INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2013, VOL. 59, NO. 4, PP. 405–413
Manuscript received November 5, 2013; revised December, 2013. DOI: 10.2478/eletel-2013-0050

Designing Method of Compact n-to-2^n Decoders

Ireneusz Brzozowski, Łukasz Zachara, and Andrzej Kos

***Abstract*—What decoder is, everyone knows. The paper presents fast and efficient method of layouts design of *n-to-2^n*-lines decoders. Two scenarios of layout arrangement are proposed and described. Based on a few building blocks only, especially prepared, and appropriate procedure of their placement, a decoder of any size can be build. Layouts of all needed fundamental blocks were designed in CMOS technology, as standard library. Moreover, some important parameters, such area, power dissipation and delay, were assessed and compared for decoders designed with proposed method and traditional. Power consumption were considered under extended model, which takes into account changes of input vectors, not only switching activity factor. All designs were done in UMC 180 CMOS technology.**

***Keywords*—Decoder, address decoder, standard cell, layouts design, CMOS technology, power dissipation, power consumption, delay**

I. Introduction

ONE of important functional blocks in digital circuits are decoders, generally *n*-to-*m*-line. They can be found in any selecting circuits, such as multiplexers, address decoders etc. They also can be used for realization of logic function. Especially, address decoders play important role in memory designing. Due to large amount of storage cells in today's memories it can be found various solutions of address decoder designs leading to power consumption reduction and performance increasing. Usually different kind of precharging dynamic decoders are used [1], [2]. Some solutions use hierarchical decoders with predecoding. In [3] authors implement binary tree decoder built of demultiplexers.

The solution of decoders design presented in this paper can be placed between hierarchical decoders and with predecoding ones. Original idea is drawn from [4]. It is based on decoders extension by adding next levels of AND gates and 1-to-2-line decoder. The procedure starts with 1-to-4-line decoder, built of four AND gates and two 1-to-2-line decoders – ordinary inverter gate (see Fig. 1).

It can be observed in schematic diagram of 3-to-8-line decoder (Fig. 1), that it can be partitioned into blocks. And in consequence a size of the decoder can be extended in easy way.

There is a problem with implementation of the decoder in CMOS technology, because AND gates are not "natural" in CMOS – their realization needs two gates, NAND and NOT serially connected. It increases number of transistors, power consumption and delay. So structure of the decoder has to be implemented directly with NOT, NAND and NOR gates only.

This paper includes the results of work carried out within The National Center for Research and Development project no. N R13 0065 10.

I. Brzozowski, Ł. Zachara, and A. Kos are with AGH University of Science and Technology, Department of Electronics, Mickiewicza 30, 30-059 Kraków, Poland (e-mail: ireneusz.brzozowski@agh.edu.pl).

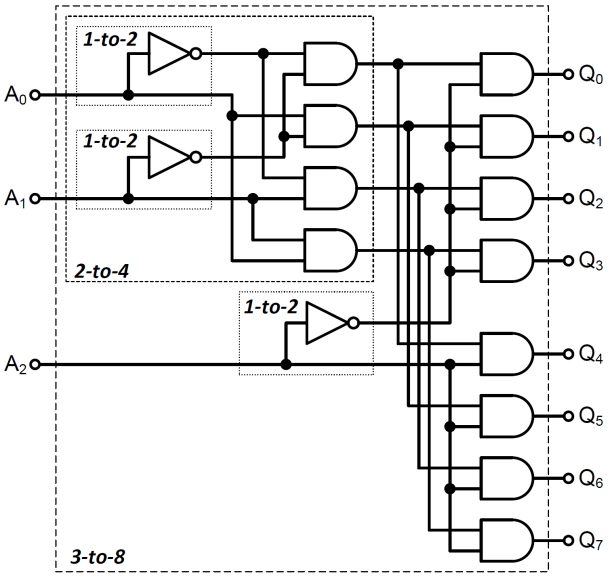


Fig. 1. A 3-to-8-line decoder - the method of decoders extension.

Firstly, the idea of decoders layout design as standard cell, was presented in [5]. But authors have made some efforts to improve previously presented solutions and in consequence obtained results are quite satisfactory. One of the important is reduction of decoders area and small increasing the speed.

II. Decoder Building Blocks Implemented in CMOS

A. General Idea

Classical implementation of a decoding function consists in direct realization of all minterms of the function, as products with AND gates. So, for *n*-inputs decoder standard products can be represented as follows:

$$m_i = l_0 l_1 l_2 ... l_{n-1} \tag{1}$$

where: l_0, l_1, ... l_{n-1} means literals – direct or negated input variable. Based on de' Morgan laws it is possible to transform (1) in such a way, that it will describe minterms implementation in CMOS technology with two-input NAND and NOR gates. Thus, a product can be exchanged by negated sum as follows:

$$a \cdot b = \overline{\overline{a} + \overline{b}} \tag{2}$$

Starting from last variable l_{n-1} using double negations and (2) we can progressively obtain equation described implementation of standard product with two-input NAND and NOR gates alternately connected in serial. Thus, following equation

described consecutive steps of minterm equation exchange to NAND/NOR realizations can be written down as:

$$\begin{aligned} m_i &= l_0 l_1 l_2 ... l_{n-1} = \overline{\overline{l_0 l_1 l_2 ... l_{n-4} l_{n-3} l_{n-2} l_{n-1}}} = \\ &= \overline{\overline{l_0 l_1 l_2 ... l_{n-4} l_{n-3} l_{n-2}} + \overline{l_{n-1}}} = \\ &= \overline{\overline{\overline{l_0 l_1 l_2 ... l_{n-4}} + \overline{\overline{l_{n-3}}} \cdot l_{n-2}} + \overline{l_{n-1}}} = ... \end{aligned} \tag{3}$$

Finally, the last level of designed decoder will always consist of NOR gates, but previous with NAND, and so on. Depending on number of variables – the decoder inputs – the first level will based on NAND or NOR gates. Important is, whether it is even or odd number. For better explanation of these dependencies below are considered three-, four- and five-variable minterms.

