Automatic Technology Migration of Analog IC Designs using Generic Cell Libraries

José Cachaço[1], Nuno Machado[1], Nuno Lourenço[1], Jorge Guilherme[2] e Nuno Horta[1]
[1]Instituto de Telecomunicações/Instituto Superior Técnico – ULisbon. Lisboa, Portugal
[2]Instituto de Telecomunicações/Instituto Politécnico de Tomar, Tomar, Portugal
nlourenco; nuno.horta@lx.it.pt; jorge.guilherme@ipt.pt

***Abstract*— This paper addresses the problem of automatic technology migration of analog IC designs. The proposed method introduces a new level of abstraction, for EDA tools addressing analog IC design, allowing a systematic and effortless adaption of a design to a new technology. The new abstraction level is based on generic cell libraries, which includes topology and testbenches descriptions for specific circuit classes. In addition to technology independence, reusing the testbenches when adding new topologies for the already implemented circuit classes also improves design productivity. The new method is implemented and tested using a state-of-the-art multi-objective multi-constraint circuit-level optimization tool for circuit sizing, and is validated for the design and optimization of continuous-time comparators, including technology migration between two different design nodes, respectively, XFAB 350 nm technology and ATMEL 150 nm SOI technology.**

Keywords*— *Analog Integrated Circuits; Electronic Design Automation; Technology Migration; Generic Cell Libraries

I. Introduction

In a System-on-Chip (SoC), the analog part, which occupies approximately only 20% of the chip area, can be the main bottleneck in design time and complexity. Intellectual property (IP) reuse supported by well-defined automated synthesis methodologies and tools is a common practice in digital design [1]. Analog IP reuse, on the other hand, is much more expensive due to their higher sensitivity to process variations and strong dependence of technology specific devices and design rules, etc. [2-4]. Therefore, the lack of analog design reuse is a major contributor for the analog-digital productivity gap. As new technologies become available, old designs need to be ported, and this technology migration process is not straightforward but time and resources consuming. Electronic Design Automation (EDA) tools do not only help finding optimal performance solutions but can also ease design reuse through technology migration [5-8].

In this work, an abstract, technology-independent interface is proposed that facilitates technology migration and analog design reuse. The proposed approach is implemented and tested using the state-of-the-art multi-objective multi-constraint circuit-level optimization tool descried in [5] for circuit sizing. The method was validated for the design and optimization of continuous-time comparators, including technology migration between two different design nodes, respectively, XFAB 350 nm technology (XH035) and ATMEL 150 nm SOI technology (AT77K).

This paper is organized as follows. In section II, the proposed solution architecture is discussed. In section III, the generic cell library is described. In section IV, the proposed approach is validated. Finally, in section V the conclusions are drawn.

II. Solution Architecture

The proposed automatic technology migration solution uses several different structures, involving different software tools to boost design reuse of projects across different technologies and different topologies in the same circuit class. The general overview of the project structure is represented in Figure 1.

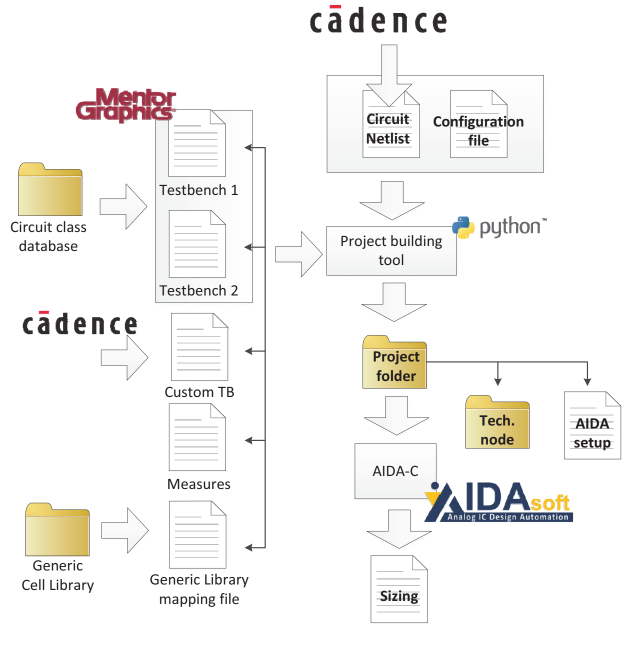

Figure 1 - General overview of the implemented tool.

