

UNIwersytet Technologiczno-Przyrodniczy
Im. Jana I Jędrzeja Śniadeckich w Bydgoszczy
Zeszyty Naukowe Nr 256
Telekomunikacja i Elektronika 13 (2010), 5-14

AUTOMATED DCT LAYOUT GENERATION USING AMPLE LANGUAGE

Andrzej Handkiewicz, Paweł Śniatała, Grzegorz Pałaszyński, Szymon Szczęsny,
Piotr Katarzyński, Michał Melosik, Mariusz Naumowicz

Poznań University of Technology
Chair of Computer Engineering
Piotrowo 3A, 60-965 Poznań, Poland
Szymon.Szczesny@put.poznan.pl

Summary: Designing SI circuits layouts is a demanding task. The process is very time consuming and there is a high risk of making mistakes. It would be much easier if there were a CAD tool doing part of the job for ourselves. This is the place where a possible solution comes in – the AMPLE script language in the ICStation environment. AMPLE is a script language that can be used to generate layouts. Apart from making a layout faster the AMPLE generator enables parametrisation of SI devices and can also be technology-independent. It provides a way for automating and speeding up the process of designing a layout. This paper presents a DCT layout generator which takes advantage of the AMPLE language and offers parametrisation that can make the design process independent from the technology used.

Keywords: AMPLE, layout, discrete cosine transform, current mirror, analog circuit

1. INTRODUCTION

Digital circuits become more and more popular nowadays. There are a lot of CAD tools that help designers in automating most parts of the design flow. However analog circuits are still very important and in some situations irreplaceable. They are, for example, widely used in image processing and video compression. Unfortunately CAD support is unsatisfying and insufficient in this matter, therefore the problem with the automatic generation of an analog layout is currently a very popular topic [1], [2], [3].

Moreover, analog circuits layout generation also requires automation and assistance of software tools which could handle the operations of generating a standard analog cell which would depend on input parameters. For instance, for a current mirror an input parameter could be the current gain, while the size of the coefficient matrix might be a parameter for a DCT implementation in the CMOS technology.

Authors of the article took into consideration all those demands for CAD tools and proposed the following solution – automated generation of a DCT layout using the AMPLE script language – described in the following chapters. The proposed method shows how to reduce total time of designing analog circuits layouts, also the way of parametrising the designing process and shows how to prevent making mistakes.

2. DISCRETE COSINE TRANSFORM

This article uses the concept of the two-dimensional Discrete Cosine Transform (DCT), therefore in this section we shall review some basic information about it.

Using the Discrete Cosine Transformation implementations is very important in the process of compressing images. JPEG and MPEG are only two, most popular examples of many standards which use DCT. It has been observed that the two-dimensional (2-D) Discrete Cosine Transform has the output similar to the one produced by the Karhunen-Loeve transformation [4] and uses image-independent basis functions. This is the reason why DCT-based image coding is applied to all video compression standards [11]. In these standards, the image is divided into 8x8 blocks in the spatial domain and DCT transforms them into 8x8 blocks in the 2-D frequency domain. Such a block size is convenient in respect of computational complexity. Larger sizes are also possible, however they do not offer any significant improvement concerning the level of compression.

2-D DCT in the size of 8x8 can be expressed as [5]:

$$y_{k,l} = \frac{c(k)c(l)}{4} \sum_{i=0}^7 \sum_{j=0}^7 x_{ij} \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, \quad (1)$$

Where $k, l = 0, 1, \dots, 7$ and

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0, \\ 1, & k \neq 0. \end{cases} \quad (2)$$

Assuming that the matrices Y and X are composed of elements y_{ij} and x_{ij} , where $i, j = 0, 1, \dots, 7$ respectively, the relation (1) can be also written in the matrix form as:

$$Y = CXC^T \quad (3)$$

and the matrix of coefficients C is as in (4), where:

$$\begin{aligned} a &= \cos(\pi/16); \\ b &= \cos(2\pi/16); \\ c &= \cos(3\pi/16); \\ d &= \cos(4\pi/16); \\ e &= \cos(5\pi/16); \\ f &= \cos(6\pi/16); \\ g &= \cos(7\pi/16); \end{aligned}$$