$$m_{i3} = l_0 l_1 l_2 = \overline{\overline{l_0 \cdot l_1} + \overline{l_2}} \tag{4}$$

$$m_{i4} = l_0 l_1 l_2 l_3 = \overline{\overline{\overline{l_0} + \overline{l_1}} \cdot l_2 + \overline{l_3}} \tag{5}$$

$$m_{i5} = l_0 l_1 l_2 l_3 l_4 = \overline{\overline{\overline{l_0 \cdot l_1} + \overline{l_2}} \cdot l_3 + \overline{l_4}} \tag{6}$$

The first stage is always a *2*-to-*4*-line decoder realized with NAND or NOR gates, described by literals l_0 and l_1 in (4), (5), and (6). Considering above equations for whole a decoder it can be noticed that in case of NOR operations literals are negated, but in case of NAND not. It leads to different place of inverters in blocks of NOR and NAND gates. For NANDs the inverter has to be used for less significant half of the decoder outputs, but for NORs for more significant half. Such procedure of a decoder creation ensures logic "1" at selected output. But dual proceeding can be applied to obtain logic "0" at selected output of the decoder. In this case last stage of the decoder will be made of NAND gates.

In the next paragraph detailed description of building blocks and schematic diagrams are presented and further on two scenarios of interconnections are shown.

B. Schematic Diagrams

Based on above presented detailed derivations all blocks needed to create any size decoder can be designed. So, following blocks are developed. Blocks of the first level are BASE_NOR and BASE_NAND. There are *2*-to-*4*-line decoders (Fig. 2). Next blocks are NORS and NANDS, which consist of four NOR or NAND gates respectively (Fig. 3). And the last ones are NNORS and NNANDS consisting of four appropriate gates with inverter. These blocks are used in next levels. Schematic diagrams of blocks are presented in Fig. 4.

Designing of any size decoder, on schematic level is very simple using above described blocks and previously presented principles of theirs placing and connecting.

One can observe from Fig. 1, that for third and next stages of the decoder are enough to use one inverter for upper or lower half of gates. But in first scenario of blocks connections authors used blocks presented in Fig. 4. It was dictated by equalization of blocks dimensions and simplification of interconnections. Additionally, as small as possible number of blocks were prepared and easy way to placing them was developed.

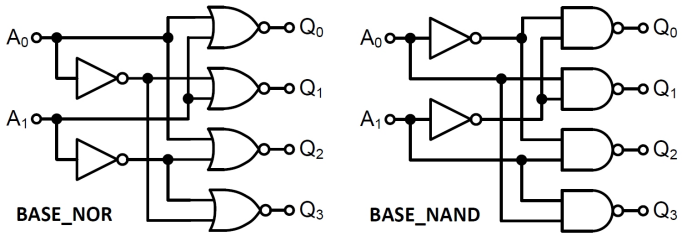


Fig. 2. Blocks BASE_NOR and BASE_NAND - starting blocks.

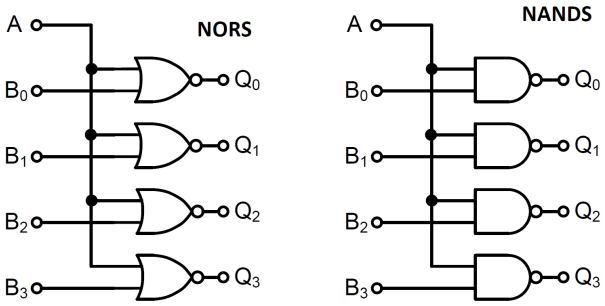


Fig. 3. Blocks NORS and NANDS - next levels blocks.

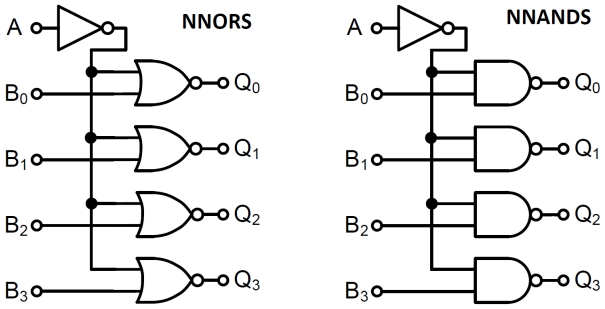


Fig. 4. Blocks NNORS and NNANDS - next levels blocks.

The second scenario of blocks routing based on the same building blocks and new connections cells. Moreover previously mentioned inverter was placed only one time for particular stage of a decoder. Even though the logical diagrams were the same it required designing of new layouts for blocks consists of NAND and NOR gates. Differences will be explained in detail in next paragraphs with layouts description.

C. Layouts

The important stage of the work was such preparation of layouts of the building blocks, in order to obtain easy and flexible way to create decoders of any size. Beside these blocks, with gates let's say – functional, auxiliary blocks with connection lines are needed too. Layouts of connection blocks should be arranged in such way, that building of any decoder will be natural, without additional complex design rules. They should be universal and their number has to be as small as possible. Because in bigger decoders connections can take much more space than functional blocks the second target is to

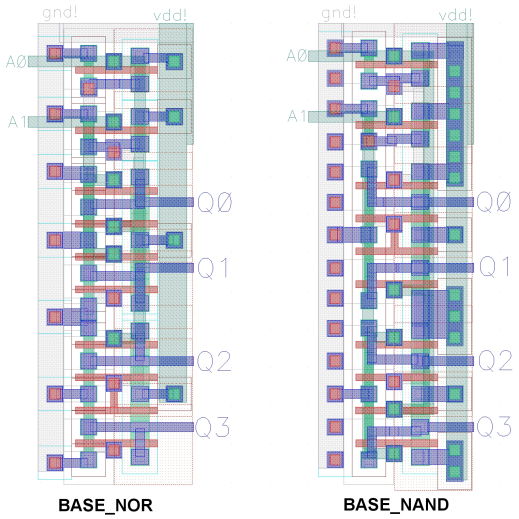


Fig. 5. Layouts of the first functional blocks.

