DO WE NEED SO MANY CELLS FOR DIGITAL ASIC SYNTHESIS ?

J.L. Noullet, A. Ferreira-Noullet
Institut National des Sciences Appliquees, Toulouse, France

KEYWORDS : SYNTHESIS, ASIC, STANDARD-CELL, CELL LIBRARY, DESIGN OPTIMIZATION

Abstract: The study presented in this paper is focused on library development for digital circuit synthesis. An experimental study was made in order to evaluate the impact of the size of the library on the performance of the synthesis process. For these experiments, a collection of 8 libraries based on the same technology and the same standard cells was prepared, with sizes ranging from 218 cells downto 10 cells. A set of 5 circuits were used as benchmark material. The results of these benchmarks show that the concept of "Reduced ASIC Cell Library" is valid : the loss of performance induced by replacing a 218-cell library by a 17-cell library is not always detectable, and when it is, its typical value is less than 10%, compensated by the benefit of faster library development and potential library optimization.

INTRODUCTION

The digital IC design activity is presently dominated by the standard cell approach. The use of pre-defined libraries of standard cells allows fast and reliable design of ASICs, and is also widespread for the design of non-regular parts of standard ICs like microprocessors, DSP, etc... The first commercial standard cells were intended for "manual design", more precisely gate-level design entry. Now there is a general trend to prefer the use of synthesis software tools [1], which are able to generate automatically gate-level descriptions from behavioral descriptions written by the designer, in a standard language like VHDL or Verilog.
The standard cells libraries are commercial products, and a consequence of the competition between vendors is an inflation of the size of the libraries.
Being involved in the development of small libraries in the university environment, we proposed this small study to confirm that it makes sense to create "small libraries".

1. REFLEXIONS ABOUT LIBRARIES

When manual gate-level entry was usual, designers felt more comfortable with libraries offering a wide range of functions, and appreciated the efforts made by the library vendors to increase the size of the libraries.
For example, the designer who had a 8-input AND gate ready-made was happier than the one who had to build it by putting together smaller gates.
Now the designer has little interaction with the gates, and is more concerned with the performances of the synthesis programs.
If we look at the circuits created by a state-of-the-art synthesis program (Synopsys), we observe that it seems to prefer gates with no more than 4 inputs (figure 1).

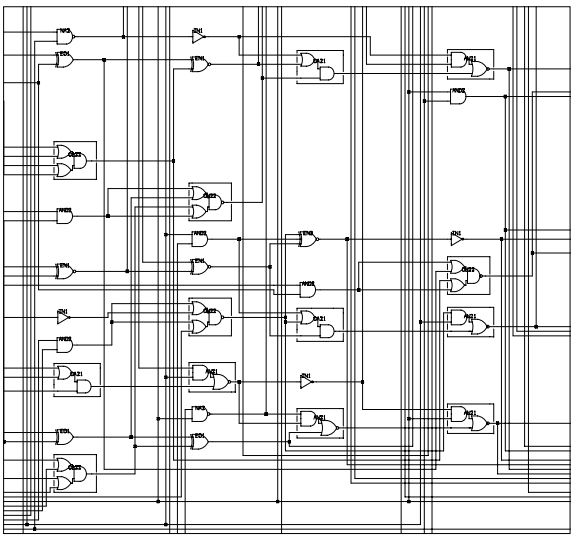

Figure 1

Another aspect of the evolution of the digital IC design is the domination of the synchronous design strategy, making obsolete asynchronous flip-flops.
The typical designer has to accept the libraries as they are proposed by the foundries, but in the university environment the situation is sometimes different.
For example, we developed for research purposes our own library on a Gallium Arsenide process [2]. Gallium Arsenide DCFL style allows only NOR gates [3], and this gave us the opportunity to verify that Synopsys is able to map any synchronous circuit on a library containing only an inverter, three NOR gates and a D-flipflop.
We had also a strong motivation to develop a CMOS library for educational purpose, because the commercial libraries do not let the user see the layout inside the cells. This confidentiality barrier is rather frustrating for the teacher and the students.
In the other hand, the frequent changes in technology make the libraries a very short term investment, and it is clear that the effort of re-designing some 250 cells every 2 years is not justified by the desire of making things more visible.

