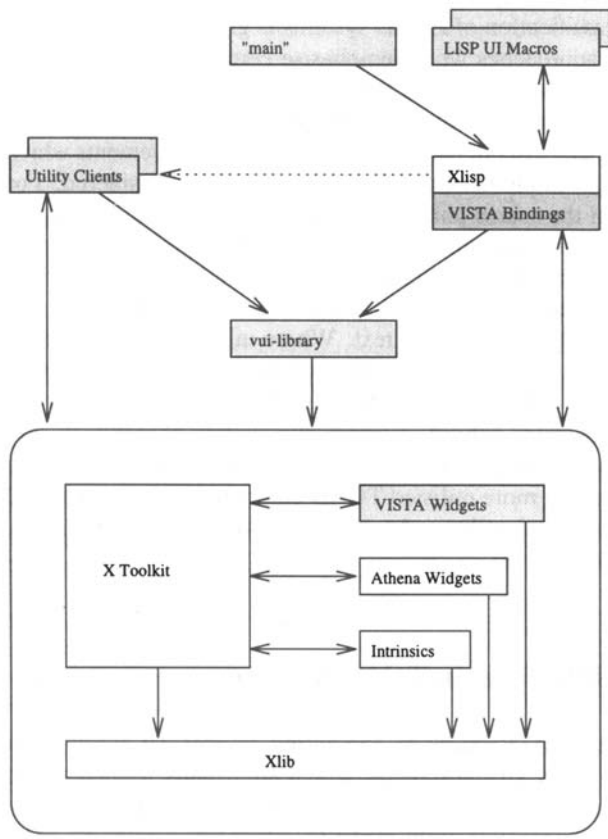

Figure 2.1 *The structure of the* VISTA *user interface. Shaded boxes represent extensions to the public domain products* XLISP *and the MIT* X Window System. *The arrows indicate the sequence of function calls between different parts of the user interface.*

ods, so called Intrinsics [51], for creating and using widgets, which appear to the user as pop-up windows, scroll bars, text-editing areas, labels, buttons, etc. Basic functionality is provided by the generic Athena widgets, which are part of the

public domain X Window System distribution from the Massachusetts Institute of Technology. This widget set was used rather than any other open standard, because a migration from these generic widgets to another widget set (like OSF/Motif [57]) is significantly easier than vice versa.

A widget-wrapping layer has been put on top of these widgets in order to achieve some widget-set independence. All widgets are created and modified via specific functions rather than via the generic interface of the X Toolkit. This facilitates the potential migration of the entire user interface onto another X Toolkit-based platform.

In addition, specialised VISTA widgets have been developed on top of the widget-wrapping layer for supporting TCAD-related information flow [54] [55]. The VISTA widgets are also created and accessed via specific functions, so that they can more easily be replaced by other widgets, should the need arise.

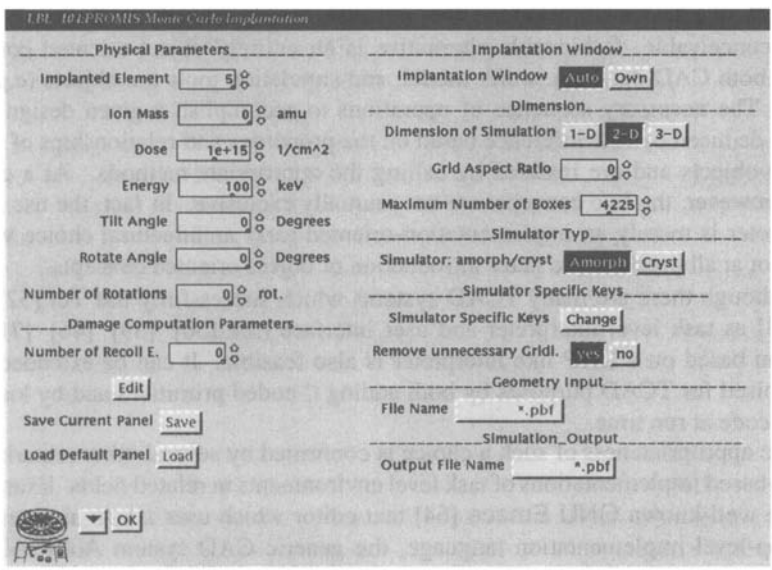

Figure 2.2 *This widget aggregate is the user interface for editing the parameters of a* PROMIS *Monte Carlo ion implantation step.*

Batch-mode applications which do not require direct user interaction (like all classical simulation programs) can also be equipped with a user interface, either for direct control (which is less often required) or for operation supervised by the Simulation Flow Control (SFC) module as described later. The situation is depicted in Figure 2.1, where the user interface is implemented as a task-level extension language (`LISP`) macro, entirely decoupled from the tool code. The widget macro shown in Figure 2.2 is such a tool control panel. Most tool control panels are created from formal specifications of the tools by simple interface generators which

are implemented in the extension language. This relieves the application engineer from the need to use low-level programming to create new tool control panels. The separation of the user interface code from the tool itself is an important means to stabilise and unify the user interface behaviour and to relieve the application code from the burden of the user interface implementation.

2.3 The Task Level in TCAD Systems

Looking at existing systems in CAD areas, it seems obvious that an interpreter is the classical solution for the task level environment ([2] [35] [36] [39] [42] [77]). Both single interactive actions and more complex flow control can be executed in an interpreted environment.

A UNIX- (or any other operating system) shell based solution does not fulfil the portability requirement, whereas the use of an integrating master application (like an interactive device editor) alone does not offer nearly enough flexibility.

A conceivable, fashionable alternative is an entirely object-oriented concept where both CAD data, e.g. wafer model, and simulation tools are objects (e.g. [8] [75]). The necessary sequence of operations to accomplish a given design goal can be deduced by rule inference based on the properties and relationships of these TCAD-objects and are realised by calling the appropriate methods. At a closer look, however, the two concepts are not mutually exclusive. In fact, the use of an interpreter is merely an implementation-oriented early architectural choice which does not at all preclude the latter introduction of object-oriented concepts.

Although there are many TCAD systems which successfully use Tcl [52] and Tk [53] as task level interpreter and user interface (see [36] [39] [46] [71]) a solution based on a LISP like interpreter is also feasible. It can be extended and customised for TCAD purposes by both adding C coded primitives and by loading LISP code at run time.

The appropriateness of such a choice is confirmed by several other remarkable, LISP-based implementations of task level environments in related fields. Examples are the well-known GNU Emacs [64] text editor which uses LISP as extension and top-level implementation language, the generic CAD system AutoCAD [2] which derives much of its success from third-party applications implemented in the SCHEME-like extension language AutoLISP, or Winterp [44] ("Widget *Interp*reter"), an experimental user interface prototyping environment which is part of the public domain X Window System distribution of the Massachusetts Institute of Technology. Other integrated CAD systems which allow for the definition of complex, data-driven control flows often use LISP as the major implementation language [8] [21].

A required task-level (LISP) function can be implemented as

- a LISP coded function loaded at run time,
- a C coded function which is linked with the LISP interpreter, or
- an external application, implemented as separate executable

Whichever method is used does hardly make a difference at the interpreter level. The functionality is always presented as a single `LISP` function.

For all batch mode applications (these are applications that do not require user interaction) the task level integration and the presentation integration (according to [30] and [73] that is the provision of applications with a homogeneous user interface) are accomplished within a single context. `LISP` functions are used to create a layer of virtual applications on the task level for one or more physical applications (executables). Figure 2.3 shows two characteristic examples.

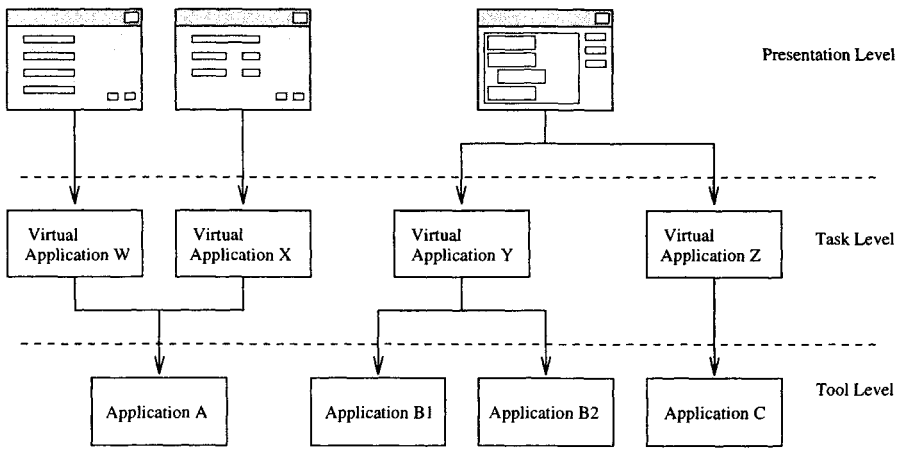

Figure 2.3 *A set of physical applications is mapped to a set of virtual applications on the task level. The user interface for these virtual applications is implemented in the task level programming environment, leaving the physical applications entirely unaffected.*

A typical example for a one-to-many mapping ("Application A" in Figure 2.3) is a device simulator, where for every desired type of device characterisation a function that performs the required simulation step and extracts the requested parameter is implemented ("Virtual Applications W" and "X" in Figure 2.3). For process simulation the situation is inverted. Several physical applications are run consecutively under control of a single presentation.