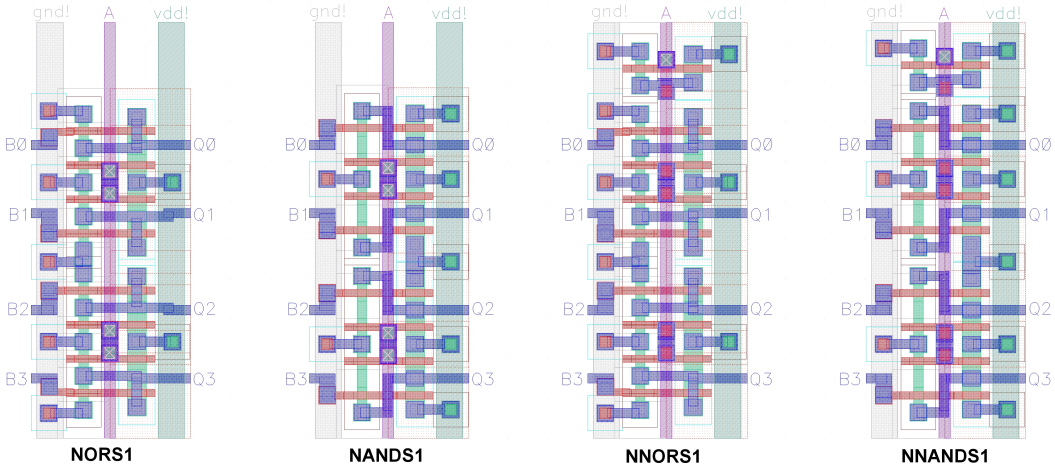


Fig. 6. Layouts of remaining functional blocks - the first version.

minimize area of the connection cells even at the cost of their layouts design complexity. So, authors prepared two versions of connection cells.

Layouts of all building blocks were designed in CMOS UMC 180nm technology in CADENCE environment. Figure 5 shows the first functional blocks: BASE_NOR, BASE_NAND, which are used in both versions of decoders designing. And Fig. 6 shows remaining functional blocks: NORS1, NANDS1, NNORS1, and NNANDS1 designed in the first version.

In the second scenario of decoders designing the same blocks: BASE_NAND and BASE_NOR are used at inputs, but remaining functional blocks are little bit different. Also two blocks with four NAND gates and two blocks with four NOR gates were designed but through them two lines for input signal *A* are placed. The NANDS2 and NORS2 use direct input signal (*A*) but NNORS2 and NNANDS2 use negated signal (*NA*) – Fig. 7. The line *A* is made with metal4 and *NA* with metal1 layers. So, the line *A* is placed directly above line *NA* which is connected to inputs of gates in NNORS2 and NNANDS2 cells. In NORS2 and NANDS2 cells the *NA* line have to bypass vias from the *A* line to gates inputs, made with metal1. Additionally the inverter from these blocks was moved to supply cell.

The connection blocks designed for the first version of decoders are shown in Fig. 8. These blocks consist of metal1 and metal2 lines and vias between them. The cells are named appropriately to shapes of metal lines: Z=1, ZH1, ZII1, ZL1, ZY1 (“1” indicates the first version). The connection blocks in the second version have reduced area thanks to use of four metal lines stacked one above other. Layers of metal2 to metal5 form vertical lines and metal1 horizontal. Layouts of all cells are presented in Fig. 9a. Additionally Fig. 9b shows cross-section of one cell. Such solutions reduces area but outputs of the decoder are not in increasing order.

There are other two cells with metal lines only, which are used to forming supply lines for blocks including gates – named: S_c1 and S_g1 (Fig. 10). In the second version the inverter from NNORS2 and NNANDS2 is placed in supply cell and feeds the negated line *NA* (Fig. 11). The supply cells ensures creation of supply lines for all stages of the decoder.

Described blocks compose a complete library for building the decoders. In order to efficient usage of the library,

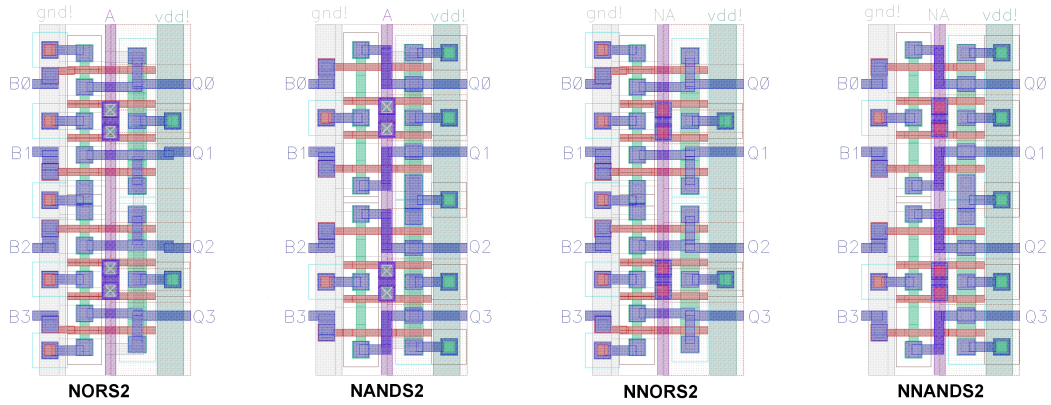


Fig. 7. Layouts of remaining functional blocks - the second version.

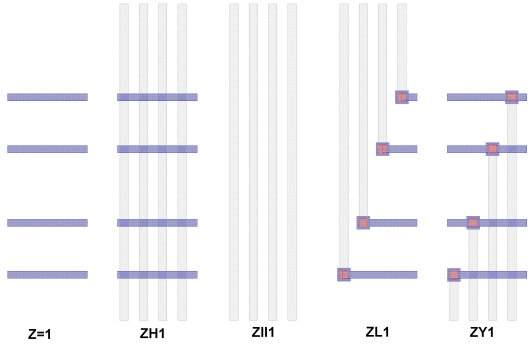


Fig. 8. Layouts of connection blocks in the first version.