The implemented tool is coupled to AIDA-C [5], an EDA tool for automatic analog IC sizing using multi-objective optimization techniques and ensuring circuit's robustness

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

through PVT corner considerations. The structures represented in Figure 1 can be summarized in the following parts:

- Generic Cell Library, a set of abstract symbols which can be mapped into specific technology devices, allowing the creation of technology-independent circuit schematics and netlists, which can be easily ported to a different technology;
- Circuit class database, a set of topology-independent testbenches to extract measures which characterize a specific circuit class;
- Project building tool, a tool which creates the project structure for AIDA-C, integrating the circuit class database, the generic cell library and the circuit netlist, based on the data fed by the configuration file;

A. *Circuit Class database*

In the proposed method, circuit netlists and simulation testbenches required for optimization and characterization of a circuit belonging to a specific class of circuits are kept in a database for future use. These are full custom circuit and testbench descriptions. The only constraint to reuse the available testbenches (which are topology independent) is having a matching subcircuit's name and port order. Additionally, project specific testbenches can be created at project level, and later added to the database, again to increase reuse and avoid unnecessary mistakes when redefining them for a new design.

B. *Design-flow*

The creation of a new project for an already setup topology starts with only two files: the circuit netlist and the configuration file.

The circuit netlist must be described using the devices in the generic cell library. This can be conveniently exported from the Cadence Virtuoso Schematic Editor (making use of the generic library symbols described in Section IV.a). Alternatively, some of the previously defined circuit topologies netlists already present in the circuit class database can be used.

The configuration file contains all the information required for the project building tool to create the project, such as the technology and the circuit class (so that the appropriate testbenches can be loaded) but also the topology-dependent restrictions (such as the devices' operating region). An example configuration file is shown in Figure 2.

```
TECHNOLOGY: xh035
CLASS: CComparator
CIRCUIT: ../CComparator/Netlist/chu.cir
MEASURES:
+SAT:{M0,  M23}  VOV=200m  DELTA=150m  You  can  add
different specs...
+SAT:{M24, M1} VOV=100m DELTA=50m
+TRI:{M5} DELTA=100m
```

Figure 2 – Example of a configuration file for the project build tool.

In this example, the specified netlist, will be tested using the predefined continuous comparator testbenches, in the XFAB 350 nm technology (XH035). Transistors M0 and M23 must be in saturation (overdrive larger than 200mV and saturation margin larger than 150mV), as well as M24 and M1 (although with different values for the overdrive lower bound, which is, in this case, 100mV, and for the saturation margin which musts be larger than 50mV). Transistor M5 must be in triode, with a margin larger than 100mV.

The Project build tool, implemented in Python, is then run. The target project structure is created: Testbenches are imported from the circuit class database, topology-dependent measure files are generated and the technology mapping files are added, based on the configuration file. Finally the optimization tool configuration is also automatically generated (based on the circuit class default structure and the devices' constraints in the configuration file). The designer should edit the generated configurations according to the project's demands as the design variable' ranges, corners, constraints, etc, are generated with default values. In the particular case of AIDA-C, the setup is described in an XML file. To set the project-specific values, the designer can be edit the XML file or use AIDA-C's Graphical User Interface. The circuit optimization can now be run to obtain the corresponding devices' sizes.

As the netlist was described using generic devices, porting the design to a different technology is achieved simply by changing the appropriate option in the configuration file. If a given topology does not meet the project demands, a new topology can easily be tried, since all the testbenches are topology independent and the project building tool automatically creates the topology dependent measure files.