The main property of 2-D DCT, with respect to implementation, is separability. On the basis of the matrix equation (3), written in the form (5) we can realize 2-D DCT with two 1-D ones. The matrix X denotes one input 8x8 block, and its transposition X^T in the relation $Z = X^T C^T$ means that it is read out column by column. The matrix Z, containing intermediate results, is obtained with the use of 1-D DCT, and can be saved in a memory array.

$$C = \frac{1}{2} \begin{bmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{bmatrix}, \quad (4)$$

$$Y = Z^T C^T, Z = X^T C^T \quad (5)$$

As we can see, the matrix of coefficients (4) only depends on the size of the 2-D Discrete Cosine Transformation. Therefore, it is possible to parametrise the algorithm for designing a circuit which could calculate the DCT. This article proposes the idea of a DCT layout generator based on an AMPLE script. The next part briefly describes the AMPLE language.

3. WHAT IS AMPLE?

AMPLE is an abbreviation for Advanced Multi-Purpose Language which is a standard used in the Mentor Graphics Common User Interface [7], [8] common to all Falcon Framework-based applications. End-users can use AMPLE to modify and extend Falcon Framework-based software functionality by evaluating expressions, creating and assigning variables, defining and executing new functions and directly executing built-in functions. AMPLE supports the C library and module dynamic linking to the existing in-house, as well as third-party solutions bound to the Falcon Framework. Moreover, it supports popular statements like, for instance: iterations, multi-conditional branching, loops, loop's interrupts and terminations.

AMPLE and the Common User Interface contain constructors for defining custom menus, prompt bars, forms, function keys and strokes, as well as user's custom functions and commands. Implemented scripts can be included as an integral part of the general design flow for automating the design of SI circuits. Due to these reasons AMPLE is perfect for our layout customisation.

4. DCT LAYOUT GENERATOR

The idea of writing a layout generator is to design a schematic, create a layout using Schematic Driven Layout (SDL) and finally to analyse this layout in order to create a generator which could take input parameters. All these steps are covered in the sections below.

4.1. SCHEMATIC

Current is the signal that drives information in the SI technology [5]. Therefore, multiplication operations (scaling signals) from (1) can be realised using current mirrors.

Addition operations are realised in the nodes according to the Kirchoff's first rule – the Kirchoff's current law. The schematic of an 8x8 DCT which is shown in Figure 1 has been chosen for the implementation.