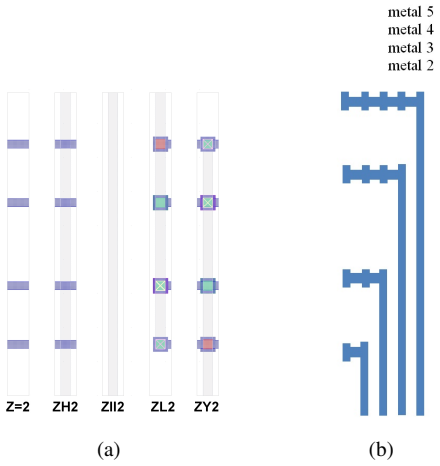


Fig. 9. Layouts of connection blocks in the second version (a), and cross-section of the cell ZY2 (b).

appropriate design rules for placing of the cells are needed. Generally, the rules come from theoretical analysis presented in previous paragraph (IIA). Details will be explained in the next paragraph based on examples of decoders designing.

III. Layouts of Example Decoders

Three example decoders were built with prepared library of cells for universal designing of n-to-2^n-lines decoders. The examples will be used for explanation of cells placing. Figure 12 shows layout of *3*-to-*8*-lines decoder designed with

Fig. 10. Layouts of supply cels S_c1 and S_g1 - the first version.

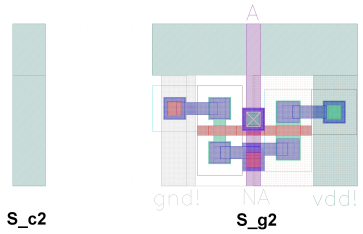


Fig. 11. Layouts of supply cels S_c2 and S_g2 – the second version.

cells in the first version. The decoder is built of two stages. The last level is consists of NOR gates arranged with two blocks. The NORS1 block is used for lower outputs (here: Q0 ÷ Q3), and the NNORS1 for upper outputs (Q4 ÷ Q7). At the first stage the BASE_NAND block is used. For connecting of the stages only ZY1 and ZL1 cells are used. Cells S_c1, and S_g1 ensure supply. Figure 13 shows the same decoder made with cells designed in the second version – appropriate cells were used. It can be seen that its outputs are not arranged in ascending order. The reason is designing of connection cells ZL2 and ZLY2, as is shown in Fig. 9.

It is the simplest example, so only two connection blocks are enough. In case of larger decoders realization of connections is a little more difficult. It is illustrated in the following examples.

Let us consider the *4*-to-*16*-lines decoder. Its schematic diagram divided in blocks is shown in Fig. 14. Similarly to previous decoder the last stage consists of NOR gates. But now, due to number of the decoder outputs, two NORS and two NNORS blocks have to be used for build of this decoder. Previous stage consists of NAND gates. In case of these gates NNANDS blocks are used for lower outputs, but for higher ones NANDS have to be used – see (3). It is inversely to blocks with NOR gates. And finally, at the first level the BASE_NOR is placed. Interconnections between the first and the second level are the same as in *3*-to-*8*-lines decoder. But between

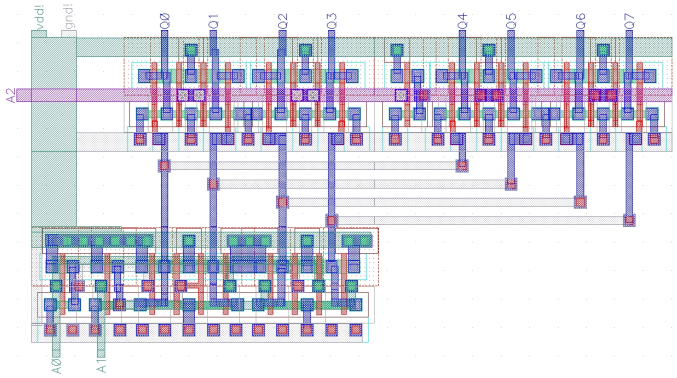


Fig. 12. Layout of *3-to-8-line* decoder – the first version.

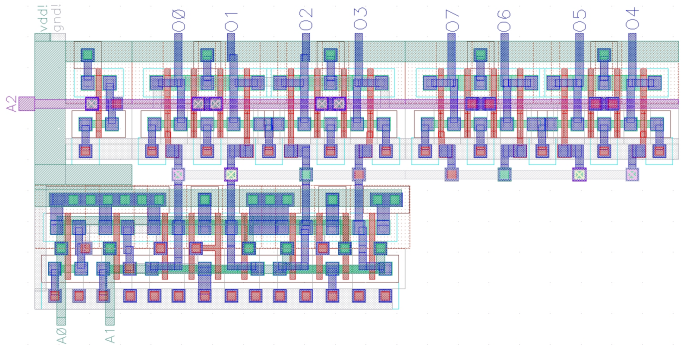


Fig. 13. Layout of *3-to-8-line* decoder – the second version.

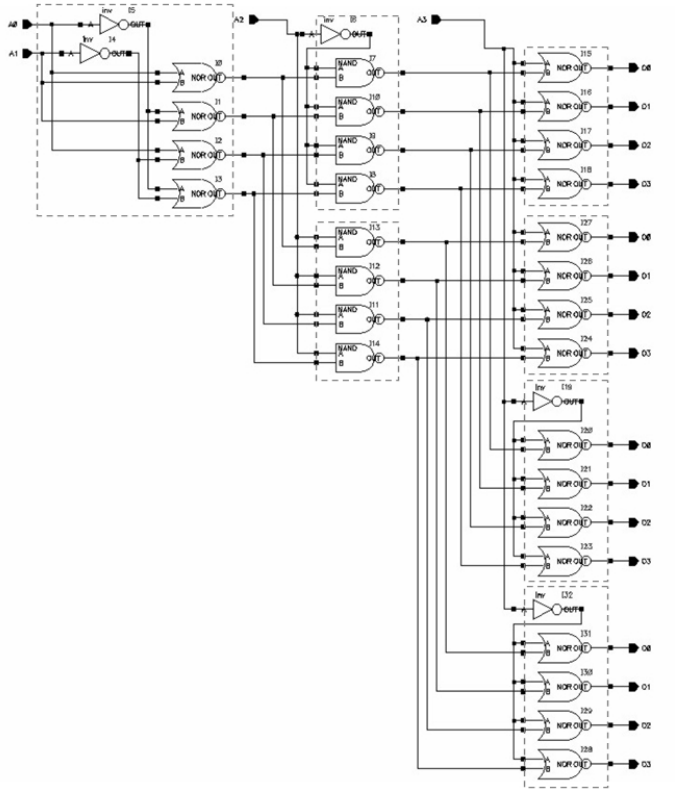


Fig. 14. Schematic diagram of *4-to-16-line* decoder with functional blocks.