III. Generic Cell Library

To prove the proposed method, a generic cell library with a limited number of devices was developed, together with its mapping to two target technology nodes: XFAB 350 nm ATMEL 150 nm SOI (AT77K). This generic cell library acts as a technology-abstraction layer to enhance design reuse. Circuit netlists are created using the generic devices, which are then wrapped to a specific technology for simulation. Thus allows using the same circuit netlist seamlessly for simulation in different technologies, facilitating technology porting.

A. *Generic Devices*

The generic library contains 10 abstract devices: one ELT NMOS, one regular NMOS, two PMOS, three poly resistors, two MIM capacitor and one vertical PNP bipolar transistor. Table I presents the available devices and the corresponding devices for the specific technologies. Not all devices have a match for both the considered technologies.

To allow creation of circuit schematics in the Cadence environment, each cell element also has symbol (shown in Figure 3) and eldoD views. CDF configurations guarantee that parameters are passed when the netlist is generated. Schematics can be created using the available symbols. When generating the netlist for simulation, for instance in ADE L, the appropriate mapping file must be included. The netlist can also be exported and included in the database.

TABLE I. GENERIC CELL LIBRARY DEVICES AND SPECIFIC COUNTERPARTS

Generic Name	Description	AT77K	XH035
nmos1v8	1.8V NMOS transistor	nmos	N/A
nmos3v3 elt	ELT 3.3V NMOS transistor	nfetox3_ring	nmos_elt
pmos1v8	1.8V PMOS transistor	pmos	N/A
pmos3v3	3.3V PMOS transistor	pfetox3	pmos
rpoly	Poly resistor	rplow	rp1
rpolyhr	Poly resistor	rphigh	rpp1
rpolyxr	Poly resistor	N/A	rhp1
mimcap	MIM capacitor	cmim34	cmm
dmimcap	Double MIM capacitor	N/A	cdmm
pnp_vertical	Vertical PNP transistor	pnp vert	qp4

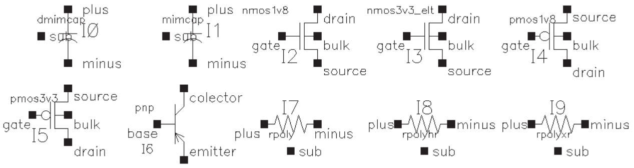

Figure 3 - Virtuoso symbols for generic devices

B. *Process corners and Monte Carlo analysis*

As variability and worst case models are of the utmost importance in analog IC design. This models must be also included in a generic library. While different foundries may provide different parameters sets, or corners, models, they can be grouped in meaningful generic corners as the physics behind the underlying model should be present in all processes. The implemented generic cell library supports 4 different process corners: worst power (wp), worst speed (ws), worst one (wo) and worst zero (wz). Monte Carlo (mc) analyzes are also supported, but not simultaneously with corner analyzes. Table II shows the matching of the library's corners with the technology specific corner options.

TABLE II. GENERIC CELL LIBRARY PROCESS CORNERS AND MONTE CARLO SPECIFIC COUNTERPARTS

Generic Name	XH035	AT77K
typ	tm	mos nom, rnom, cnom
wo	wo	mos fsc, rnom, cnom
wp	wp	mos bsc, rlow, clow
ws	ws	mos wcs, rhigh, chigh
wz	wz	mos sfc, rmom, cnom
mc	mc	mc + matching

C. *Technology mapping*

Technology mapping is the central procedure of the proposed method as it makes the interface between the generic, abstract devices and the specific target technology devices. For ease of use and integration with circuit simulators, the mapping is implement as a device library for the circuit simulators being considered. The library encodes the correspondence between the abstract and specific devices. The abstract devices in Table I are defined as sub-circuits. The heading of this sub-circuit definition (name and port order) is the same throughout different mapping files, allowing abstraction from the definition. Inside the sub-circuit, the correspondent device in the target technology is instantiated, connecting its terminals to the sub-circuits' and properly passing the input parameters.

