Design of basic CMOS cell library

J.W.W. Chong
R.G. Forbes

Indexing terms: Integrated circuits, Simulation, Cell library design

Abstract: The paper was developed from an undergraduate student project carried out in 1989. The aims were to investigate the principles behind cell library design, and then to design a rudimentary cell library for use in connection with full-custom IC design teaching and projects. The library itself was targeted at the ES2 2 μm n-well process, and was built using the RACAL-REDAC 'ISIS' full-custom software suite, but the principles discussed should be of more general interest. The paper considers the general options available, the sequence of design decisions that need to be taken, and factors that may influence these decisions. Some unexpected and unresolved difficulties with simulators and simulation are reported, and some general conclusions about designing a cell library are noted.

1 Introduction

The paper is based around an undergraduate project [1] to design a rudimentary cell library for use in connection with full-custom IC design projects.

The project originated as follows. A difficulty encountered in the early stages of the UK Higher Education ECAD Initiative was that legal restrictions would prevent most users from seeing 'inside' the cells in the library provided under the initiative. Further, this library was in a 3 μm technology rapidly becoming obsolete, and was difficult to put into use for other reasons. The open literature is not very plentiful, and did not describe in detail how to set about designing a cell library. For all these reasons it seemed a good idea to try to design a small cell library of our own.

The objectives of the project thus became

(i) to find out what cells to include in such a library

(ii) to establish what design decisions were necessary

(iii) to design and layout a set of cells suitable for fabrication via the ES2 2-μm n-well process, using the RACAL-REDAC ISIS software suite.

[ISIS was at that stage one of the full-custom design suites available under the initiative, and the ES2 process was accessible to the Higher Education sector.]

Technology, of course, moves rapidly onwards, so details of cells designed two to three years ago would now be of limited interest. Instead, we concentrate on the principles of cell library design, as we now understand them.

Paper 8740G (E10), received 2nd January 1992

The authors are with the Department of Electronic and Electrical Engineering, University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom

Documentation for the complete cell library forms part of the project report [1]; the senior author [RGF] has a copy available for inspection.

2 Technical background

The prior generation of a library of reusable basic cells is a well tried methodology for saving time in the creation of large circuit designs. This philosophy can be applied at various levels, but our interest here is in low-level libraries that include cells corresponding to the basic logic gates and basic D-type elements. A library where all the cells are of a standard height makes the 'strip-farming' methodology of chip layout easy, and facilitates the application of automatic routing (or place-and-routing) techniques. The term 'standard cell library' often refers to such a library.

In many such libraries the cell layouts are hand-designed for a particular fabrication process. In consequence, the layout details may have significant commercial value; this, perhaps, is one reason for the paucity of information about cell-library design in the open literature. One of the few freely available descriptions is a book [2] specifying a 3 μm library, known as the CMOS3 library, designed at the instigation of the US National Security Agency (NSA). The library has had wide industrial and commercial use in the USA, and forms the basis of the MOSIS educational fabrication scheme there. Reference 2 was a useful source of ideas for the layout phase of our project.

Some libraries, including the CMOS3 library and the one we designed, have handcrafted layouts in which the width of each transistor can be chosen on the basis of some optimisation argument. An alternative methodology (see, for example, Reference 3) basically employs p-type and n-type transistors of fixed widths, arranged in parallel rows, with the ratio of widths chosen in accordance with some generalised optimisation argument. This methodology usually has a well defined intracell interconnection scheme, and connects transistors in parallel when optimisation requires large channel widths. The ES2 'optimised array' approach [4] is a methodology of this general type.

The design, optimisation and characterisation of a cell library is a manpower-intensive activity, that relates uneasily to a world where process technology develops rapidly, and minimum process linewidth gets reduced every two years or so. In response to this, attempts have been made to create libraries (or library generation facilities) which, to a greater or lesser extent, are technology and process independent. The NSA 'Scalable cell family', based on a 'lambda-rule' philosophy and described in Reference 5, is one example of this. It is basically a fixed-width-ratio methodology as described above. At the cost of reduced optimisation in timings

and/or silicon usage, the library is created and held in a form which is said to be readily 'scaled' as process linewidth reduces. Whether (for example) contact sizes will in fact scale in the same way as minimum channel length, and whether circuit capacitances will in fact scale uniformly, are debatable issues.