next stages (2^{nd} and 3^{rd}) all designed connection blocks will be used. Layout of the decoder, designed in the first version, is shown in Fig. 15. The shape of lines corresponds to schematic diagram (Fig. 14). And for comparison Fig. 16 shows the same decoder made in the second version. The schematic diagram (Fig. 14) does not correspond exactly to the layout in the second version, because there are no inverters in NNORS2 and NNANDS2.

The method of connection cells placing will be better explained using larger design. So, additionally in Fig. 17 is presented layout of 5 to 32 lines decoder with connection cells marked in colors. The placing method can be described based on observation of interconnections between two last stages of gates. Cells ZY and ZL are placed on the top-right to bottom-left diagonals only. Other appropriate cells are placed above and below the diagonals, making specific triangles. Above the ZY diagonal cells are placed with parallel, horizontal lines (Z=). But under the diagonal cross connections are needed, so cells ZH are used. In bottom part with ZH diagonal, above it parallel, vertical connections are used (ZII). And below the diagonal horizontal lines are ensured by Z= cells. This scheme of connection will be used in larger decoders, only number of cells increase exponentially according to number of outputs.

IV. Comparative Analysis of Decoders

Three decoders designed using proposed method in two versions were compared with decoders designed in traditional style – with multi inputs gates. An example of such decoder is shown below, i.e. *3*-to-*8*-lines (Fig. 18). From layouts of the decoders netlists were extracted with parasitic elements RC and simulated for assessment of theirs energy and time parameters. Moreover, area of layouts and number of transistors were compared too.

A. Energy Parameters

Authors used extended power model, which based on gates driving way [6], for power estimation accuracy improvement. This power model consists in taking into account changes of input vectors. Inputs of a circuit are driven together (vector of primary inputs), not separately. The parameter describing the circuit activity is called probability of a gate driving way. Generally, this is a probability of a change of the gate input vector between two values [7]. For considered gate it is needed to have into account all possible changes of input vectors. Thus for n-input gate it gives 2^{2n} changes. Figure 19 shows all driving ways for 2-input gate.

Currents flowing through the gate terminals are measured for each change of the gate input vectors. Inputs and supply terminal are taken into account. These values and value of supply voltage are used for calculation of the equivalent capacitances. So it represents portion of charge flowing through a terminal of the gate under specific input vector change. Considering all possible changes of input vectors, tables similar to that shown in Fig. 19 are obtained for each terminal of the gate. Generally the model of power consumption can be represented as it is shown in Fig. 20 with set of tables

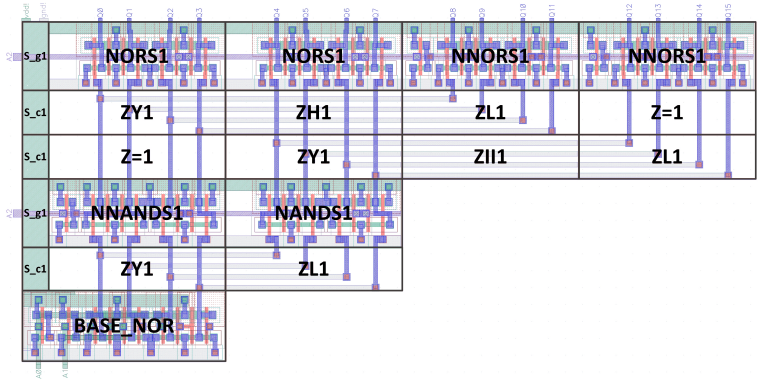


Fig. 15. Layout of *4-to-16-line* decoder – the first version.

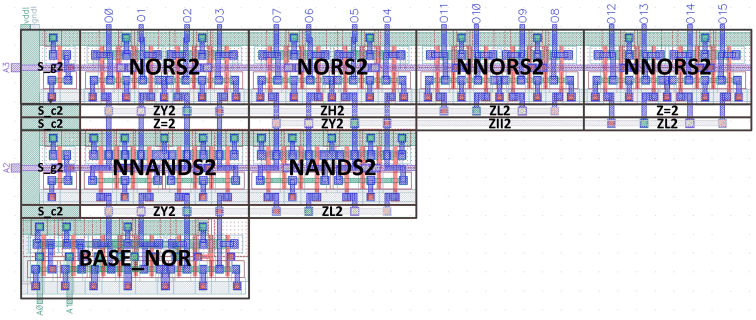


Fig. 16. Layout of *4-to-16-line* decoder – the second version.

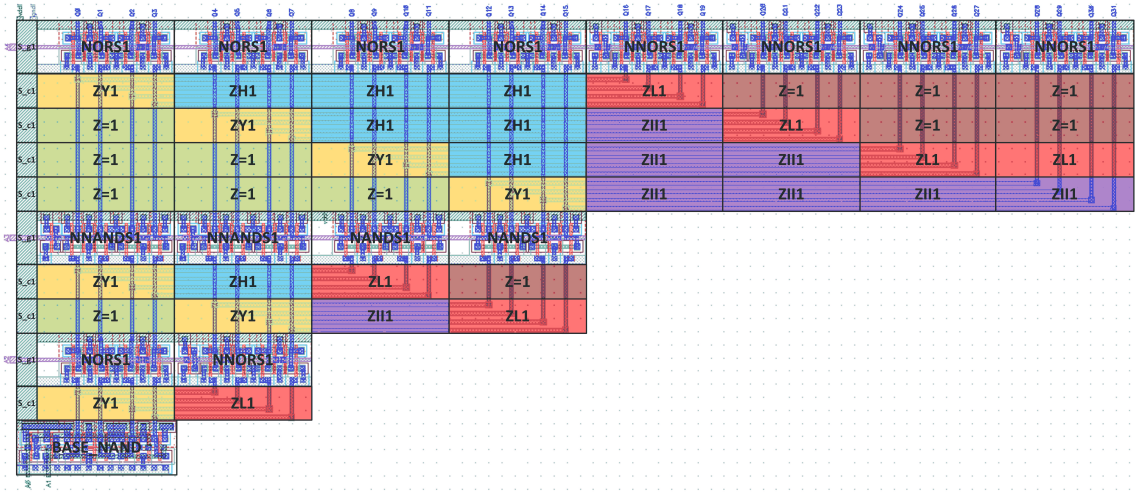


Fig. 17. Layout of *5-to-32-line* decoder – idea of connection cells placeing.