Note that this feature is only possible with a dedicated task level that separates the applications from the presentation level. This is an important prerequisite for the portability and continuity of the user interface. Furthermore, the intermediate task level layer allows the user interface to be tailored to design tasks instead of application peculiarities.

2.3.1 The VISTA solution

VISTA's task level environment is predominantly based on XLISP [4], a comprehensible and compact LISP interpreter. The highly portable C source code of this public-domain product is freely available.

To preserve the consistency and simplicity of XLISP and in order to provide a homogeneous procedural interface and programming environment, the X Window System interface in Figure 2.1 for XLISP was implemented from scratch in VISTA. As there are other C-coded parts of the framework which need to be accessible on the extension language level, a generic automatic method for linking given functions with the XLISP interpreter has been implemented.

The object-oriented callback concept of the X Toolkit has been generalised in a very straightforward manner and successfully applied to those parts of the TCAD framework where a strict decoupling of functional modules and high flexibility of the control flow is desirable. It is obvious that this is of special value for a flexible task level implementation.

Events coming from the X Window System system are passed to the XLISP interpreter. If a LISP expression has been associated with the activated widget at creation time, this expression is then evaluated by the interpreter and can be used to change parameter values, trigger the execution of a simulator or start the evaluation of a LISP program.

The same callback concept is also used for the control of simulator execution. If a simulation tool terminates, it signals the termination to the parent process, which again causes an associated callback expression to be evaluated. Callbacks can be triggered by the user interface, error handler, network layer, or by the termination of child processes.

Many computationally extensive design tasks, like statistical computations, exhibit intrinsic parallelism can hence be effectively and easily parallelised using the callback technique. Whenever there is no data dependency (which is the case for, e.g., stationary multiple operating point analysis) the simulations can be done in parallel and "only" need to be distributed on different workstations or servers and synchronised at the end.

Figure 2.4 shows two (simulation) processes which are run in parallel. The termination of simulation process 1 causes the termination callback 1 to be executed and can be used to trigger the computation of the next step. The choice between a purely sequential simulation flow and a parallel simulation flow is merely a matter of callback configuration and does not require any major change or adaptation of the task-level programs. The subprocesses can be distributed on different hosts to utilise the latent computational power of common workstation clusters.

The *Simulation Flow Control* (SFC) module [54] of VISTA is a high-level utility of the task level responsible for the definition, management, and execution of simulation sequences. The main field of application is the reproduction of fabrication processes, which usually involves several independent process simulators. The task flow is stored in a simulation flow description using symbolic names to call simulation tools. Larger process flows can be assembled from predefined sequences, the

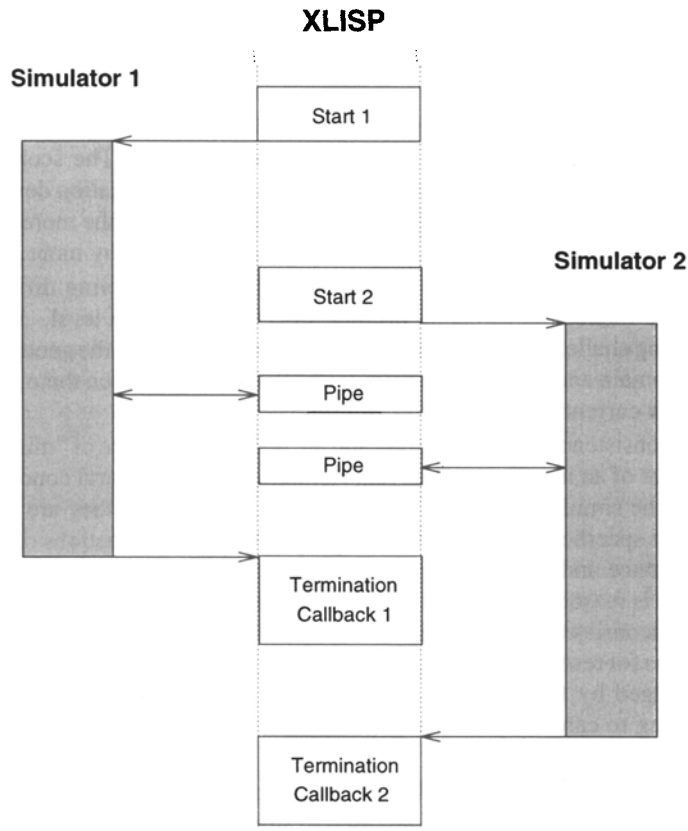

Figure 2.4 *Timing diagram of two (independent) simulations which are run in parallel under control of the* `XLISP` *interpreter.*

results of intermediate steps may remain available upon request for later analysis. The SFC functionality includes dependency analysis, interfaces for process parameter variation, and lot splits. The SFC is entirely coded in `LISP`, uses the callback concept for simulator synchronisation, and is executed by the `XLISP` interpreter.

2.4 Tool Integration in TCAD Systems

The main motivation for TCAD systems is the integration of different simulators. It is of course always possible to integrate another tool into an existing set of tools, but the decisive criterion to assess the support offered by the framework is the initial effort and the maintenance effort involved in the integration. Another criterion

for a proper application framework architecture is whether adaptations of already integrated tools are required when other, new tools are integrated or changed.

From a software point of view, a significant amount of TCAD methodology is technology-independent. This generic functionality must be provided as part of the framework, usually in form of libraries and applications. The scope of these services ranges from integrated CASE tools (e.g. [20]) for application development, scientific visualisation, interactive geometry editing facilities, to the more traditional TCAD methodology like grid generation, interpolation, and many more.

There are many situations in TCAD, especially when coupling different simulators, where interesting semantic problems occur on the data level. Among the most intriguing challenges are potential inconsistencies between the geometry of the simulation domain and one or more grids with attributes defined on them, altogether describing the current state of a wafer sub-domain.

These inconsistencies are inevitable, and are not a consequence of "misbehaving" applications or of an insufficient data representation or architectural concept. Given the fact that the simulators are granted "expert status" and that they are allowed to focus on their specific task, they will always use different abstractions of the wafer state and produce and affect just the data relevant to the specific problem they are modelling. It is reasonable and justified that they neglect all other issues, and hence these latent inconsistencies must necessarily arise. It is the duty of the framework to provide means for resolving those conflicts and inconsistencies. Once these semantic gaps are bridged by the framework, simulators can work together constructively without having to care for potential grid- and geometry-related conflicts they may create.

There are numerous grid-related problems and conflicts that do arise when multiple state-of-the-art simulation tools are used to simulate practical device fabrication steps. Some of these problems can be solved by interpolation services.

However, several problems require a grid handling in between two calls of simulation tools that goes beyond simple interpolation. Amongst them are the mapping of non-geometry-conforming grids onto geometry-conforming ones, splitting of grids which extend across several "segments" (layers), superposition of attributes (e.g. an additional boron implant in a structure where boron had already been present), a change in the geometry to which the doping information must be adapted, or the extraction of a subdomain of the geometry for a simulation tool.

In a more general formulation, it is required to map a set of grids of different types, with or without spatial overlap, geometry conforming or not, and with several "attributes" (dopants or such) defined on them, onto a new set of geometry conforming grids. The most important constraint of this mapping is the qualitative statement that there should be very little information or accuracy lost.

2.4.1 The VISTA solution

To solve this generic problem in VISTA rigorously, a framework tool has been implemented which is used to resolve all of the conflicts listed above (VORONOI [27]).

The re-gridding problem is sub-structured in several smaller problems which lead to fairly independent functional modules that perform operations on common data.

The basic idea to overcome any of the grid/geometry consistency problems rigorously is to treat grid and geometry points equally by merging the set of geometry points and the set of grid points and by finally (re-)triangulating the resulting point cloud using a constrained Delaunay triangulation.

Such functionality is a minimum requirement of a TCAD system, and the implementation of grid conversion and related problems in general is a key to the evaluation of TCAD systems.

2.5 A Shallow Trench Isolated CMOS Technology

A major challenge of multi-tool TCAD lies in the robust reproduction of increasingly complex manufacturing processes. This challenge is particularly prevalent in modern fabrication processes with a tight interaction between structuring and doping techniques, like planarisation and trench isolation processes. A fully planarised, trench-isolated $0.25\mu m$ CMOS process has been presented in [9] [10] [11] and is used here to demonstrate the coupling of several specialised process simulation tools by means of the VISTA framework.