Another line of approach holds the layout topology in a 'sticks-based' form, that can be fleshed out using a process-specific information file. The resulting layout can then be compacted automatically. The 'CHIPWISE' design software [6] developed by Waller and colleagues at the University of Kent is basically of this type. We look again at such techniques in Section 7.

Some more sophisticated techniques of cell library generation, based on compilation, are described by Buurma *et al.* [7] and Martinez *et al.* [8].

3 Design issues

In this Section we look at specific issues involved in the design of our library. Some ideal requirements for a standard cell library might be

(*a*) comprehensiveness
(*b*) simplicity and ease of use
(*c*) minimal silicon utilisation
(*d*) good characterisation and full documentation
(*e*) near fully-handcrafted performance for full circuit designs using the library cells
(*f*) cells fully laid out in accordance with process rules
(*g*) technology and process independence
(*h*) easy to maintain and update

Some of these requirements, particularly those in the second half of the list, are incompatible. As already indicated, the general philosophy chosen for our library was a conventional one: cells of standard height which are suitable for strip-farming, internal power buses running parallel to the rows, signal inputs and outputs taken to both top and bottom edges of each cell and transistors able to be designed individually. This philsosophy is the same as that of the cell library originally provided under the ECAD initiative, and that of the NSA CMOS3 library, which both influenced our initial thinking.

Working within this philosophy, this Section now describes the factors involved in the choice of the set of cells to be designed, cell pin pitch, cell height, width of cell power bus, well sizes and transistor sizes.

3.1 Which cells to include?

The balance in principle here is between comprehensiveness, enabling the designer to find just the function he or she requires in the library, and the costs associated with library creation and maintenance, cell characterisation, and limited lifetime due technology migration.

Our conclusion was that a small library should contain the following cells:

* 2,3,4 input NAND-gates	* 2-1 multiplexor
* 2,3,4 input NOR-gates	* transmission gate
* inverter (low-drive capacity)	exclusive-OR gate
* buffer (high-drive capacity)	full adder (1-bit)
resettable D-type latch	* feed cell
clock driver	book-end cell

This choice conforms with the approach of Sunter [9], whose philosophy resembles that behind the reduced instruction set computer. He statistically analysed cell usage in past designs (chiefly in telecommunications), and found that about 25 cells accounted for more than 99% of cell placements. Our set of cells is, with a few exceptions, a subset of his. In practice, only the cell designs marked * could be completed within the timescale of the project; but a resettable D-type latch was designed in the month following project submission.

3.2 Width of power and ground buses

The major constraint on minimum bus width is electromigration, which occurs when the current density exceeds a threshold value, about 1–2 $\mu A/\mu m^2$ [10]. Factors that need to be considered include the average load on each output node, the desired power-rail voltage and switching frequency, and the number of cells (fed from a single power bus) likely to be switching simultaneously. We concluded, partly from our estimations, and partly from the CMOS3 library's use of an 8 μm bus in 3 μm technology, that a power/ground bus width of 7 μm should be sufficient for a 2 μm technology.

These buses were placed 7 μm in from the top and bottom edges of the cell, to accommodate the substrate taps.

3.3 Cell pin pitch

Practical constraints on small pin pitch are set by the minimum separation of the centre lines of the gates of two connected transistors laid out side by side, and by the design-rule requirements concerning 'diagonally adjacent' contacts in the routing channel. In practice, it seems useful to have the pin pitch as a simple multiple of the linewidth; the multiplier in the 3 μm libraries we looked at was 4, so for our library we chose a pin pitch of 8 μm. This accommodates the practical constraints.

3.4 Cell height

As pointed out in Reference 9, the resettable D-type emerges as a vital cell in a typical sequential-logic semi-custom design, so it is the cell height of a well laid out D-type that will set the standard height of the library.