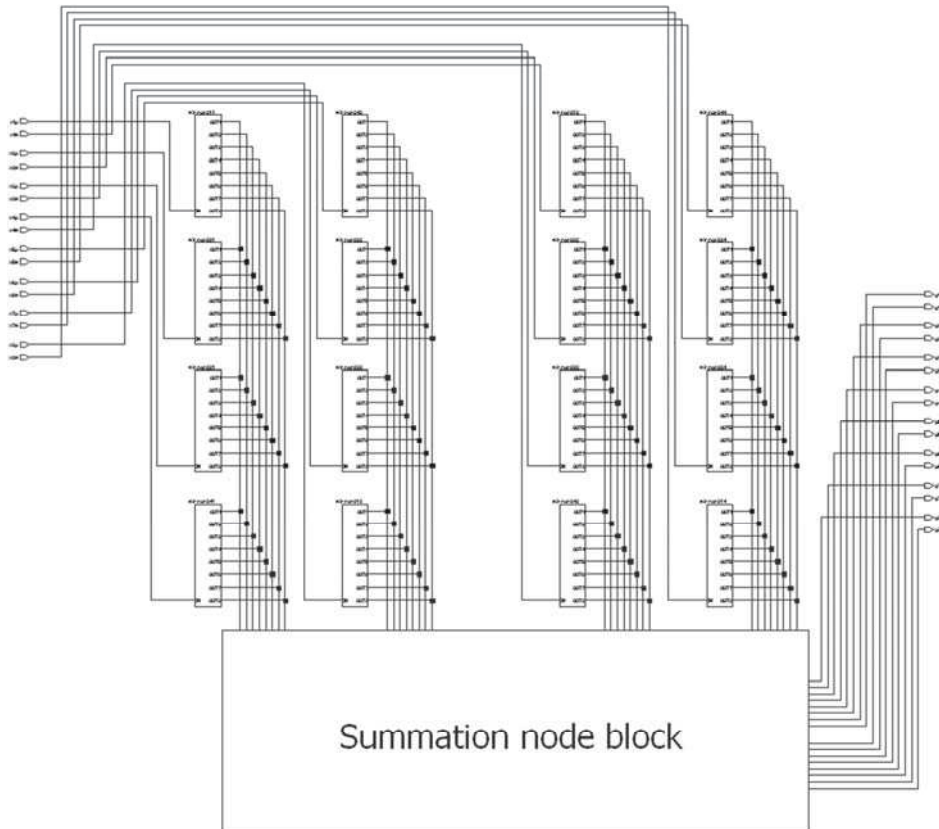


Fig. 1. Schematic of the DCT circuit

The schematic consists of many elements built in the same way, therefore it was constructed hierarchically. Basic cells of the hierarchy are the output stages of current mirrors (in fact they are inverters) with different NMOS and PMOS lengths but the same widths. One of such stages is shown in Fig. 2. The 8x8 DCT includes 8 current mirrors (with one input and eight outputs – Fig. 3) and each one, as an output stage, includes a cell shown in Figure 2. Connections in the block of summation nodes depend on the signs of the coefficients in the matrix C^T . Because of balanced structure, the top-level schematic of 1-D DCT consists of sixteen 1-input, 8-output current mirrors.

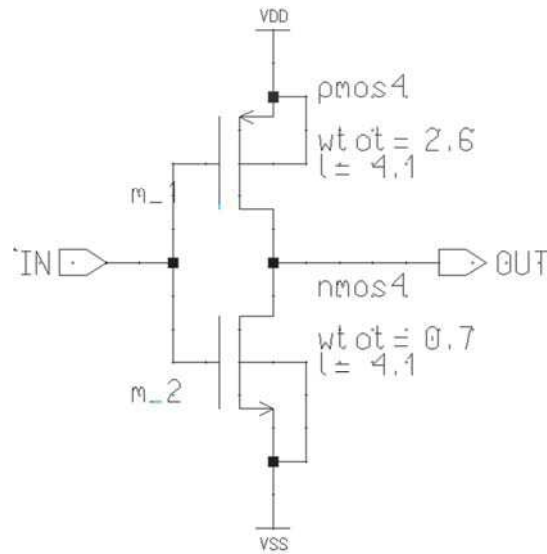


Fig. 2. Output stage of current mirror

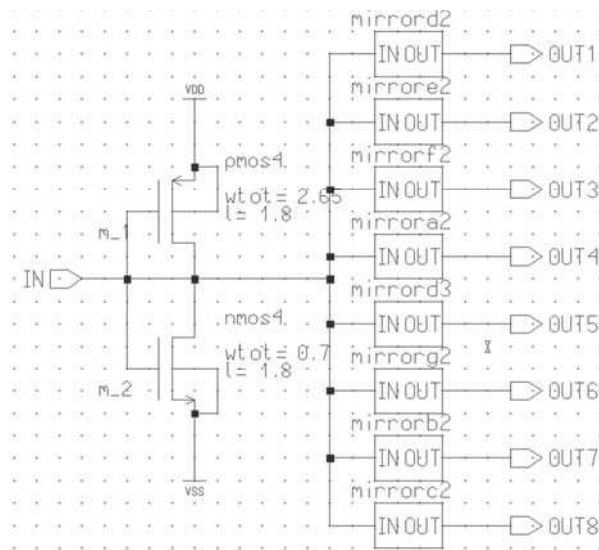


Fig. 3. 8-output current mirror

2-D DCT can be realized in two ways: with and without a memory array. Two circuits shown in Figure 1 are necessary to realize 2-D DCT if the memory array for intermediate results Z in (5) is used [5] and sixteen circuits are necessary for the realization without the memory array [6].