The Monte Carlo simulation module of PROMIS [65] is used for the simulation of all ion implantation steps. It is a traditional FORTRAN simulator and has been integrated into VISTA directly using the PAI/PIF [17]. It reads only the device geometry and produces the resulting doping profiles on a single, non geometry-conforming tensor product grid which covers the bounding box of the entire device geometry.

TSUPREM-4 [69] is used to simulate the diffusion steps. It has been integrated by means of a wrapper which converts the wafer state from VISTA PIF to TIF (*TMA*'s Technology Interchange Format) and vice versa. TSUPREM-4 requires and produces doping information on several boundary-conforming triangular grids, one for each geometry segment.

The simulator ETCH [66] [67], newly developed using VISTA as implementation basis, utilising high-level libraries [16], is used for the simulation of etching and deposition steps. It reads only the geometry and produces a new, changed geometry. As ETCH uses a cellular data structure to perform purely geometrical operations, it cannot change the wafer state mesh accordingly. Hence, after etching and deposition steps, the (old) wafer state mesh does not match the new geometry produced.

The auxiliary tool SKETCH is used to perform simple changes of the wafer geometry due to spin-on, exposure, and mask strip steps.

The VORONOI re-gridding and interpolation service is used by the SFC to achieve and maintain consistency between grid and geometry, and to transform the structural and doping information between tool-specific and wafer state compliant form.

Table 2.2 *Process simulation flow for the* $0.25\mu m$*, fully planarised, shallow trench isolated CMOS process.*

step	simulator	SFC operation	parameters
1	SKETCH	create-subdomain	"Si", $5.6 \times 2.5\mu m$
2	SKETCH	spin-on	"Resist", $0.2\mu m$
3	SKETCH	expose	Mask 1
4	SKETCH	strip-material	"Exposed"
5	ETCH	plasma-etch	misc. etch rates, 320 seconds
6	SKETCH	strip-material	"Resist"
7	ETCH	iso-depo	"SiO2", 60 seconds at $125nm/s$
8	SKETCH	spin-on	"Resist", $0.25\mu m$
9	ETCH	iso-etch	misc. etch rates, 330 seconds
10	SKETCH	spin-on	"Si3N4", $1\mu m$
11	SKETCH	expose	Mask 2
12	SKETCH	strip-material	"Exposed"
13	PROMIS	monte-carlo-implant	"Phosphorus", $5 \cdot 10^{11}\ 1/cm^2, 60keV$
14	SKETCH	strip-material	"Si3N4"
15	SKETCH	spin-on	"Si3N4", $1\mu m$
16	SKETCH	expose	Mask 3
17	SKETCH	strip-material	"Exposed"
18	PROMIS	monte-carlo-implant	"Boron", $5 \cdot 10^{11}\ 1/cm^2, 60keV$
19	SKETCH	strip-material	"Si3N4"
20	TSUPREM-4	diffusion	inert, 3.5 minutes at 1000°C
21	SKETCH	spin-on	"Resist" $0.5\mu m$
22	SKETCH	expose	Mask 4
23	SKETCH	strip-material	"Exposed"
24	ETCH	plasma-etch	misc. etch rates, 200 seconds
25	SKETCH	strip-material	"Resist"
26	ETCH	iso-depo	"SiO2", 60 seconds at $1.1nm/s$
27	SKETCH	spin-on	"Poly", $0.25\mu m$
28	ETCH	iso-etch	misc. etch rates, 310 seconds
29	SKETCH	spin-on	"Si3N4", $1\mu m$
30	SKETCH	expose	Mask 5
31	SKETCH	strip-material	"Exposed"
32	PROMIS	monte-carlo-implant	"Arsenic", $2.5 \cdot 10^{14}\ 1/cm^2, 33keV$
33	SKETCH	strip-material	"Si3N4"
34	SKETCH	spin-on	"Si3N4", $1\mu m$
35	SKETCH	expose	Mask 6
36	SKETCH	strip-material	"Exposed"
37	PROMIS	monte-carlo-implant	"Boron", $2.5 \cdot 10^{14}\ 1/cm^2, 30keV$
38	SKETCH	strip-material	"Si3N4"
39	TSUPREM-4	diffusion	inert, 1 minute at 970°C

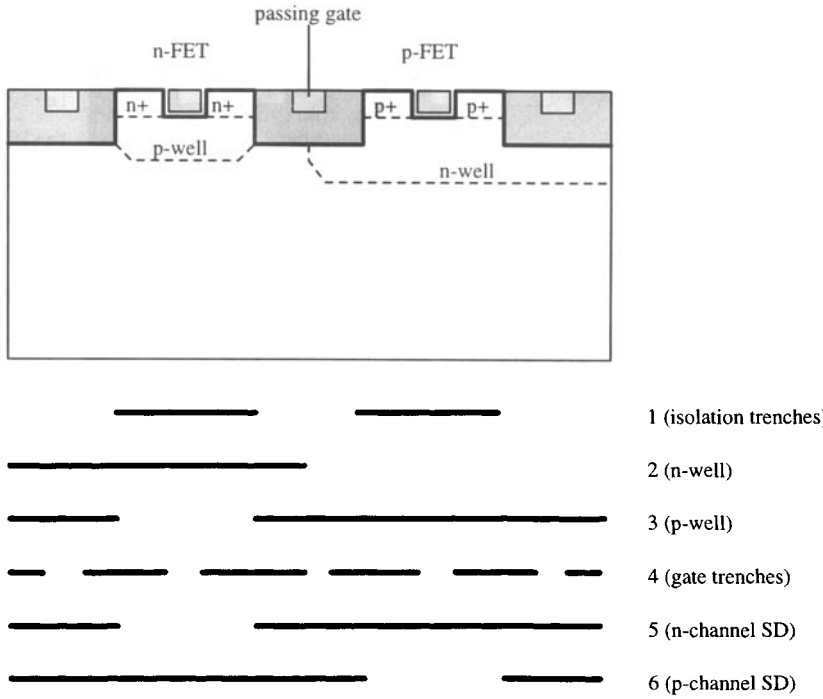

Figure 2.5 *Final CMOS structure and symbolic mask information.*

Table 2.2 shows the simulation steps executed by the SFC. This "virtual process" is a simplification of the real fabrication process. The virtuality of the simulation enables us to, e.g., "expose and develop" a nitride mask directly (step 10). This will be justified as long as the lithography and resist development of this mask layer is not a critical process step.

The desired final device structure is shown in Figure 2.5, together with the one-dimensional (symbolic) mask information used in steps 3, 11, 16, 22, 30, and 35.

Figure 2.6 shows the phosphorus doping of the n-well after the trenches for the device gates and for the passing gates have been etched. This etch step has removed part of the geometry at the wafer surface. The doping in the void has been automatically removed, and the grid has been made geometry-conforming to produce a consistent wafer state after completion of step 24.

Figure 2.7 is a screen dump of the `xpif2d` visualisation client showing the final planarised geometrical structure of the n-channel transistor along with the phosphorus concentration (after step 28). The visualisation uses the vector graphics widget of the VISTA user interface.

Figure 2.8 shows a typical VISTA session where the simulation flow editor is used to edit the process flow. The parameters of the last step (the TSUPREM-4 wrapper in step 39) of the process flow described is being edited. The parts of the

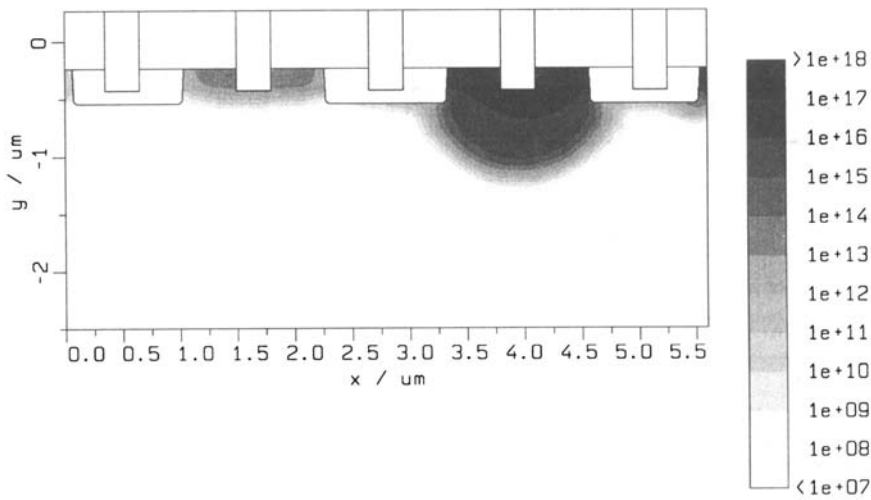

Figure 2.6 *n-well doping profile (phosphorus) after etching the trench for the planarised poly gate. Result of step 24.*

user interface shown are all created and executed by the task level shell.