At this point, we can imagine some parallelism with the evolution of computers architecture. While microprocessor manufacturers were increasing every year the complexity of the instruction sets of their machines, research laboratories proposed the concept of "Reduced Instruction Set Computer" (RISC) as an alternative to "Complex Instruction Set Computer (CISC). The lack of sophisticated instructions in the RISC computers is compensated by a more efficient implementation of a few basic instructions.
We propose, at least under the form of a school exercise, the introduction of "Reduced ASIC Cell Library" (RACL).

2. EXPERIMENTAL STRATEGY

To investigate the RACL concept, we decided to prepare a collection of libraries of different sizes, based on the same technology, and to use them to synthesize a set of example circuits for comparison.

2.1 Libraries

We used as starting material a library supplied by a CMOS foundry, for a 0.6 micron process, and we made our experimental libraries just by building different subsets of this big library.
First, we observed that some of the 248 cells of this library were outside the scope of our study. After removing these cells (I/O pads, tri-state buffers), we obtained the "R0" library, the biggest of our collection (216 cells), the contents of which is summarized in table 1.
Then we remarked that the vendor made a huge effort by offering each cell with 3 different drive capabilities (buffered gates) in order to allow the synthesis program to drive conveniently the nets having a high fanout (a big capacitive load). But this can be achieved at nearly the same cost by the insertion of high drive inverters or buffers, this is why we made the second library by removing from R0 all the buffered cells except of course the inverters and buffers. Additionnaly, we removed the JK flip-flops and the constant logic level cells, which are not recommended in modern design. The result is R1, a library containing 120 cells.
After this we introduce the observation that the synthesis program makes very little use of combinatorial gates having more than 4 inputs (fig. 1). Moreover, we know that CMOS gates with more than 5 inputs are made by combination of several smaller gates, and our policy is to leave this task to the synthesis program itself.
Thus we continued by removing 42 combinatorial cells having more than 5 inputs. We also removed 12 flip-flops, 2 latches and 3 buffers that did not look essential, to obtain the R2 library containing 61 cells.

Table 1 : summary of the contents of the R0 library (216 cells)

Category	Examples	Number	Observation
single boolean level	nand, nor	42	(3 drive strengths)
single boolean level complemented	and, or	42	(3 drive strengths)
double boolean level	andnor, ornand	28	(up to 9 inputs)
double boolean level complemented	andor, orand	28	(up to 9 inputs)
exclusive or	xor, xnor	12	(3 drive strengths)
multiplexer	mux2, mux4, mux8	6	(2 drive strengths)
inverters	inv	5	(5 drive strengths)
buffers	buf	5	(5 drive strengths)
D flip-flops	dff	16	(async reset, set...)
multiplexed flip-flops	sdff	16	
JK flip-flops	jkff	8	
latches	dl	6	
logic constant generators	logic0, logic1	2	

At this point, we noticed that the vendor provided 24 cells composed of a single gate followed by a buit-in inverter (e.g. AND made of NAND + INV). This gives comfort to the hand-designer but is not essential for synthesis, since the synthesis program is supposed to find by itself where it is optimal to put inverters.

We obtained R21 (37 cells) by removing from R2 all these complemented cells.

In parallel, we made R22 (42 cells) also from R2, by removing all combinatorial cells having 5 inputs (cells with more inputs are already out). At the same time we removed the multiplexed flip-flops, also called scanpath flip-flops because of their role in scanpath test-specific circuitry.

We did not consider scanpath circuits in our study, but we observed that Synopsys sometimes uses these cells for optimizing circuits that are not concerned with scanpath.