4.2. SCHEMATIC DRIVEN LAYOUT

Providing that we have a schematic we are able to use the ICStation environment to build a layout for this particular circuit [9], [10]. Some features, like the Schematic Driven Layout (SDL, mentioned above) are very helpful in the custom VLSI design. We can also take advantage of tools such as auto-routing or auto-placement. With the support of such software we are able to create a layout, however and unfortunately only for this particular schematic.

If we decide to change the number of inputs and outputs, we have to create a new schematic and repeat the whole process from the very beginning. Moreover, if we change the technology to a newer one, for example: where certain distances between elements can be smaller, we need to re-create the layout once again. Therefore, this is what we demand from our AMPLE script: parametrisation of the DCT layout.

4.3. AMPLE SCRIPT

Our generator was written after taking into consideration the SDL result. Because of the layout's complexity the script was divided into sections and functions responsible for elementary tasks.

The beginning of the script contains declarations of the input data (the size of the current mirror), as well as parameters setting up, for example: distances between elements. This is very useful because thanks to it we can read design rules from a special file, thus making the script technology-independent. In such a situation, the script would open this special file and, taking into consideration the rules, it would minimise the area necessary to make a layout.

The second part of the AMPLE generator includes calculations which check, depending on the values of parameters, for example: dimensions and position of the specified current mirror's layout. Heights and widths of inverters are calculated taking into consideration the sizes of power supply ports and distances between ports and transistors. After the above steps are completed, it is possible to generate the layout of our DCT.

Initially, as a result of a function, a simple input stage of the current mirror is obtained. The previously specified argument defines the length of the NMOS and PMOS transistors. Other arguments stored in a vector define sizes of transistors in output stages. Subsequent steps are shown in Figure 4.

At first, transistors are created. PMOS transistors and NMOS transistor, all of the same length, are placed at the top and at the bottom, respectively. Due to the common width of all PMOS transistors the scaling factors are obtained as the relation (6) of the length L_i of the transistor in the output stage to the length L_{in} of the transistor in the relevant input stage.

The sizes of transistors for the whole current mirror can be automatically calculated using dedicated tools [12]. Thanks to the solution with common widths it becomes easy to design current mirrors with the same heights, which allows to place all of them using the verse strategy (a design strategy of putting cells next to each other in rows), which allows minimisation of the area of the chip.

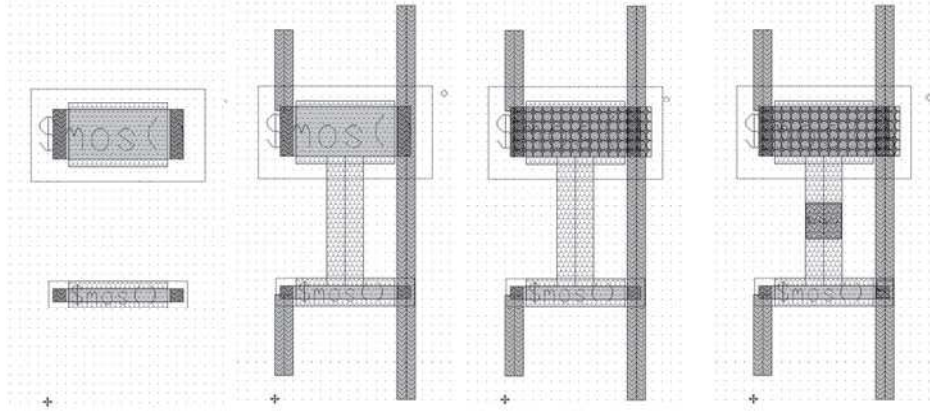


Fig. 4. 8-output current mirror

$$\alpha_i = \frac{L_i}{L_{in}} \quad (6)$$

After placing transistors the connections are drawn. Afterwards FIMP and NTUB layers, and eventually VIA's are drawn. As an example a piece of code with a command which creates a via between the first and the second metal layers is presented below:

```
extern net_out_w = 1.0;
extern nmos_l = 3.775;
```

```
local outvia_x = inv_w(nmos_l) - net_out_w;
local outvia_y = vss_h + vss_to_nmos;
```

```
$add_point_device("$via", @via, [], [x+outvia_x, y+outvia_y], [{"via_w", net_out_w},
{"via_l", net_out_w}, {"t", "m1m2"}, {"cinx", void}, {"ciny", void}, {"dorc", "d"}],
@placed);
```

As we can see it allows us to parametrise dimensions of a via which is defined using the `net_out_w` variable. It is also possible to easily describe the position of a via ($x + outvia_x, y + outvia_y$), which can be calculated using appropriate constants with port sizes and the distance between the VSS port and nmos transistors, also using predefined functions calculating the width of the inverter stage. In the same way all paths, layers and vias can be drawn with the possibility of parametrisation.