2.6 An Ultra-Low-Power CMOS Technology

The application of TCAD is perfectly suited to investigate specific constraints of technologies. The following example demonstrates a technology design study for Ultra-Low-Power applications. By decreasing the supply voltage and adapting the threshold voltages, a great reduction of the power consumption can be achieved at the expense of an increase in gate delay. This can be compensated to a certain extent by employing parallelism in the systems design so that, for the same overall performance, the total power consumption is drastically reduced compared to conventional CMOS techniques [37]. We demonstrate the feasibility of ultra-low-power CMOS structures and determine a lower limit for the supply voltage, depending on the type of digital circuit technique.

The processes under consideration are recessed-well dual-gate processes with a very thin gate oxide (5nm and below) to obtain controllably low threshold voltages. The source/drain dopings are formed by single shallow implants and a conventional furnace anneal. The G/S and G/D overlap capacitances can be controlled with a spacer formed prior to the S/D implants.

As a consequence of the low voltages, ultra-low-power processes differ from conventional CMOS processes in several points: because of the low V_{DD}, the hot-carrier problem does virtually not exist, and therefore an LDD (lightly doped drain) process is not necessary. Also, no GIDL (gate induced drain leakage) can occur.

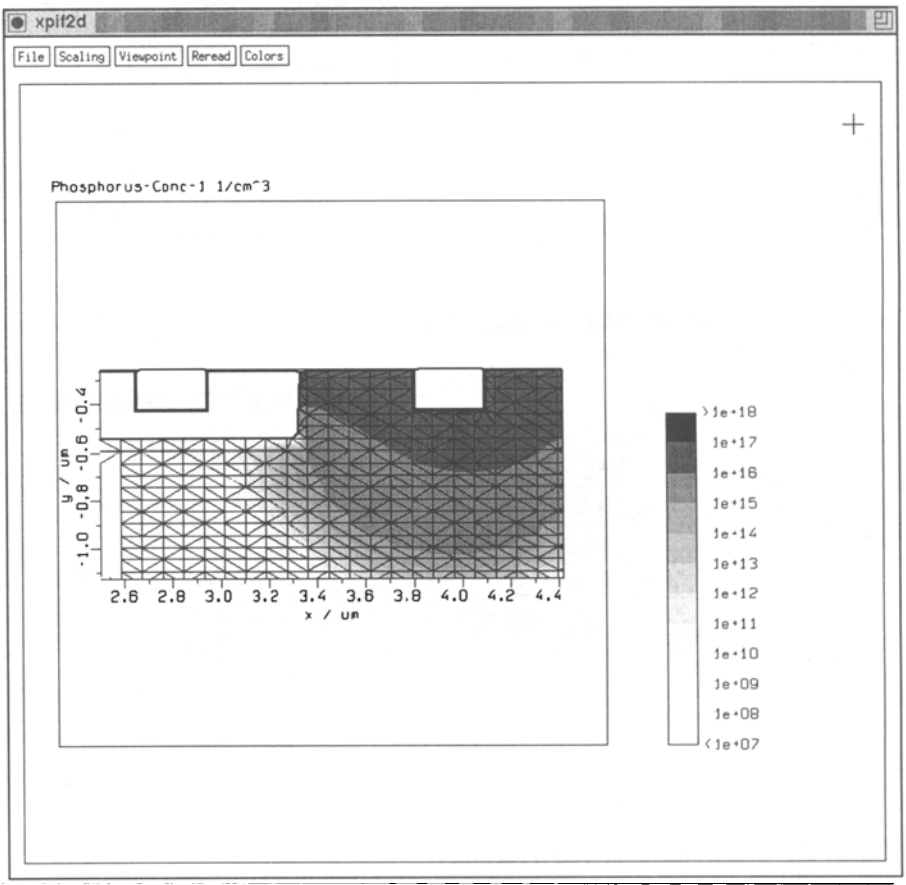

Figure 2.7 `xpif2d` *(the* VISTA *visualisation client) showing a detail of the n-well and a nearby passing gate structure. Result of step 28.*

As for very low V_{DD} the devices must operate in the weak-inversion regime, the difference of the carrier mobilities μ_n, μ_p can be roughly compensated by adjusting the threshold voltages to achieve symmetric inverter transfer characteristics. This compensation does not work, however, in the transient case because the speed is mainly determined by the strong-inversion part of the input characteristics. The sub-threshold behaviour is crucial because it determines the achievable ratio of I_{on}/I_{off}, which is limited by $e^{V_{DD}q/kT}$ and decreases as $V_{Tn,p}$ are made smaller. Therefore, "zero-V_T" transistors are not desirable at a very low V_{DD} of a few 100mV. On the other hand, if $V_{Tn,p}$ are too high the speed becomes unacceptably low. A major challenge is to achieve controllably low threshold voltages. Although the adjustment of $V_{Tn,p}$ with a bulk bias seems very attractive, this method is not practical (using a traditional layout) because of the significant overhead due to the additional lines

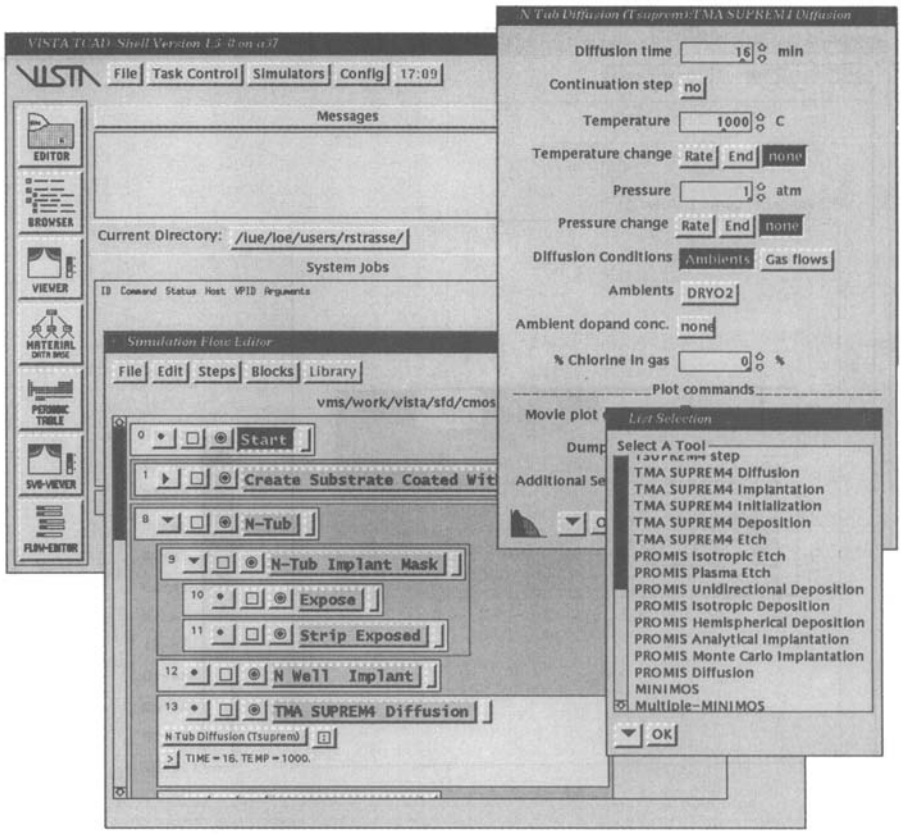

Figure 2.8 *Screen dump of* VISTA *session.*

for bulk biasing. There are, however, alternative biasing and layout methods to overcome this problem (see Chapter 3).

Another problem can arise from the very thin gate insulator. If one uses very thin thermally grown gate oxides (below 5nm), boron diffusion can considerably degrade the device behaviour. Also, boron segregation causes a deleterious effect, especially in the sub-threshold regime. On the other hand, there are several options for the gate insulator. Nitrides or oxinitrides may prove good alternatives to conventional (pure SiO_2) gate oxides. Silicon nitride can be used as an effective diffusion barrier, and ultra thin Si_3N_4 layers with low defect densities are easier to fabricate [45] than oxides. The tunnelling barrier, lower than that of SiO_2, is still acceptable for ultra-low-power CMOS devices because of the low voltages and also because a controllably small gate current is allowed.