containing energy parameters – equivalent capacitance – for each terminal of the gate.

Decoders designed using proposed method (with blocks) and in traditional way (with multi-input gates) were simulated in conditions allowing obtaining equivalent capacitance for above described extended power model. Input, driving signals had such parameters, that only dynamic, capacitive power consumption occurred in tested circuits. There was no quasi-short. Static lose in used technology (180nm) can be omitted.

Obtained results of energy parameters assessment for considered decoders are collected in tables and additionally compared in graphs. For example, only for the smallest decoder, tables below contain values of internal load and all equivalent capacitances. Results for block decoder designed with the first version of cells are collected in Tab. I and Tab. II. Equivalent capacitance for the decoder designed in the second version are shown in Tab. III and Tab. IV. And the last two tables contain values for traditionally designed decoder. The internal capacitance (C_{Lint}) represents energy loses regarding to internal load of the decoder. The capacitance C_{Lall} is sum of all capacitance values for corresponding driving ways ($C_{\mathrm{Lint}}$and $C_{\mathrm{In\_x}}$ of all inputs). So for example, based on Tab. I and Tab. II input capacitance of the block decoder (version one) can be easily calculated.

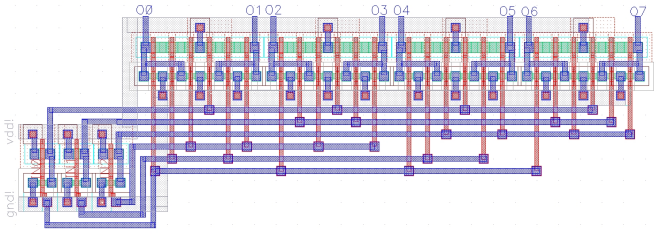
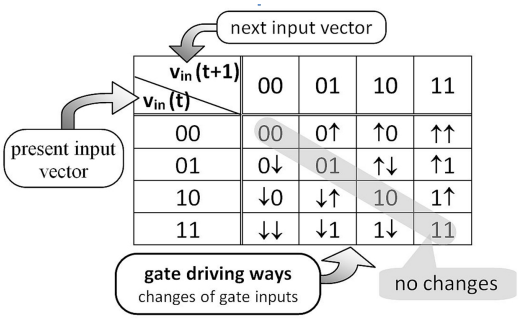


Fig. 18. Traditional 3 to 8 lines decoder built of 3-input NOR gates.



$v_{in}(t+1)$ / $v_{in}(t)$	00	01	10	11
00	00	0↑	↑0	↑↑
01	0↓	01	↑↓	↑1
10	↓0	↓↑	10	1↑
11	↓↓	↓1	1↓	11

Fig. 19. All possible changes of input vectors for 2-in. gate – gate driving ways.

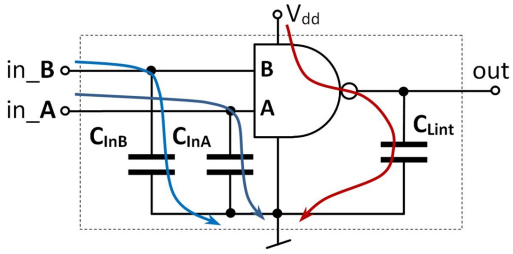


Fig. 20. Capacitive model of dynamic power dissipation for 2-input NAND.

Values of equivalent capacitance vs. driving way are useful for power consumption estimation but comparison of decoders are arduous. So, for easier assessment of decoders above values can be summed. Table VII presents summed values of input, internal, and all equivalent capacitances. These summed values represent power consumption for uniform probability of input vectors change. The comparison of decoders is presented in Tab. VII. The power consumption profit is shown in percentage versus traditional decoders.

It can be seen that for larger decoders the lowest power consumption is obtained for block decoders designed in the second scenario.

For better analysis values of equivalent capacitance can be collected for the same driving ways of all considered decoders. Such bar graphs are presented in Fig. 21. In this way energy parameters of decoders can be easy compared for specific driving ways.

Detailed analysis of such graphs allows a designer to choose decoder characterized by lower power consumption. But the probability of input vectors changes are needed.

TABLE I
EQUIV. CAP. C_{Lint} FOR 3 TO 8 LINES BLOCK DECODER VER. ONE [FF]

Present vector \ Next vector	***000***	***001***	***010***	***011***	***100***	***101***	***110***	***111***
000	0.00	13.98	15.48	20.37	5.04	16.27	17.56	21.99
001	21.91	0.00	24.43	15.80	24.31	4.74	26.09	17.59
010	19.36	19.64	0.00	15.01	21.64	21.83	4.73	18.04
011	27.71	18.41	22.08	0.00	29.81	20.53	25.02	4.66
100	14.27	27.84	28.00	33.09	0.00	14.16	15.64	19.63
101	33.85	14.25	35.49	28.17	21.94	0.00	23.56	14.85
110	31.33	31.37	14.27	28.68	19.39	19.68	0.00	14.07
111	39.15	30.23	34.89	14.34	28.73	18.48	22.93	0.00

TABLE II
ALL EQUIV. CAP. C_{Lall} FOR 3-TO-8-LINES BLOCK DECODER VER. ONE [FF]