A study of 3 μm libraries (all based on a 12 μm pin pitch) showed cell heights varying between 130 and 150 μm. It seemed possible to lay out a relatively compact D-type transparent latch with a height of about 130 μm, so using a rough scaling argument we set our cell height to be 11 times the pin pitch, i.e. 88 μm for the 2 μm process. No particular difficulty resulted from this choice, but possibly 96 μm would have been better. Sunter [9], on the basis of experience rather than theory, used a factor of 10 in his 1.5 μm technology.

3.5 W_p/W_n and height of n-well

The ES2 process parameters give the ratio μ_n/μ_p of carrier mobilities as 510/175. The usual arguments (e.g. Reference 10), assuming equal channel lengths $L_n = L_p$, thus imply that optimum noise margins and symmetrical timings are achieved (for an inverter) with a channel-width ratio W_p/W_n of about 3 : 1.

A further consideration, however, is proposed by Sung Mo Kang [11], who argues that a figure of merit for a cell library should be derived from the product of cell area and delay time. Making various reasonable but slightly arbitrary assumptions, he shows that this figure of merit is a concave function of W_p/W_n that has a minimum at W_p/W_n slightly greater than 1. As a compromise between the two conflicting results, he suggests a W_p/W_n ratio of 2.

For multiple-input gates, the optimised relative widths of p-transistors and n-transistors will vary from the inverter result, but in different directions for NOR and NAND gates.

In broad terms, these results indicate that the space in the cell available for p-transistors should be somewhat greater that that for n-transistors. We found it difficult to analyse this in a satisfactory quantitative manner, and the simulation inconsistencies described in the following Section made the situation worse. It was eventually decided to have the internal n-well boundary exactly halfway down the cell side. This in fact gives slightly more space for p-transistors because the p^+-diffusion regions may come within 3 μm of the n-well edge but the n^+-diffusion regions may come only within 7 μm.

No particular difficulty resulted from this choice, given that we used the HSPICE simulation results described below in the optimisation of transistor widths, but possibly a larger cell height and a larger n-well would have been better. (However, if the simulation results had been closer to those of the usual elementary theory, then the situation might have been different, with a somewhat greater proportion of the cell being needed for the p-transistors.)

4 Simulators and simulation

4.1 Background considerations

Much of the project time was in fact taken up with establishing what methodology to use for the simulations. The difficulties involved deserve detailed mention, because they are a general consequence of the ISIS approach.

The ISIS software suite has its own sophisticated simulator, 'HYLAS', based on the FOSS transistor model. For circuit-level simulation, this gives rise to a peculiar set of difficulties, because for accurate results it is necessary to use the parameters relevant to the targeted process. However, fabrication houses (and ES2 in particular) tend to provide only SPICE parameters.

In principle, FOSS parameters could be derived from appropriate experiments, but there was no available resource to do this. So, prior to this project, two other routes had been explored.* First, they tried to simulate, using SPICE, the experiments that would be involved in deriving the FOSS parameters. A problem here is that the experiments represent extreme cases and it is unclear how accurate SPICE is in these circumstances. Secondly, use was made of an optimisation program written at the Rutherford Appleton Laboratory which converts SPICE parameters to FOSS parameters. The problem here is that the parameters so derived do not necessarily have a good physical meaning, so the reliability of the model away from the region of phase space where the optimisation was performed is questionable.

We thus did have FOSS-model parameters for the ES2 process, but they were known not to be fully reliable. The university also has a 'level 2' SPICE running on a PRIME machine, and at the time of this project had just acquired the SUN-based HSPICE available under the ECAD initiative. The latter can be used in a 'level 2' mode by inserting the parameters as used in the PRIME SPICE, and setting other HSPICE parameters appropriately. The export of data from the VMS-supported ISIS system is not a problem because there is a utility in ISIS that converts HYLAS circuit descriptions (and some other information) to SPICE format files, and these can be transferred over the departmental network to the UNIX-based SUN server.