The procedure is illustrated in Figure 4, where Figure 4 (a) shows the definition of the device pmos3v3 is for the AT77K process, and Figure 4 (b) shows the definition of the same device for the XH035 process. Creating the mapping for the generic cell library to a new technology just involves the creation of new mapping files. As each technology is different, this is a manual task to be done by the designer, where the devices in the desired technology which are closest to the generic description must be chosen and instantiated inside the generic name sub-circuit, as exemplified above. Special care must be taken with the input parameters. If the input parameters are different than the generic ones, a conversion must be made.

```
.SUBCKT pmos3v3 drain gate source bulk
   M1 drain gate source bulk pfetox3
              +L={(L) * 1u} W={(W) * 1u} M=Mul
.ENDS
```

(a) pmos3v3 mapped to AT77K

```
.SUBCKT pmos3v3 drain gate source bulk
   XM15 drain gate source bulk pmos
              +L={(L) * 1u} W={(W) * 1u} M=Mul
.ENDS
```

(b) pmos3v3 mapped to XH035

Figure 4 – Example of device mapping for two technologies.

The multiple corners and Monte Carlo models are supported as multiple libraries that are included in the same file, including the appropriate files from the technology's PDK.

D. *Library usage:*

The use of the generic library is straightforward, just the circuit's netlist with the generic devices. For example considering the 3.3V PMOS transistor from Table I, its instantiation with the drain connected to node VOUT, gate connected to node VIN and source and bulk connected to VCC!. might look like this:

```
XMP0 VOUT VIN VCC! VCC! pmos3v3 W=.6 L=.36 Mul=1
```

The W and L parameters are set in micrometers, i.e., length is 0.36 µm, width is 0.6 µm and multiplicity is 1.

For simulation, the generic library must be wrapped by a specific technology mapping. This is done by including the appropriate libraries from the mapping file, which maps the generic device names to actual devices. In the implemented generic libraries two libraries must be included: the devices library and one of the case libraries (typical, corner or Monte Carlo), but using a different arrangement would be trivial. For instance, if the following lines are present at the beginning of the testbench file, the simulation will be run in the specified technology (Atmel AT77K), for the worst-one corner.

```
.LIB 'at77k.lib' wo
.LIB 'at77k.lib' devices
```

Since these lines are added automatically by the project build tool, and the netlist is described with generic elements, provided that the at77k.lib mapping file is available in the same directory, which the project build tool also ensure. The circuit can easily be simulated for several technologies by simply changing the library includes at the beginning of the testbench file.

IV. Examples

To demonstrate the effectiveness of the approach a database was developed for continuous-time rail-to-rail comparators following the proposed design flow and using the previously described generic library and project build tool. As a first example, the tool was used to size a Chu comparator [9] in the XH035 nm technology. Since the comparator does not operate linearly, no restrictions were made regarding the transistors' operating region in the project configuration file.

The design target was set as the minimization of both propagation delay and current consumption. Corner validation was done, as an example, for 6 corners combining PVT variation and different common-mode voltage values (to guarantee rail-to-rail operation). A larger number of corners can be specified by the designer. Transistors' width, length and multiplicity are the optimization variables. Random offset requires Monte Carlo simulations, which are too computationally demanding to be done during optimization, and is thus characterized post-optimization. AIDA-C's optimization project, generated by the build tool, was run. Table III shows the typical and worst case measures for the considered corners, in one of the solutions.

To demonstrate the technology migration advantages, the same circuit topology was optimized for the ATMEL 150 nm SOI technology (AT77K). This was achieved only by changing the target technology line in the configuration file and re-rerun the project build tool and the optimizer. Typical and worst case measures for one of the solutions are also shown in Table III.

Finally, to show a topology change, a project was created to optimize a Park [10] topology in the AT77K technology, demanding only a line change in the configuration file and some manual adjustments to the constraints. The performance of one obtained solution are shown in Table IV.