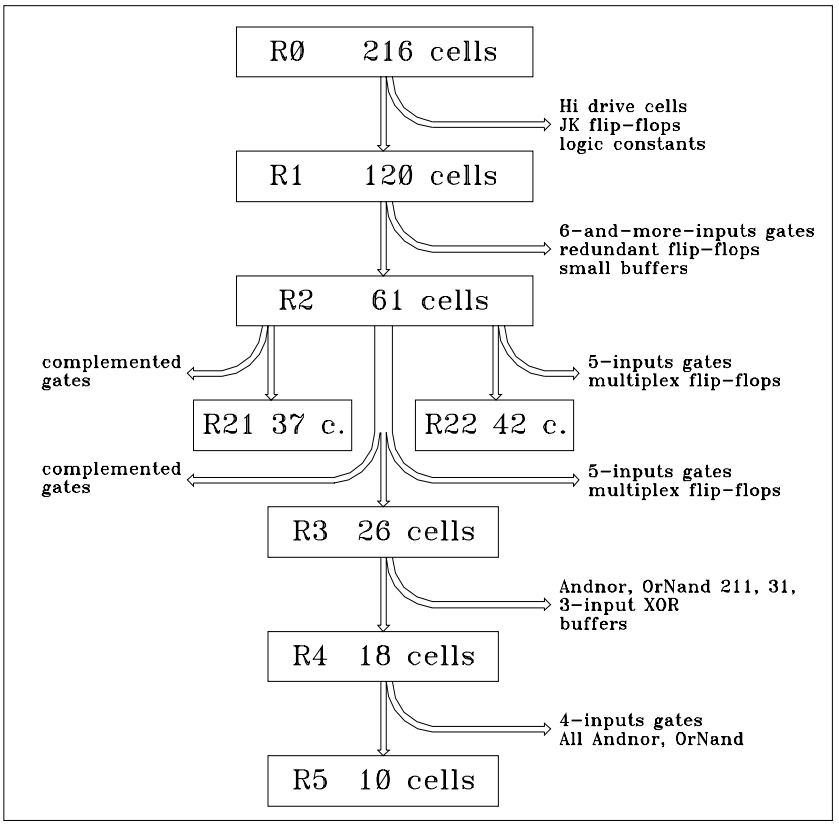

Figure 2

Library R3 is just the intersection of R21 and R22. It contains only 26 cells.
R4 was obtained by removing half of the remaining double boolean gates and all the remaining 2-stages buffers. Then it becomes possible to give a complete list of this library of 18 cells : table 2.

Table 2 : complete contents of the R4 library (18 cells)

single boolean level	nand2, nand3, nand4, nor2, nor3, nor4
double boolean level	andnor21, andnor22, ornand21, ornand22
exclusive or	xor2
multiplexer	mux2
inverters	inv (4 drive strengths)
D flip-flops	dff, dff_reset

To continue the process beyond exageration, we made R5 (10 cells) by removing the 4-input gates, the andnor, the ornand and 2 inverters.
Figure 2 gives a global vision of the substractive process used for this experiments.

2.2 Benchmark designs

We selected 5 circuits described in behavioral style (RTL) using the Verilog language.
The first (“d7seg”) is the “4 bits to 7 segments” decoder, a basic school exercise. The second (“hours”) is the hours counter, also a school exercise. The third (“cnt_ones”) gives a binary number representing the number of “ones” present in the input word. For the benchmark, it is parametrized to an input word length of 32 and an output word length of 5. This circuit is a spectacular example, since with only 4 lines of Verilog it can keep the synthesis program busy for hours ! These three circuits were described in [1].
The fourth is the complete interface module for a slave circuit on the “i2c” bus, and the fifth is a parallel 10 bits by 12 bits unsigned multiplier (22 bits output).
These circuits were synthesized with each of the 8 libraries with the following optimization goals :
- minimize area
- fault coverage 95% with priority to area
- structure logic with boolean optimization
- mapping effort : high

In addition, the multiplier was synthesized again with timing constraints :
- max path from any input to any output <= 10 ns
- structure logic with timing driven optimization.

3. RESULTS

A total of 48 optimization were made (8 libraries * 6 circuits, counting the multiplier twice), and the total areas reported by Synopsys are displayed in table 3. (Total area is the sum of the area of each cell and the interconnect area estimated for each net as a function of its fanout)
We expected the area to increase gradually as we reduce the size of the library. This is the general trend that is actually observed, but there are some exceptions : the best circuit is not always obtained with the biggest library. In terms of area, the best “hours” is obtained with R22, the best “cnt_ones” with R21, the best “i2c” with R1, the best “mul” with R22.