Using the initial function and the vector with input arguments, an n-output current mirror can be generated in a loop containing functions which create an output stage, add a relevant input stage and proper connections. The number of output stages in a particular current mirror is the same as the size of the DCT which is set in the part of the script containing parameters. The result has been shown in Figure [5].

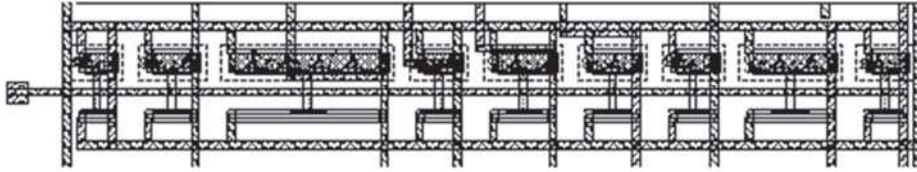


Fig. 5. Layout of a single current mirror

The next part of the AMPLE script uses a function which generates a current mirror layout and builds the DCT. The number of iterations in which the mirrors are generated, depends on the number specified at the beginning of the script. Connections between mirrors are also placed in this part. Finally input and output ports must be created to finish the generation of the DCT layout. This is the place where we can decide if we want them on the left, right, top or bottom side of the layout.

If the size of the DCT is set to 8x8, the layout shown in Figure 6 will be generated. It is worth noticing that using AMPLE and the presented method of dividing the circuit to 8-output current mirrors we can quickly and automatically obtain the layout of a circuit which consists, in this specific case, of 288 transistors. The method is also a solution for avoiding making mistakes during the process of designing a layout.

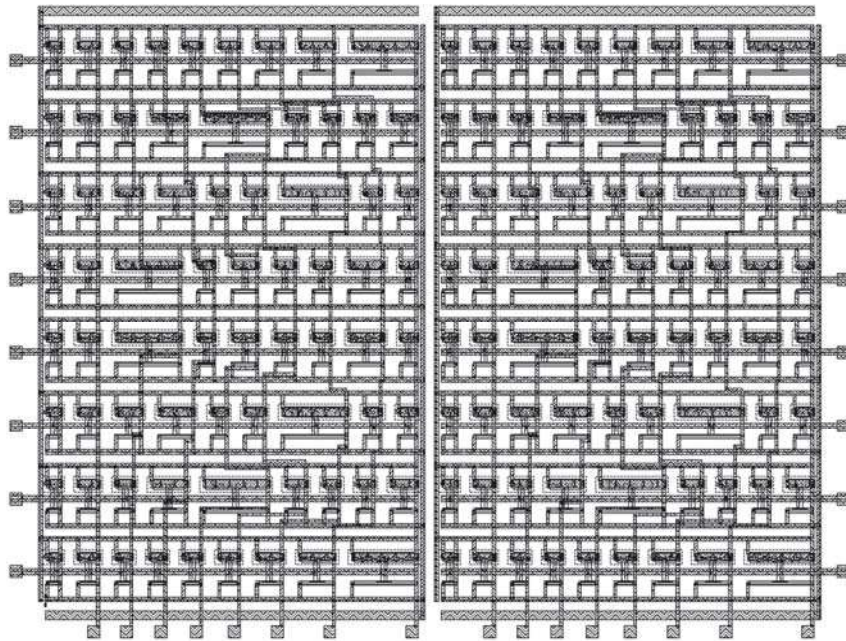


Fig. 6. 8x8 DCT Layout

It would take many hours to design such layouts in a traditional way. Furthermore, there would be no possibility for a fast rebuilding of circuits to other sizes. AMPLE allows to obtain circuits in a short time and offers the possibility of rebuilding.