Present vector \ Next vector	***000***	***001***	***010***	***011***	***100***	***101***	***110***	***111***
000	0.00	22.63	24.32	37.59	16.16	36.10	37.48	50.39
001	21.91	0.00	33.34	24.30	35.46	15.83	46.08	37.25
010	19.36	28.35	0.00	23.45	32.80	41.71	15.83	37.61
011	27.71	18.75	22.08	0.00	40.95	32.02	36.18	15.75
100	14.27	36.46	36.91	50.23	0.00	22.92	24.59	36.90
101	33.85	14.25	44.30	36.73	21.94	0.00	32.46	23.35
110	31.33	40.08	14.27	37.11	19.39	28.38	0.00	22.51
111	39.15	30.57	34.89	14.33	28.85	18.82	22.93	0.00

TABLE III
EQUIV. CAP. C_{Lint} FOR 3-TO-8-LINES BLOCK DECODER VER. TWO [FF]

Present vector \ Next vector	***000***	***001***	***010***	***011***	***100***	***101***	***110***	***111***
000	0.00	21.37	21.68	34.56	16.44	34.44	34.72	47.53
001	19.52	0.00	29.60	21.74	33.30	16.09	42.46	34.69
010	18.16	27.11	0.00	22.12	31.80	40.62	16.09	35.94
011	25.26	17.52	19.53	0.00	38.66	30.95	33.79	16.07
100	16.03	36.63	35.95	48.64	0.00	21.58	21.88	33.82
101	32.91	16.05	42.13	35.89	19.54	0.00	28.66	20.80
110	31.52	40.31	16.07	37.26	18.16	27.11	0.00	21.17
111	38.28	30.75	33.73	16.13	26.35	17.53	20.32	0.00

TABLE IV
ALL EQUIV. CAP. C_{Lall} FOR 3-TO-8-LINES BLOCK DECODER VER. TWO [FF]

Present vector \ Next vector	***000***	***001***	***010***	***011***	***100***	***101***	***110***	***111***
000	0.00	12.76	14.27	18.80	5.07	14.45	16.00	20.35
001	19.52	0.00	22.14	14.69	21.91	4.76	23.71	16.23
010	18.16	18.47	0.00	13.74	20.42	20.60	4.76	16.23
011	25.26	17.19	19.53	0.00	27.30	19.24	22.41	4.74
100	16.03	28.06	28.48	33.03	0.00	12.88	14.38	18.06
101	32.91	16.05	34.74	28.76	19.54	0.00	21.20	13.75
110	31.52	31.67	16.07	28.88	18.16	18.46	0.00	12.79
111	38.28	30.41	33.73	16.13	26.23	17.19	20.32	0.00

B. *Time Parameters*

The second analysis of designed circuits was assessment of delay times. Based on simulations the t_{pHL} and t_{pLH} for critical paths were measured. Decoders were loaded with capacitance of 10fF. It is approximately equivalent to three inverters. The rise and falling time of input signal were equal to 100ps. Values of the worst delay times for designed decoders are collected in Tab. IX. Average delay time is marked by t_p.

TABLE V
EQUIV. CAP. C_{Lint} FOR 3-TO-8-LINES TRADITIONAL DECODER [FF]

		Next vector							
		000	***001***	***010***	***011***	***100***	***101***	***110***	***111***
Present vector	***000***	0.00	6.75	6.72	7.81	9.86	8.61	8.71	9.35
	001	18.21	0.00	22.90	5.39	23.00	4.05	21.61	5.75
	010	19.89	21.51	0.00	6.60	24.22	23.08	8.33	8.48
	011	38.44	20.15	18.28	0.00	40.52	21.09	23.29	3.92
	100	20.58	22.53	21.42	23.82	0.00	6.59	6.77	7.60
	101	38.11	18.20	37.55	21.40	18.31	0.00	20.26	5.37
	110	38.95	37.68	20.60	22.53	19.93	21.32	0.00	6.44
	111	61.16	37.92	41.03	18.14	41.86	20.03	19.19	0.00

TABLE VI
ALL EQUIV. CAP. C_{Lall} FOR 3-TO-8-LINES TRADITIONAL DECODER [FF]

		Next vector							
		000	***001***	***010***	***011***	***100***	***101***	***110***	***111***
Present vector	***000***	0.00	20.46	21.01	35.27	22.58	34.58	35.42	48.96
	001	18.21	0.00	38.12	20.12	36.16	17.09	49.88	33.60
	010	19.89	35.04	0.00	20.31	36.89	49.10	21.04	34.47
	011	38.44	20.17	18.30	0.00	53.65	34.03	36.75	16.95
	100	20.58	36.12	35.40	51.36	0.00	20.31	21.16	35.06
	101	38.11	18.21	52.47	36.15	18.31	0.00	34.98	19.97
	110	38.95	51.06	20.61	36.09	19.94	34.93	0.00	20.14
	111	61.16	37.77	41.42	18.17	41.91	20.04	19.33	0.00

It can be seen that in cases of bigger decoder, the block decoders are faster than traditional ones. The best results are obtained for decoders designed in version two. Block decoders have multi-stage structure. Thus, it would seem, that they should be slower. But traditional decoders, at inputs of gates, have long lines, which increase input capacitances. It is also observable in results of energy parameters assessment.

C. Area of Decoders

Another important parameter of integrated circuits is area of layout. Table X includes dimensions of designed decoders. Proposed method of decoder designing results in almost triangular shape of layout. But the table contains width and high of the circuits. So, some part of the area, approximately third part of whole area, can be used for other circuits in case of the first version. Using cells designed in the second version dimensions are decreased and in case of *5*-to-*32*-line decoder the area is reduced to 52% in comparison with the first version.

a)

b)

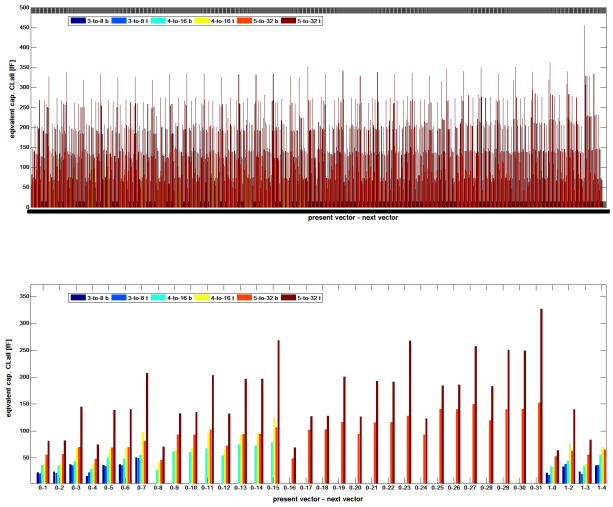


Fig. 21. Eqivalent capacitance (C_{Lall}) of the decoders vs. driving ways a) all values, b) fragment.