Table 3 :Total area after synthesis (arbitrary units)

Circuit	R0	R1	R2	R21	R22	R3	R4	R5
d7seg	14.68	14.68	14.68	15.70	14.68	15.70	15.70	17.07
hours	36.33	35.54	35.54	36.40	35.54	36.40	39.23	39.25
cnt_ones	221.45	221.45	221.15	205.20	222.78	207.64	206.17	216.81
i2c	245.38	229.93	231.84	232.41	232.70	234.34	239.11	262.93
mul	617.35	617.35	617.74	664.25	612.45	663.78	678.91	762.72
mul/timing	730.20	735.84	726.67	824.73	749.07	757.33	771.11	954.29

In fact, there is no evidence in table 3 that one of the 6 libraries from R0 to R3 is better than the others.
Moreover, we observe that the performance loss of R4 is always less than 10% compared to the best solution.
With only 18 cells, this library even beats R1 on some designs.
The drawbacks of cell number reduction appear clearly only with R5.

Table 4 : Critical path delay after synthesis (ns)

Circuit	R0	R1	R2	R21	R22	R3	R4	R5
d7seg	2.42	2.42	2.42	2.72	2.42	2.72	2.72	2.83
cnt_ones	25.81	25.81	25.82	25.82	25.56	24.99	24.46	24.48
mul	14.80	14.80	14.96	15.86	14.83	15.34	15.60	16.86
mul/timing	10.00	10.33	10.58	11.25	10.44	12.12	11.84	11.57

The critical path delays reported by Synopsys are displayed on table 4, only for the combinatorial designs. (expressing timing performances for sequential circuits is a bit more complicated)
We see that for the first three designs (synthesized with area constrained), Synopsys yielded very homogeneous timing performances, even if it was not explicitely instructed to do so. The conclusions drawn from table 3 are confirmed : little differences between libraries from R0 to R4 .
The cell number reduction produces more visible effect on the timing constrained design “mul/ timing”, where R4 suffers from a delay increase of about 18%.
We also tried to see if there is a significant variation of the CPU time consumed by the synthesis program when varying the library size, but the differences were too small to be of some interest.

CONCLUSION

This experimental study confirms, beyond the most optimistic predictions, that it is not essential to have a great number of cells in the libraries used for synthesis.
The R4 library (enumerated in table 2) with 18 cells is a good representative of “Reduced ASIC Cell Library” (RACL), it contains the essential cells for an efficient operation of the synthesis program.

The reduced libraries suffer from a loss of performance more visible on designs optimized for speed (timing driven optimization), but it must be recalled that in this study, the cells used for the reduced libraries were simply taken from the big libraries. We can imagine that if the same design effort had been concentrated on 18 cells instead of being spread on 216 cells, the cells would have been more optimized in terms of transistor sizing and layout compaction.

In the university environment, the RACL concept makes feasible the development of home-made libraries, with the benefit of transparency for teaching and research.
In the industrial context, the RACL context can help shortening the time-to-market of the new processes.

THE AUTHORS

Dr. Jean Louis Noullet is lecturer at the “Institut National des Sciences Appliquees” (INSA), Toulouse, France, and technical manager of the semiconductor process training laboratory at the “Atelier Interuniversitaire de Microelectronique”, Toulouse, France.
E-mail : noullet@aime.insa-tlse.fr

Dr. Ana Ferreira-Noullet was associate lecturer at the “Institut National des Sciences Appliquees” (INSA), Toulouse, France, and is now with Motorola Inc., Semiconductor Products Sector, in Toulouse, France.
E-mail : ana@aime.insa-tlse.fr

REFERENCES

[1] J.L. NOULLET, A. FERREIRA, P. BOURDEU D’AGUERRE, “Digital Synthesis : Guidelines for the Beginner”, Proc. of the 4th International Workshop on Mixed Design of Integrated Circuits and Systems MIXDES’97, Poznan (Poland), 12-14 June 1997.

[2] J.L. NOULLET, E. CRETE, D. LAGRANGE, J.Y. MAYVIAL, M. TORRES, A. FERREIRA,”Digital GaAs MESFET Chip for a Radio-Astronomy Spectrometer”, Proc. of the European Design and Test Conference EDTC’95, Paris (France), March 1995.

[3] J.L. NOULLET, A. FERREIRA, E. SICARD, “Digital GaAs Circuits in Practice”, Proc. of the 2nd Mixed Design of VLSI Circuits Conference, Krakow (Poland), May 1995.