TABLE III. Chu comparator simulation in XH035 and AT77K

Measure	Const.	XH035		AT77K		Unit
		Typ.	Worst	Typ.	Worst	
Current consumption		94	108	69	75	µA
Average delay		0.64	3.74	0.55	3.03	ns
Up prop. delay (large step input)	≤ 10	0.79	6.13	0.83	5.69	ns
Down prop. delay (large step input)	≤ 10	0.50	6.80	0.28	4.89	ns
Up prop. delay (200 µV step input)	≤ 100	18.9	67.0	36.1	72.9	ns
Down prop. delay (200 µV step input)	≤ 100	23.1	76.3	37.3	85.9	ns
Settling time	≤ 500	47.3	344	109	118	ns
Systematic Offset	≤ 1000	27.5	382	77.8	876	µV
Sensitivity	≤ 1000	280	344	215	255	µV

V. Conclusions

In this work a solution to enhance the analog IC design automation of technology migrations and increase design reuse was presented. A systematic design flow supported by a generic device library and a project building tool that automatically generates that required setup for a state-of-the-art multi-objective multi-constraint circuit-level optimization tool were shown to greatly increase design productivity in retargeting for new technology and adding new topologies for already defined circuit classes, further closing the design productivity gap. Continuous comparators were used as a case-study example, showing the effectiveness of the approach for technology migration and design reuse in the sizing stage of the analog project, while AIDA-C's multi-objective multi-constraint circuit-level optimization achieves good performance results.

TABLE IV. Park comparator simulation in AT77K

Measure	Const.	Typical	Worst	Unit
Current consumption		146	371	µA
Average delay		0.25	0,46	ns
Up prop. delay (large step input)	≤ 10	0.29	0.55	ns
Down prop. delay (large step input)	≤ 10	0.20	0.44	ns
Up prop. delay (200 µV step input)	≤ 300	55.5	282	ns
Down prop. delay (200 µV step input)	≤ 300	139	273	ns
Settling time	≤ 500	0.61	0.90	ns
Systematic Offset	≤ 10	1.22	9.10	mV
Sensitivity	≤ 1000	203	246	µV

References

[1] M. F. Barros, J. M. Guilherme, and N. C. Horta, "Analog circuits and systems optimization based on evolutionary computation techniques", Springer, 2010.

[2] J. Scheible and J. Lienig, "Automation of analog ic layout: Challenges and solutions," in Proceedings of the 2015 Symposium on International Symposium on Physical Design, pp. 33–40, 2015.

[3] G. Gielen, E. Maricau, and P. De Wit, "Design automation towards reliable analog integrated circuits," in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 248-251, 2010

[4] G. Gielen, E. Maricau, and P De Wit, "Designing reliable analog circuits in an unreliable world" in Proceedings of the IEEE 2012 Custom Integrated Circuits Conference, pp- 1-4, 2012

[5] N. Lourenço, R, Martins, A. Canelas, R. Póvoa and N. Horta, "AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation" Integration, the VLSI Journal, 2016,

[6] N. Lourenco, R. Martins, N Horta (2015) Layout-aware sizing of analog ICs using floorplan & routing estimates for parasitic extraction. In: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, 9-13 March 2015

[7] H. Habal and H Graeb (2011) Constraint-Based Layout-Driven Sizing of Analog Circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30(8):1089-1102

[8] E. Afacan, S. Ay, F.V. Fernandez, G. Dundar, and F., Basckaya (2014) Model based hierarchical optimization strategies for analog design automation. In: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 24-28 March 2014

[9] W.-S. Chu and K. W. Current, "A cmos voltage comparator with rail-to-rail input-range," Analog Integrated Circuits and Signal Processing, vol. 19, no. 2, pp. 145–149, 1999.

[10] S. Park, E. Greeneich, and T. DeMassa, "Low-power transistor-string and new rail-to-rail comparator in a/d converter," in Circuits and Systems, 1999. 42nd Midwest Symposium on, vol. 1, pp. 194–197, IEEE, 1999.