In preliminary experiments, we simulated an inverter using HYLAS, PRIME SPICE, and HSPICE (level 2). Discrepancies in timing between HYLAS and the SPICE programs ranged up to a factor of about 2. At this point we deemed the SPICE results more reliable because the SPICE parameters were provided by ES2, whereas the FOSS-model parameters were derived indirectly.

There were discrepancies of between zero and 30% between HSPICE and the standard SPICE on the PRIME. We also found, as expected, that HSPICE was faster, and had much less in the way of convergence difficulties. Because of this, and because of the electronic data transfer route between HYLAS and HSPICE, we standardised on the latter.

There are many different levels of model in HSPICE, and ES2 provide data for levels 2 and 6. Initial investigations showed that the level 2 and level 6 models did not produce the same results, but lack of time prevented follow-up. Thus the simulations in this project were based on HSPICE level 2.

4.2 Simulation and transistor sizes

As already indicated, a basic issue when constructing a cell library using our philosophy is how to choose the widths of the transistors, and in particular the ratio of p-transistor to n-transistor widths. We chose to design for approximately equal rise and fall times, based on a 10% to 90% criterion, assuming a 0.5 pF load and a 10 ns input ramp.

In retrospect, because output rise and fall times in general turn out to be somewhat less than 10 ns, it might have been better to have assumed a somewhat sharper standard input ramp (say 2 ns), and/or to have made some investigation into the effects of the sharpness of the input ramp. However, this seemed a relatively unimportant matter at the time.

A puzzling feature of the simulations, which is not understood, was that HYLAS and HSPICE gave rather different results for optimum device ratios. Thus, for an inverter with $W_n = 10$ μm, HYLAS gave an optimum ratio W_p/W_n of about 3, in accordance with elementary theory, whereas HSPICE gave an optimum ratio of about 1.4. Checking of data entry did not disclose any obvious mistakes.

In the context of a research project, an apparent anomaly like this would obviously need to be properly investigated. Within the context of the tight timescale of an undergraduate project, an awkward and arbitrary decision has to be made, and we chose to use HSPICE and the parameters directly provided by ES2, notwithstanding the apparent clash with elementary theory.

For design purposes, simulations were then carried out for the various gates to be included in the library, these resulting in appropriate sets of 'ideal' transistor widths. In the layout stage some small compromises were necessary, particularly with the three and four-input gates, to fit the transistors within the cell sizes chosen, and some re-simulations were necessary. After layout was completed, area and periphery capacitances for the sources and drains of transistors were extracted manually and where appropriate were fed back into the HSPICE calculations.

As preparation for the documentation phase, final post-layout simulations were carried out, using these extracted capacitances where appropriate and loads

* PATEL and FORBES, Surrey (unpublished work).

varying from zero up to 1.5 pF. Results were obtained for rise and fall times, and for the rising and falling propagation delays (defined by reference to 50% voltage levels at input and output). A small facility program, written in C† was used to extract the timing data from the SPICE outputs.

5 Layout

As already indicated, cells were laid out using the ISIS software suite, which has interactive design-rule checking, and post-layout circuit-topology and DRC checks. The procedures are fairly standard and need no description here, but some general points deserve notice. First, meandering transistors were employed where possible, to maximise use of the available space in each cell. Secondly, multiple contacts (particularly to drain and source diffusions) were used when possible, to reduce contact resistance and enhance reliability. Thirdly, proper attention needs to be paid to potential cross-boundary design-rule violations.

6 Machine-readable output and documentation

A main part of the project output was, obviously, in the form of computer files. For each cell these provide a hardware description language (ISIS-HDL) description of the circuit, a spice-format description of the circuit and fully-validated files describing the silicon layout. Files describing the corresponding logic symbols were not produced because these were already available on the system as a library facility.

Production of good-quality paper documentation was regarded as important. For each cell there was produced:

(*a*) a summary sheet showing

(i) outline of cell, with size and position of bristle points indicated,

(ii) conventional logical symbol

(iii) circuit schematic with transistors labelled and their widths shown

(*b*) a plot of timing data, showing how various characteristic times vary as a function of load, derived by simulation as described in Section 4

(*c*) a diagram in black and white of the cell layout. This was extracted automatically from the cell layout files, using a utility program written by B.M. Cook. The resulting file is transferred electronically from the VAX to the departmental UNIX system and used to drive a laser printer.