Sometimes area of integrated circuits is represented by number of transistors. But in case of decoders most of area is occupied by connection lines. Fortunately authors have proposed the second version of cells to build of block decoders. Table VII shows quantity of transistors in traditional and block decoders versus number of their inputs.

From above table it is observed that using 2-input gates for decoders design total number of transistors increases slowly for block decoders. In case of 8 input decoder number of transistors in the traditional one, is doubled with comparison to the block decoder.

V. CONCLUSION

The universal method of decoders design was presented in this paper. Suitable library consisting of needed cells for decoders construction in easy way was prepared. Two versions of cells were developed and designed. The method can be easy automated allows synthesis of any size decoders.

Three decoders were designed with using of proposed and traditional method and their parameters were assessed. Power consumption, time delay, and area are considered. Analyzing

TABLE VII
SUMMED VALUES OF EQUIV. CAPACITANCE FOR DECODERS [FF]

	C_{Lall}	C_{Lint}	C_{Lin}	C_{Lall}	C_{Lint}	C_{Lin}	C_{Lall}	C_{Lint}	C_{Lin}
decoder	***traditional***			***block ver. one***			***block ver. two***		
3-to-8	1766	1112	654	1643	1186	457	1567	1130	437
4-to-16	17391	10349	7042	12349	9281	3069	11668	8685	2983
5-to-32	173321	97985	75336	91110	67289	23821	81880	60519	21361

TABLE VIII
SUMMED VALUES OF EQUIV. CAP. – COMPARISON [%]

	C_{Lall}	C_{Lint}	C_{Lin}	C_{Lall}	C_{Lint}	C_{Lin}	C_{Lall}	C_{Lint}	C_{Lin}
decoder	***traditional***			***block ver. one***			***block ver. two***		
3-to-8	100.0	100.0	100.0	93.0	106.7	69.9	88.7	101.6	66.8
4-to-16	100.0	100.0	100.0	71.0	89.7	43.6	67.1	83.9	42.4
5-to-32	100.0	100.0	100.0	52.6	68.7	31.6	47.2	61.8	28.4

TABLE IX
DELAY TIMES FOR DECODERS [PS]

	t_{pLH}	t_{pHL}	t_p	t_{pLH}	t_{pHL}	t_p	t_{pLH}	t_{pHL}	t_p
decoder	***traditional***			***block ver. one***			***block ver. two***		
3-to-8	852	827	839.5	455	*156*	305.5	*428*	194	311.0
4-to-16	788	427	607.5	655	394	524.5	*609*	*372*	490.5
5-to-32	1252	712	982.0	960	497	728.5	*850*	*449*	649.5

TABLE X
DIMENSIONS AND AREA OF DECODERS [μM]

	width	**high**	**area**	**width**	**high**	**area**	**width**	**high**	**area**
decoder	***traditional***			***block ver. one***			***block ver. two***		
3-to-8	9.62	29.21	281.00	10.70	22.54	241.2	8.98	20.96	188.2
4-to-16	10.68	68.58	732.43	17.38	43.50	756.0	14.28	38.76	553.5
5-to-32	12.24	159.00	1946.16	34.60	85.42	2955.5	20.86	74.36	1551.1

TABLE XI
NUMBER OF TRANSISTORS IN DECODERS

decoder	**tradit.**	**block**	**diff.**
1-to-2	2	2	0.0%
2-to-4	20	20	0.0%
3-to-8	54	54	0.0%
4-to-16	136	120	11.8%
5-to-32	330	250	24.2%
6-to-64	780	508	34.9%
7-to-128	1806	1022	43.4%
8-to-256	4112	2048	50.2%

obtained results it is seen that decoders built with blocks have better parameters in almost all cases. Especially in case of the second version of block decoders obtained parameters are the best. Thanks to dimensions reduction parasitic capacitance was reduced and in consequence better performance of the decoders was reached.

Proposed method gives easy and fast designing of decoders and they have better parameters than traditional ones. Power consumption was considered with using of extended model. Such approach allows detailed analysis of designed circuits thanks to exploration of input vectors and in consequence selection of the best solution for given conditions.

REFERENCES

[1] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2003.

[2] M. A. Turi and J. G. Delgado-Frias, "High-performance low-power selective precharge schemes for address decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 9, pp. 917–621, September 2008.

[3] Y. Ren, M. Gansen, and T. Noll, "Low power 6T-SRAM with tree address decoder using a new equalizer precharge scheme," in *Proceedings of IEEE International System-on-Chip Conference*, Niagara Falls, USA, 12–14 September 2012, pp. 224–229.

[4] M. M. Mano and C. R. Kime, *Logic and Computer Design Fundamental*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2004, pp. 148–151.

[5] I. Brzozowski, Ł. Zachara, and A. Kos, "Universal Design Method of n-to-2^n decoders," in *Proceedings of International Conference MIXDES 2013*, Gdynia, Poland, June 2013.

[6] I. Brzozowski and A. Kos, "A new approach to power estimation and reduction in CMOS digital circuits," *Integration, the VLSI Journal*, vol. 41, pp. 219–237, February 2008, Issue 2.

[7] ——, "Calculation Methods of New Circuit Activity Measure for Low Power Modeling," in *Proceedings of International Conference on Signals and Electronic Systems (ICSES)*, Krakw, Poland, September 2008, pp. 133–136.