Both process and device simulation were done using VISTA to allow for quick process design and evaluation [55]. For the electrical characterisation of the devices MINIMOS [19] [60] was used to calculate a matrix of drain currents $I_D(V_G, V_D)$

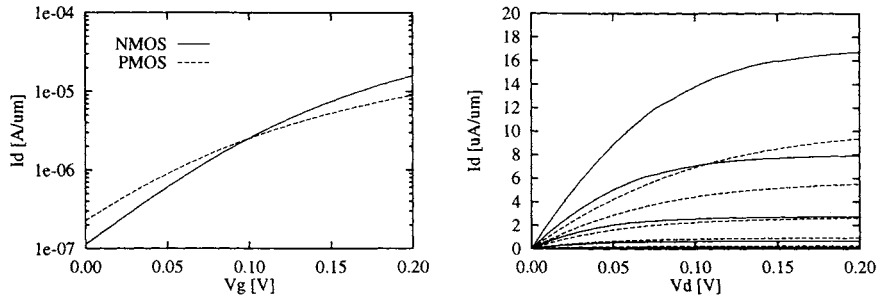

Figure 2.9 *Input characteristics (left) and output characteristics (right) of process A* (V_{DD} = 200mV).

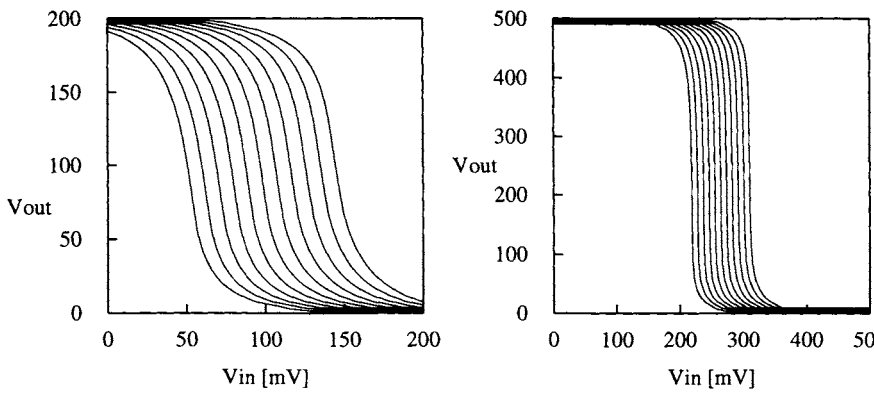

Figure 2.10 *Inverter transfer characteristics for (from left to right)* W_n/W_p = 10.0, 6.3, 4.0, 2.5, 1.6, 1.0, 0.63, 0.4, 0.25, 0.16, 0.1, *process A (left) at* V_{DD} = 200mV *and process B (right) at* V_{DD} = 500mV.

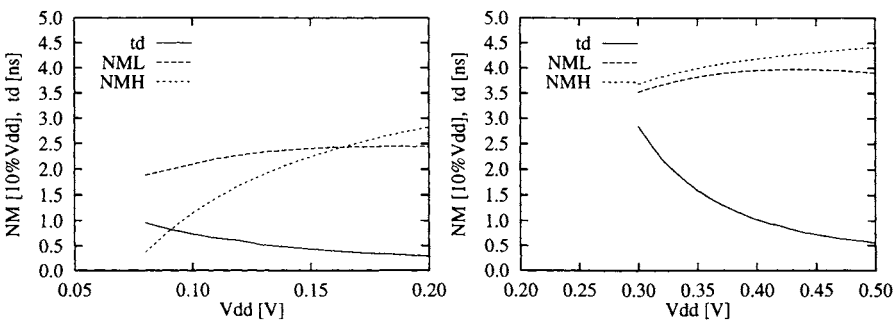

Figure 2.11 *Noise margins and delay time vs.* V_{DD}, *process A (left) and B (right).*

over a range of V_G and V_D for the p-channel and n-channel transistor. Based on these data, a fast and accurate table-driven DC analysis of simple gates and inverters is possible [59].

The bulk effect could also have been included, but for the given devices and voltages it was found insignificant. The dynamic behaviour was estimated from capacitance data obtained by AC analysis with MINIMOS.

The simulated processes were a 0.35μm process (A) for static logic and a 0.5μm process (B) for dynamic logic. The processes were designed for proper DC characteristics but were not optimised for speed. The device characteristics for process A are shown in Figure 2.9. Figure 2.10 shows the inverter transfer curves.

Figure 2.11 shows the noise margins and the inverter delay as a function of the supply voltage. It can be seen that a ring oscillator built with process A would work even at $V_{DD} = 80$mV, and by using additional inverters at the gate inputs and outputs one could also design digital circuits for $V_{DD} < 100$mV, but the overhead of the additional components would be considerable.

Table 2.3 *Noise margins (in $\%V_{DD}$) for a simple inverter and a 3-input NAND gate, and inverter delay, leakage time, switching energy, and static power consumption.*

process	V_{DD}	$V_{T,n}$	$V_{T,p}$
A	0.2	0.067	-0.059
B	0.5	0.26	-0.24

process	$I_{off,n}$	$I_{off,p}$	$I_{on,n}$	$I_{on,p}$
A	$0.14 \cdot 10^{-6}$	$0.27 \cdot 10^{-6}$	$16.7 \cdot 10^{-6}$	$9.4 \cdot 10^{-6}$
B	$0.7 \cdot 10^{-9}$	$2.8 \cdot 10^{-9}$	$25.6 \cdot 10^{-6}$	$16.7 \cdot 10^{-6}$

Table 2.4 *Simulated device characteristics. The threshold voltage was defined as $|I_D(V_T)| = 1\mu$A/μm. All voltages are in V, all currents are in A/μm.*

process	$NM_{H,inv}$	$NM_{L,inv}$	$NM_{H,gate}$	$NM_{L,gate}$
A	28	23	13	39
B	38	44	31	49

process	t_d	t_l	E_s	P_{stat}
A	0.29ns	7.2ns	0.65fJ	41nW
B	0.55ns	1.3μs	4.3fJ	0.88nW

For Process B, the ratios of I_{on}/I_{off} in Table 2.4 are in the order of 10^4 and the ratio of τ_l/τ_d (leakage time to delay time) is about 2300, which is rather low for

dynamic logic. For process A, it can be seen from Table 2.3 that for 3-input NANDs with minimal transistors the high-noise-margin NM_H is already very low.

From these data we conclude that the limits for the supply voltage will be at 200mV for static logic and 500mV for dynamic logic with a fan-in of 3 at T = 300K.

To determine how close the two processes are to the absolute lower limits for V_{DD} we can use an analytical MOSFET model in the weak-inversion regime (see also Chapters 3 and 14): $I_D = I_s\,(1 - \exp(-V_{DS}/U_T))\exp(V_{GS}/U_T)$ from which we can derive an expression for the maximum inverter gain: $-G_{max} = \exp(\frac{V_{DD}/2}{U_T} - 1)$.

Using the criterion $-G_{max,crit} > 4$ for static logic with a fan-in of ≤ 3 yields a minimum supply voltage of $V_{DD,crit} = 83\text{mV} = 3.2U_T = X_{crit}U_T$ which is much less than $V_{DD} = 200\text{mV} = 7.8U_T = XU_T$, corresponding to a maximum gain of $-G_{max,opt} = 47$. However, the actual maximum gain is $-G_{max} = 7.7$, corresponding to a "fictitious" supply voltage of $V_{DD,eff} = 112\text{mV} = 4.3U_T = X_{eff}U_T$.

This can be interpreted as follows: the ratio X/X_{crit} tells how close a technology is to the lower limit of V_{DD}, regardless of the circuit performance. X/X_{eff} tells the percentage of V_{DD} which is used up to fulfil the criterion; the remainder serves to increase the performance. X_{eff}/X_{crit} is an indirect measure for the sensitivity of a process (in this case the ratio is 1.34; a value below 1 would mean a violation of the criterion).

Similarly, a criterion for the ratio I_{on}/I_{off} can be defined with $X_{crit} = \ln(I_{on}/I_{off})_{crit} = \ln(I(V_{DD,crit})/I(0))$, $X = V_{DD}/U_T$, and $X_{eff} = \ln(I_{on}/I_{off}) = \ln(I(V_{DD}/I(0))$. Applying $(I_{on}/I_{off})_{crit} = 10^4$ with $I_{on}/I_{off} = \sqrt{(I_{on,n}I_{on,p}/I_{off,n}I_{off,p})}$ to process B yields $X : X_{eff} : X_{crit} = 19.2 : 9.6 : 9.2$ which suggests a high sensitivity to process fluctuations.

2.7 A High-Electron-Mobility-Transistor Technology

One of the major issues of future TCAD applications is the investigation of novel device structures, amongst which the heterostructure devices have a prominent position. The following discussion of HEMT (high electron mobility transistor) simulation also highlights some aspects of TCAD in the simulation of such devices [62].

Figure 2.12 gives the structure of a low-noise HEMT which is currently in industrial use. The device consists of several layers, of which the undoped channel ($In_{0.2}Ga_{0.8}As$) is responsible for current transport, whereas the doped supply layer ($Al_{0.2}Ga_{0.8}As$) isolates the channel from the gate due to its large band gap. The supply layer doping stops about 2 nm above the channel to reduce scattering induced by dopants. The channel sits on top of an undoped substrate (GaAs), and source and drain contacts are put on special cap layers (GaAs) on both sides. The doping of the supply layer is typically some 10^{17} to some 10^{18} cm^{-3}.