5. SUMMARY

It becomes more and more often required to propose and improve methods for the automation of the analog circuits layout design process. The AMPLE script language is a perfect solution in case of a need to have custom layout generators, especially involving circuits that are often used as subcircuits in larger devices. We then only need to set input parameters, run the script and the layout is ready. Many hours of designing a layout are saved for other purposes. In addition, it is possible to make a generator independent from the technology used. In other words, all of the repeatable and time consuming tasks connected with layout generation can be automated using AMPLE scripts. In our work we have shown how to quickly design a layout of an analog circuit with the possibility of parametrisation, using the DCT circuit as an example.

BIBLIOGRAPHY

- [1] Yilmaz E., Dundar G., 2009. Analog layout generator for CMOS circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems archive, Volume 28, Issue 1, pp. 32-45.
- [2] Graeb H., Balsa F., Castro-Lopez R., Chang Y.W., Fernandez F.V., Lin P.H., Strasser M., 2009. Analog layout synthesis – Recent advances in topological approaches. Design, Automation & Test in Europe Conference & Exhibition, DATE'09, pp. 274-279.
- [3] Castro-Lopez R., Guerra O., Rocca E., Fernandez F.V., 2008. An Integrated Layout-Synthesis Approach for Analog ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume: 27, Issue: 7, pp. 1179-1189.
- [4] Jain A.K., 1999. Fundamentals of Digital Integrated Circuits: Analysis and Design, McGraw-Hill, Inc.
- [5] Handkiewicz A., 2002. Mixed-Signal Systems: A Guide to CMOS Circuit Design. John Wiley and Sons.
- [6] Pankaala M., Virtanen K., Paasio A. i in. 2006. An analog 2-DCT processor. IEEE Trans. Circ. and Syst. for Video Technol. Vol. 16, no. 10, pp.1209-1216
- [7] Mentor Graphics Corporation, 2005. AMPLE for IC Flow Reference Manual, 1.1 edition.
- [8] Mentor Graphics Corporation, 2005. AMPLE for IC Flow User's Manual, 1.1 edition.
- [9] Mentor Graphics Corporation, 2005. IC Station Reference Manual, 1.1 edition.
- [10] Mentor Graphics Corporation, 2005. IC Station User's Manual, 1.1 edition.
- [11] Bhaskaran V., Konstantinidies K., 1995. Image and Video Compression Standards, Kluwer Academic Publishers, Boston, MA.
- [12] Rudnicki R. Chosen tools for automatic design of switched-current circuits, (in Polish), Ph.D. dissertation, Poznań University of Technology.

AUTOMATYCZNA GENERACJA LAYOUTU UKŁADU DCT
PRZY POMOCY JĘZYKA AMPLE

Streszczenie

Projektowanie layoutów układów SI nie jest zadaniem łatwym. Proces ten wymaga dużych nakładów czasu, istnieje ogromne ryzyko popełnienia pomyłki przez projektanta, a projektowane układy są zależne od technologii, co wymusza ich całkowitą przebudowę w sytuacji zmiany technologii na nowszą. Zadanie to byłoby dużo prostsze, gdyby istniały narzędzia CAD automatyzujące proces projektowania. W obszarze tym możliwe jest wykorzystanie zaproponowanego w artykule rozwiązania – użycie skryptowego języka AMPLE dostępnego w środowisku ICStation. Oprócz możliwości szybszego zaprojektowania prototypu, generator stworzony przy pomocy języka AMPLE umożliwia parametryzację projektowanych urządzeń SI, które stają się niezależne od technologii. Stanowi to daleko idące udoskonalenie procesu projektowania układów scalonych wykonanych w technice SI. Niniejszy artykuł opisuje zaproponowaną metodę automatycznego generowania layoutów przedstawiając jako przykład kolejne etapy realizacji układu DCT.

Słowa kluczowe: AMPLE, layout, dyskretna transformata kosinusowa, zwierciadło prądowe, układ analogowy