As an illustration, the documentation produced for the 3-input NOR-gate is shown in the Appendix.

7 General comments

The work described above can be viewed both as a student project, and as a more serious attempt to design a cell library (actually, to find out how to do so).

As a student project it proved very successful (and in fact emerged as the top project of the year). Other experience, however, has shown that this type of project is not entirely suitable for weak students or for those who are poor at coping with engineering uncertainties. The main drawback of the project was that we did not expect as many problems with the simulation as in fact were found. Consequently, the time needed for simulation at the design stage was underestimated, and it was impossible to design all the cells originally planned. In practice, we achieved only the main combinational cells (marked with an asterisk in the list in Section 3.1) during the timescale of the original project. Some latch cells were added later, in a separate exercise.

As must be evident from this paper, the project has also been useful in its role as a cell-library design study, in that we now have a much better idea of the factors to be taken into account, and the steps that need to be taken, when designing a cell library. However, a major conclusion here is that there are apparent anomalies in the behaviour of the simulators available to us, that deserve more detailed investigation.

There are also some general conclusions to be drawn, particularly for libraries designed in the philosophy we used. First, it seems strongly advisable to use a simulator for which the chosen silicon foundry will provide process parameters derived directly from measurements. In practice, in the UK Higher Education ECAD Initiative, this means parameters for SPICE or a variant. Secondly, there are some consequences of the current relentless drive to smaller linewidths. Any fully-designed cell library has a short half-life, probably at most a few years. So there is merit in using a RISC-like philosophy and restricting it to a relatively small number of basic cells. However, in practice, keeping even a small cell library up to date probably involves more manpower than is easily available in most university departments.

Obviously, our library has involved handcrafted layout. For the more general job of keeping a cell library up to date, it might be argued that we should use a system that holds the layout topology in a technology-independent form, as described in Section 2. That can, in principle, be fleshed out using a file customised to a particular set of design rules, and compacted. Clearly, there is the issue of whether a compacted layout can be as efficient in terms of silicon usage as a handcrafted one, particularly when meandering transistors can be handcrafted. However, more to the point is our experience that, in terms of time taken, handcrafting the layout was a relatively small part of the design process: simulation and the preparation of documentation took a far greater proportion of the time. Consequently, we suspect that, if reasonably optimised cell libraries are required, then the use of systems based on technology-independent layout techniques may not in practice have quite the advantages that they might seem to possess at first sight.

All in all, much was learnt from this project by both authors.

† PERKINS, A. (unpublished work)

8 References

1 CHONG, J.W.W.: 'Design of an ISIS cell library'. Final-year project report, Dept. of Electronic and Electrical Engineering, University of Surrey, 1989

2 HEINBUCH, D.V. (Ed.): 'CMOS3 cell library' (Addison Wesley, Reading, Mass., 1988).

3 STEBNISKY, M.W., FELLER, A., SMITH, A.M., BORGINI, F., SHARMA, S.S., and McGINNIS, M.J.: 'Approaches and tradeoffs in optimal standard cell design'. Proc. IEEE 1983 Custom Integrated Circuits Conf., May 1983, pp. 104–109

4 Semi-custom methodology used by European Silicon Structures, Bracknell, UK

5 NEWTON, R.C., and LANTZ, M.D.: 'A high performance, multi-sourcable, scalable CMOS cell family'. Proc. IEEE 1986 Custom Integrated Circuits Conf., May 1986, pp. 2–5

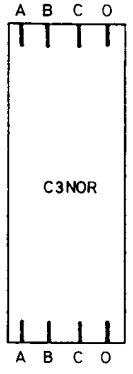
6 WAKROWSKI, L.T., WALLER, W.A.J., and WELLS, E.: 'Teaching VLSI Design using Chipwise', *Int. J. Elec. Eng. Educ.*, 1991, **28**, pp. 271–277