It is of crucial importance for the simulation of modern devices that a framework be able to handle not only the multi-layer geometry (consisting of 15 layers and

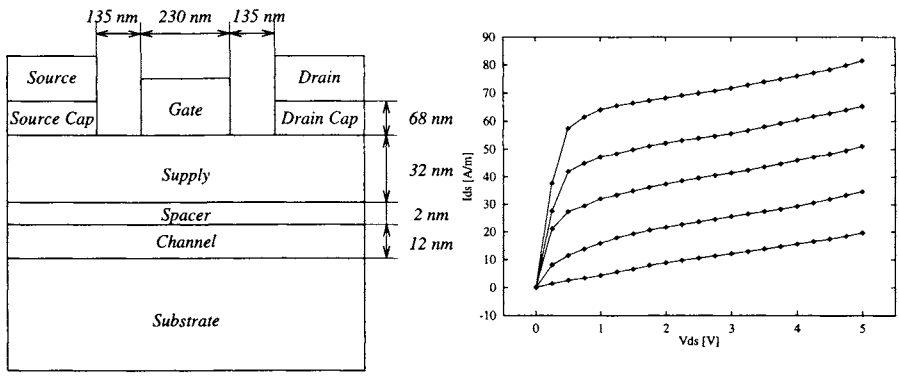

Figure 2.12 *Structure (left) and IV-Characteristics (right) of a HEMT.*

more) of such devices, but also the material information. The latter will in general not be restricted to homogeneous material alloys within the single layers, nor will it suffice to support only a handful select material types. Thus, a powerful data format must provide a solid basis for such simulations.

As for the material types, VISTA provides a material database which is used in the simulations shown here. It provides a simulation tool with basic material parameters and is extensible for the needs of specific users. Several mobility functions that can be chosen independently for the various layers are also contributed to this simulation.

To obtain reasonable results, a proper model for the carrier current and energy flux at the hetero-junction is indispensable, since carrier heating in the channel region is one main effect to determine the current. The model must be physically correct as well as numerically stable, which implies that the handling of the segments must take the segment boundaries into special account. Unless one is interested in the quantum physical situation at the boundary, the boundary should be simulated as an abrupt junction in the band edge energies and effective densities of states. These assumptions lead to an electron concentration which is discontinuous at the hetero-junction, having two distinct values at the boundary, each of which is only valid for one side of the junction.

The IV-characteristics of the HEMT (Figure 2.12) show a behaviour similar to a MOS-transistor within the usual operating area, except for the better high frequency properties and different breakdown mechanisms. Alas, without a detailed account for the specific structure it is impossible to obtain these characteristics.

It is, however, not the only task of the framework to provide material parameters. Important are tasks like design optimisation for the device in question or parameter fitting to adjust the various model parameters to the local process data. In these points, the notion of a framework shows its general superiority above single simulation runs, each of them triggered and evaluated separately by the user.

Figure 2.13 and 2.14 show the electron concentration and temperature in the HEMT presented above. One might note the different concentration values in the

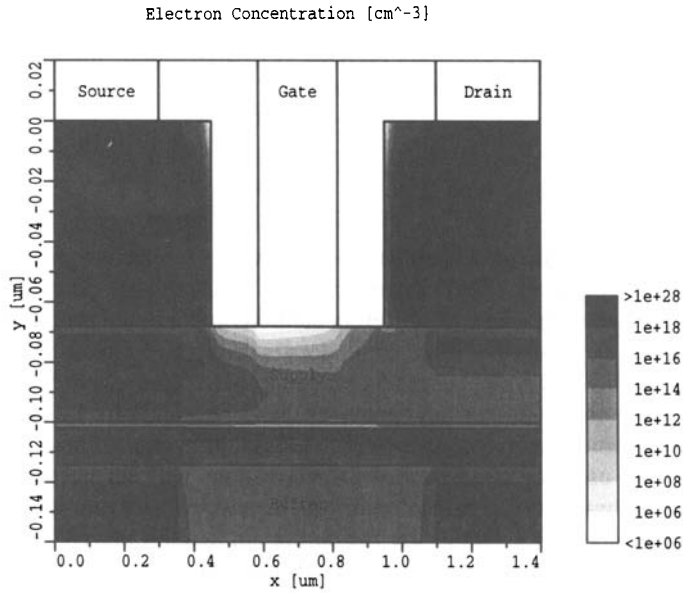

Figure 2.13 *Electron concentration in the HEMT.*

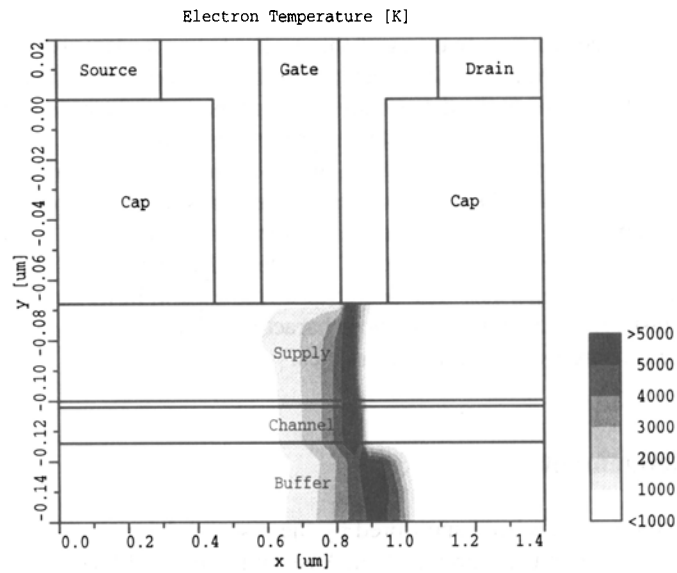

Figure 2.14 *Electron temperature in the HEMT.*

different layers, which are connected at the abrupt hetero-junction by a model that takes into account the different band edge energies on both sides. The temperature in the channel reflects the influences of the high field under the gate and the reduction due to doping in the source and drain regions. A hydrodynamic model was applied in the channel region to obtain these results.

2.8 Conclusion

It is only due to the dedicated application-framework architecture that the simulation tools integrated into a TCAD system can still focus on a specific technological problem and may even use entirely different methods and internal problem representations. The rapid evolution of these simulators is driven by fabrication technology. The generic, technology-independent functionality of the framework helps to bridge the gaps between the simulators and provides the device and process engineer with a more stable, homogeneous representation of simulation capabilities.

Whereas the users of an integrated TCAD system will not necessarily notice architectural software shortcomings, both tool developers and tool integrators will experience the retarding effect of the prevailing application-driven TCAD system approach. It is hence vital for the TCAD framework that it addresses properly not only the application of methodology, but also the entire creation process of TCAD methodology. This definitely includes support for tool integration and tool development.

The study of Ultra-Low-Power device technology, performed within the VISTA system, predicts convincing features of this technology. It also gives evidence of the practical usability of a TCAD framework in the development and verification of new designs. According to the data obtained from this study, the possibilities of low power concepts are by far not exhausted in present applications.

The HEMT transistor which has been analysed in the last example emphasises yet another aspect of TCAD in the future: frameworks will no longer be allowed to content themselves with restricting to a minimum concept of a data format. Flexibility in device geometry must come with flexibility in the physical data, as are materials and their properties. The power of a framework, however, shows best in the typical tasks required in future device characterisation: design optimisation and parameter fitting are two prominent examples.

Acknowledgements

The presented work has been sponsored by the research laboratories of AUSTRIA MIKRO SYSTEME at Unterpremstätten, Austria; DIGITAL EQUIPMENT at Hudson, USA; HITACHI at Tokyo, Japan; MOTOROLA at Austin, USA; NATIONAL SEMICONDUCTOR at Santa Clara, USA; SIEMENS at Munich, Germany; and SONY at Atsugi, Japan, and by the "Forschungsförderungsfonds für die gewerbliche

Wirtschaft", project 2/285 and project 2/299, as part of ADEQUAT (JESSI project BT1B), ESPRIT project 7236, and by ADEQUAT II (JESSI project BT11), ESPRIT project 8002.

Reference

[1] ASENTE, P.J., AND SWICK, R.R. *X Window System Toolkit, The Complete Programmer's Guide and Specification.* Digital Press, 1990.

[2] AUTODESK AG. *AUTOCAD Release 11 Reference Manual,* 1990. Publication AC11RM.E1.

[3] AXELRAD, V., GRANIK, Y., AND JEWELL, R. CAESAR: The Virtual IC Factory as an Integrated TCAD User Environment. In: Fasching et al. [15], pp. 293–307.

[4] BETZ, D.M. *XLISP: An Object-Oriented Lisp, Version 2.1,* 1989.

[5] BONING, D.S., MCILRATH, M.B., JR., P. PENFIELD, AND SACHS, E.M. A General Semiconductor Process Modeling Framework. *IEEE Trans.Semiconductor Manufacturing 5,* 4 (1992), 266–280.

[6] BROCKMAN, J.B., AND DIRECTOR, S.W. A Macromodeling Approach to Process Simulator Tuning. In: NUPAD III [48], pp. 17–18.

[7] COLE, D.C., BUTURLA, E.M., FURKAY, S.S., VARAHRAMYAN, K., SLINKMAN, J., MANDELMAN, J.A., FOTY, D.P., BULA, O., STRONG, A.W., PARK, J.W., LINTON JR., T.D., JOHNSON, J.B., FISCHETTI, M.V., LAUX, S.E., COTTRELL, P.E., LUSTIG, H.G., PILEGGI, F., AND KATCOFF, D. The Use of Simulation in Semiconductor Technology Development. *Solid-State Electron. 33,* 6 (1990), 591–623.

[8] DANIELL, J., AND DIRECTOR, S.W. An Object Oriented Approach to CAD Tool Control. *IEEE Trans.Computer-Aided Design 10,* 6 (June 1991), 698–713.

[9] DAVARI, B., KOBURGER, C.W., SCHULZ, R., WARNOCK, J.D., FURUKAWA, T., W.JOST, SCHWITTEK, W.G., DEBROSSE, J.K., KERBAUGH, M.L., AND MAUER, J.L. A New Planarization Technique, Using a Combination of RIE and Chemical Mechanical Polish (CMP). In Proc: *Int.Electron Devices Meeting* (1989), pp. 61–64.

[10] DAVARI, B., CHANG, W.H., PETRILLO, K.E., WONG, C.Y., MOY, D., TAUR, Y., WORDEMAN, M.R., SUN, J. YUAN-CHEN, HSU, C.C.H., AND POLCARI, M.R. A High-Performance 0.25-μm CMOS Technology: II–Technology. *IEEE Trans.Electron Devices 39,* 4 (1992), 967–975.

[11] D.S.WEN, CHANG, W.H., LII, Y., MEGDANIS, A.C., MCFARLAND, P., AND BRONNER, G. A Fully Planarized 0.25μm CMOS Technology. In Proc: *Symposium on VLSI Technology* (1991), pp. 83–84.

[12] DUTTON, R.W., AND GOOSSENS, R.J.G. Technology CAD at Stanford University: Physics, Algorithms, Software, and Applications. In: Fasching et al. [15], pp. 113–130.

[13] DUVALL, S.G. An Interchange Format for Process and Device Simulation. *IEEE Trans.Computer-Aided Design 7,* 7 (1988), 741–754.

[14] FASCHING, F., FISCHER, C., HALAMA, S., PIMINGSTORFER, H., READ, H., SELBERHERR, S., STIPPEL, H., TUPPA, W., VERHAS, P., AND WIMMER, K. A New Open Techology CAD System. In Proc: *21st European Solid State Device Research Conference - ESSDERC'91* (Montreux, Switzerland, 1991), M. Ilegems and M. Dutoit, Eds., vol. 15 of *Microelectr.Eng.*, Elsevier, pp. 217–220.

[15] FASCHING, F., HALAMA, S., AND SELBERHERR, S., Eds. *Technology CAD Systems* (1993), Springer.

[16] FASCHING, F. *The Viennese Integrated System for Technology CAD Applications–Data Level Design and Implementation*. Dissertation, Technische Universität Wien, 1994.

[17] FASCHING, F., TUPPA, W., AND SELBERHERR, S. VISTA-The Data Level. *IEEE Trans.Computer-Aided Design 13*, 1 (1994), 72–81.

[18] FICHTNER, W., AND AEMMER, D., Eds. *Simulation of Semiconductor Devices and Processes* (Konstanz, 1991), vol. 4, Hartung-Gorre.

[19] FISCHER, C., HABAŠ, P., HEINREICHSBERGER, O., KOSINA, H., LINDORFER, PH., PICHLER, P., PÖTZL, H., SALA, C., SCHÜTZ, A., SELBERHERR, S., STIFTINGER, M., AND THURNER, M. *MINIMOS 6.0 User's Guide*. Institute for Microelectronics, Technical University Vienna, Mar. 1994.

[20] FISHER, A.S. *CASE Using Software Development Tools*. Wiley, 1991.

[21] FUNAKOSHI, K., AND MIZUNO, K. A Rule-Based VLSI Process Flow Validation System With Macroscopic Process Simulation. *IEEE Trans.Semiconductor Manufacturing 3*, 4 (Nov. 1990), 239–246.

[22] GOPALARAO, K.S.V., MOZUMDER, P.K., AND BONING, D.S. An Integrated Technology CAD System for Process and Device Designers. *IEEE Trans.VLSI Systems 1*, 4 (Dec. 1993), 482–490.

[23] GOUGH, P.A., JOHNSON, M.K., WALKER, P., AND HERMANS, H. An Integrated Device Design Environment for Semiconductors. *IEEE Trans.Computer-Aided Design 10*, 6 (June 1991), 808–821.

[24] GOUGH, P.A. An Integrated Design Environment for Semiconductors. In: Fasching et al. [15], pp. 131–146.

[25] HALAMA, S., FASCHING, F., PIMINGSTORFER, H., TUPPA, W., AND SELBERHERR, S. Consistent User Interface and Task Level Architecture of a TCAD System. In: NUPAD IV [49], pp. 237–242.

[26] HALAMA, S., FASCHING, F., FISCHER, C., KOSINA, H., LEITNER, E., PICHLER, CH., PIMINGSTORFER, H., PUCHNER, H., RIEGER, G., SCHROM, G., SIMLINGER, T., STIFTINGER, M., STIPPEL, H., STRASSER, E., TUPPA, W., WIMMER, K., AND SELBERHERR, S. The Viennese Integrated System for Technology CAD Applications. In: Fasching et al. [15], pp. 197–236.

[27] HALAMA, S. *The Viennese Integrated System for Technology CAD Applications–Architecture and Critical Software Components*. Dissertation, Technische Universität Wien, 1994.

[28] HOBLER, G., HALAMA, S., WIMMER, W., SELBERHERR, S., AND PÖTZL, H. RTA-Simulations with the 2-D Process Simulator PROMIS. In: NUPAD III [48], pp. 13–14.

[29] HOPPER, P.J., AND BLAKEY, P.A. The MASTER Framework. In: Fasching et al. [15], pp. 275–292.

[30] I.THOMAS, AND NEJMEH, B.A. Definitions of Tool Integration for Environments. *IEEE Software 9*, 2 (Mar. 1992), 29–35.

[31] JACOBS, H., HÄNSCH, W., HOFMANN, F., JACOBS, W., PAFFRATH, M., RANK, E., STEGER, K., AND WEINERT, U. SATURN - A Device Engineer's Tool for Optimizing MOSFET Performance and Lifetime. In: NUPAD III [48], pp. 55–56.

[32] JACOBS, W. The SATURN Technology CAD System. In: Fasching et al. [15], pp. 147–162.

[33] JONES, O. *Introduction to the X Window System.* Prentice-Hall, 1989.

[34] KATO, K., SHIGYO, N., WADA, T., ONGA, S., KONAKA, M., AND TANIGUCHI, K. A Supervised Simulation System for Process and Device Design Based on a Geometrical Data Interface. *IEEE Trans.Electron Devices ED-34*, 10 (1987), 2049–2058.

[35] KLEINFELDT, S., GUINEY, M., MILLER, J.K., AND BARNES, M. Design Methodology Management. *Proc.IEEE 82*, 2 (1994), 231–250.

[36] KNEPPER, R.W., JOHNSON, J.B., FURKAY, S., SLINKMAN, J., TIAN, X., BUTURLA, E.M., YOUNG, R., FIORENZA, G., LOGAN, R., HUANG, Y.S., O'BRIEN, R.R., MURTHY, C.S., MURLEY, P.C., PENG, J., TANG, H.H.K., SRINIVASAN, G.R., PELELLA, M.M., SUNDERLAND, D.A., MANDELMAN, J., LIEBER, D., FARRELL, E., AND KURASIC, M. Technology CAD at IBM. In: Fasching et al. [15], pp. 25–62.

[37] LIU, D., AND SVENSSON, CH. Trading Speed for Low Power by Choice of Supply and Threshold Voltages. *IEEE J.Solid-State Circuits 28*, 1 (1993), 10–17.

[38] LLOYD, P., DIRKS, H.K., PRENDERGAST, E.J., AND SINGHAL, K. Technology CAD for Competitive Products. *IEEE Trans.Computer-Aided Design 9*, 11 (1990), 1209–1216.