7 BUURMA, G., MICHEL, P., and KURODA, T.: 'A high performance scalable standard cell library with true second sourcing'. Proc. IEEE 1987 Custom Integrated Circuits Conf., May 1987, pp. 241–243
8 MARTINEZ, A.M., DHOLAKIA, S., and BUSH, S.: 'Compilation of standard cell libraries', *IEEE J. Solid-State Circuits*, 1987, **SC-22**, pp. 190–197
9 SUNTER, S.: 'Designing a CMOS standard cell library'. Proc. IEEE 1987 Custom Integrated Circuits Conf., May 1987, pp. 237–240
10 WESTE, N., and ESHRAGHIAN, K.: 'Principles of CMOS VLSI design' (Addison Wesley, Reading, Mass., 1985)
11 SUNG MO KANG: 'A design of CMOS polycells for LSI circuits', *IEEE Trans.*, 1981, **CAS-28**, pp. 838–843

9 Appendix: Library documentation for 3-input NOR-gate

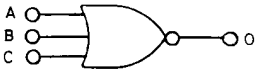
Three pages from the original project report [1] are reproduced in this Appendix. They show the information provided in the report in respect of a 3-input NOR-gate. These pages illustrate the format of the paper documentation for the cell library.

cell input/output



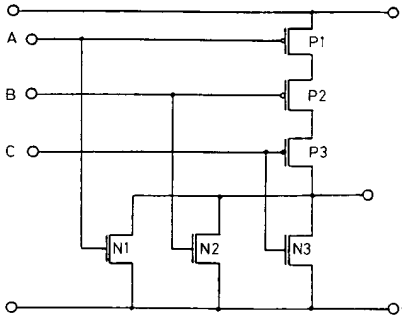
cell width 32 μm

logic symbol



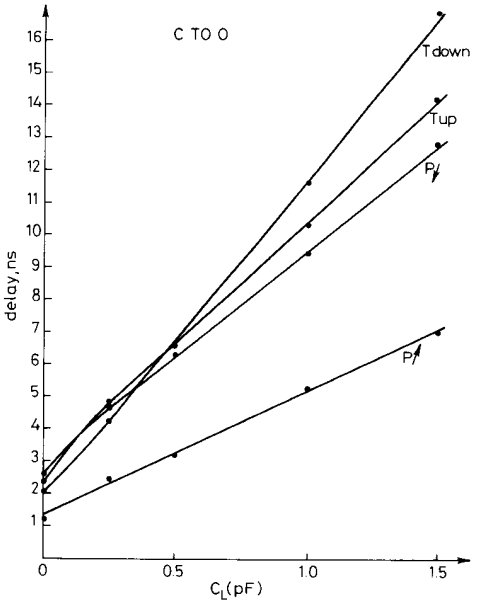
logic function $O = \overline{A + B + C}$

circuit schematic



	W	L
P1	33	2
P2	33	2
P3	33	2
N1	6.5	2
N2	6.5	2
N3	6.5	2

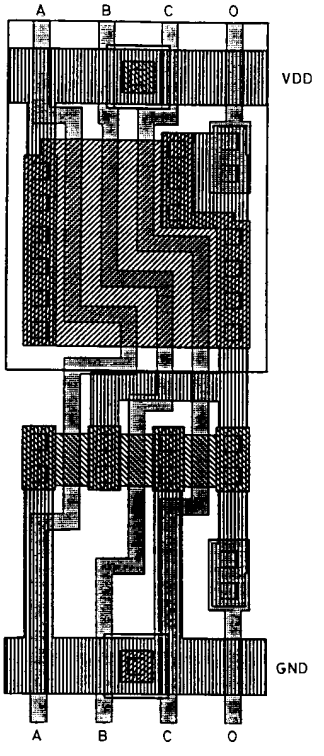
Timing information



Simulation conditions

V_{dd} = 5 Volts
temp = 27 degrees Celcius
input rise-time = 10 ns
simulator used: HSPICE H8807a

cell layout



C3 NOR at 1500x