[39] LLOYD, P., MCANDREW, C.C., MCLENNAN, M.J., NASSIF, S., SINGHAL, K., SINGHAL, KU., ZEITZOFF, P.M., DARWISH, M.N., HARUTA, K., LENTZ, J.L., VUONG, H., PINTO, M.R., RAFFERTY, C.S., AND KIZILYALLI, I.C. Technology CAD at AT&T. In: Fasching et al. [15], pp. 1–24.

[40] LORENZ, J., HILL, C., JAOUEN, H., LOMBARDI, C., LYDEN, C., DE MEYER, K., PELKA, J., PONCET, A., RUDAN, M., AND SOLMI, S. The STORM Technology CAD System. In: Fasching et al. [15], pp. 163–196.

[41] MAR, J., BHARGAVAN, K., DUVALL, S.G., FIRESTONE, R., LUCEY, D.J., NANDGAONKAR, S.N., WU, S., YU, K.-S., AND ZARBAKHSH, F. EASE—An Application-Based CAD System for Process Design. *IEEE Trans.Computer-Aided Design CAD-6*, 6 (1987), 1032–1038.

[42] MAR, J. Technology CAD at Intel. In: Fasching et al. [15], pp. 63–74.

[43] MATSUO, H., MASUDA, H., YAMAMOTO, S., AND TOYABE, T. A Supervised Process and Device Simulation for Statistical VLSI Design. In: NUPAD III [48], pp. 59–60.

[44] MAYER, N. WINTERP: An Object-Oriented Rapid Prototyping, Development and Delivery Environment for Building User-Customizable Applications with the OSF/Motif UI Toolkit. Tech. rep., Hewlett-Packard Laboratories, Palo Alto, 1991.

[45] MORIMOTO, T., MOMOSE, H.S., TAKAGI, S., YAMABE, K., AND IWAI, H. Ultrathin Nitride Gate MISFET Operating with Tunneling Gate Current. In Proc: *Ext. Abstr. of the 22nd Conf. on Solid State Devices and Materials* (Sendai, 1990), pp. 361–364.

[46] NEUREUTHER, A., WANG, R., AND HELMSEN, J. Perspective on TCAD Integration at Berkeley. In: Fasching et al. [15], pp. 75–82.

[47] NISHI, K., AND UEDA, J. Technology CAD at OKI. In: Fasching et al. [15], pp. 255–274.

[48] *Proc. Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits NUPAD III* (Honolulu, 1990).

[49] *Proc. Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits NUPAD IV* (Seattle, 1992).

[50] NYE, A. *Xlib Programming Manual*, second ed., vol. 1 of *The X Window System Series*. O'Reilly & Associates, 1990.

[51] O'REILLY, T. *X Toolkit Intrinsics Reference Manual*, first ed., vol. 5 of *The X Window System Series*. O'Reilly & Associates, 1990.

[52] OUSTERHOUT, J.K. Tcl: An Embeddable Command Language. In Proc: *1990 Winter USENIX Conference Proceedings* (1990), pp. 133–146.

[53] OUSTERHOUT, J.K. An X11 Toolkit Based on the Tcl Language. In Proc: *1991 Winter USENIX Conference Proceedings* (1991), pp. 105–115.

[54] PICHLER, CH., AND SELBERHERR, S. Process Flow Representation within the VISTA Framework. In: Selberherr et al. [61], pp. 25–28.

[55] PICHLER, CH., AND SELBERHERR, S. Rapid Semiconductor Process Design within the VISTA Framework: Integration of Simulation Tools. In Proc: *Proceedings of the IASTED International Conference* (Pittsburgh, PA, USA, 1993), M. Hamza, Ed., Modelling and Simulation, The International Association of Science and Technology for Development, pp. 147–150.

[56] PIMINGSTORFER, H., HALAMA, S., SELBERHERR, S., WIMMER, K., AND VERHAS, P. A Technology CAD Shell. In: Fichtner and Aemmer [18], pp. 409–416.

[57] PRENTICE-HALL. *OSF/Motif Programmer's Guide, Release 1.1*, 1991.

[58] SCHECKLER, E.W., WONG, A.S., WANG, R.H., CHIN, G., CAMAGNA, J.R., NEUREUTHER, A.R., AND DUTTON, R.W. A Utility-Based Integrated System for Process Simulation. *IEEE Trans.Computer-Aided Design 11*, 7 (1992), 911–920.

[59] SCHROM, G., LIU, D., FISCHER, C., PICHLER, CH., SVENSSON, CH., AND SELBERHERR, S. VLSI Performance Analysis Method for Low-Voltage Circuit Operation. In Proc: *Proc.ICSICT* (Beijing, China, 1995), pp. 328–330.

[60] SELBERHERR, S., AND KOSINA, H. Simulation of Nanometer MOS-Devices with MINIMOS. In Proc: *1990 VLSI Process/Device Modeling Workshop* (Kawasaki, Japan, 1990), pp. 2–5.

[61] SELBERHERR, S., STIPPEL, H., AND STRASSER, E., Eds. *Simulation of Semiconductor Devices and Processes* (1993), vol. 5, Springer.

[62] SIMLINGER, T., KOSINA, H., ROTTINGER, M., AND SELBERHERR, S. MINIMOS-NT: A Generic Simulator for Complex Semiconductor Devices. In Proc: *ESSDERC'95 - 25th European Solid State Device Research Conference* (Gif-sur-Yvette Cedex, France, 1995), H. de Graaff and H. van Kranenburg, Eds., Editions Frontieres, pp. 83–86.

[63] SIMPSON, M.R. PRIDE: An Integrated Design Environment for Semiconductor Device Simulation. *IEEE Trans.Computer-Aided Design 10*, 9 (Sept. 1991), 1163–1174.

[64] STALLMAN, R. *GNU Emacs Manual*, Oct. 1986.

[65] STIPPEL, H., HALAMA, S., HOBLER, G., WIMMER, K., AND SELBERHERR, S. Adaptive Grid for Monte Carlo Simulation of Ion Implantation. In: NUPAD IV [49], pp. 231–236.

[66] STRASSER, E., AND SELBERHERR, S. A General Simulation Method for Etching and Deposition Processes. In: Selberherr et al. [61], pp. 357–360.

[67] STRASSER, E., SCHROM, G., WIMMER, K., AND SELBERHERR, S. Accurate Simulation of Pattern Transfer Processes Using Minkowski Operations. *IEICE Transactions on Electronics E77-C* (1994), 92–97.

[68] TANABE, N. Technology CAD at NEC. In: Fasching et al. [15], pp. 237–254.

[69] TECHNOLOGY MODELING ASSOCIATES, INC. *TMA TSUPREM-4, Two-Dimensional Process Simulation Program Version 6*, Dec. 1993.

[70] WALKER, D.M.H., KELLEN, CH.S., SVOBODA, D.M., AND STROJWAS, A.J. The CDB/HCDB Semiconductor Wafer Representation Server. *IEEE Trans.Computer-Aided Design 12*, 2 (1993), 283–295.

[71] WALKER, D.M.H., KIBARIAN, J.K., KELLEN, CH.S., AND STROJWAS, A.J. A TCAD Framework for Development and Manufacturing. In: Fasching et al. [15], pp. 83–112.

[72] WANG, R.H., GABARA, A., AND NEUREUTHER, A.R. BTU — Berkeley Topography Utilities for Linking Topography and Impurity Profile Simulations. In: NUPAD IV [49], pp. 237–242.

[73] WASSERMAN, A.I. Tool Integration in Software Engineering Environments. In Proc: *Software Engineering Environments* (New York, 1989), no. 467 in Lecture Notes in Computer Science, Springer, pp. 137–149.

[74] WENSTRAND, J.S., IWAI, H., NORISHIMA, M., TANIMOTO, H., WADA, T., AND DUTTON, R.W. Intelligent Simulation for Optimization of Fabrication Processes. In: NUPAD III [48], pp. 15–16.

[75] WENSTRAND, J.S. An Object-Oriented Model for Specification, Simulation, and Design of Semiconductor Fabrication Processes. Tech. Rep. ICL91-003, Integrated Circuits Laboratory, Stanford University, Mar. 1991.

[76] WONG, A.S., AND NEUREUTHER, A.R. The Intertool Profile Interchange Format: A Technology CAD Environment Approach. *IEEE Trans.Computer-Aided Design 10*, 9 (Sept. 1991), 1157–1162.

[77] WONG, A.S. *Technology Computer-Aided Design Frameworks and the PROSE Implementation*. PhD thesis, University of California, Berkeley, 1992.

World Wide Web Address

`http://www.iue.tuwien.ac.at/`

Low-power HF microelectronics: a unified approach, Edited by Gerson A. S. Machado. ISBN 0 85296 874 4