Politecnico di Torino

Integrated Systems Technology

prof. Gianluca Piccinini
prof. Marco Vacca

Modelling of Multiplexers

Group 8

Authors	*student ID*
Jacopo Acquafresca	232417
Carlo Alberto Cristofanini	224689
Marco Guerra	231317
Lorenzo Mairone	228293

Academic year 2016/2017

Contents

Chapter 1

Introduction

Aim of this project is to carry out a comparative analysis at system level between different typology of CMOS multiplexers. We will perform several evaluations concerning the static and dynamic power consumption, the occupied area and time delay in different digital circuit configurations, within 2010 HP, LOP and LSTP technologies. Physical constants, system parameters and technological variables have been taken from TAMTAMS, a useful tool allowing to predict system level features starting from technology variables. In this project we will present multiplexer circuits built with NAND gates, enabling a fast and interesting analysis of the performances.

1.1 Multiplexer architecture

The multiplexer (1.1) is a combinational logic circuit designed to switch several input lines toward a single common output line by the application of a control signal.

Multiplexers can be either digital circuits made from high speed logic gates used to switch digital or binary data. In digital electronics, multiplexers are also known as data selectors because they can "select" among different input lines. They are used as a method to reduce number of data lines required in a circuit design or when a single data line or data bus is used to transfer two or more different digital signals.
Generally, the selection of each input line in a multiplexer is controlled by an additional set of inputs called control lines, and according to the binary condition of these control inputs, either "high" or "low", the appropriate data input is connected directly to the output.
Normally, a multiplexer has an even number of (2^N) data input lines with N selection inputs.
In figure 1.3 is shown the logic circuit of the simplest multiplexer unit next to his truth

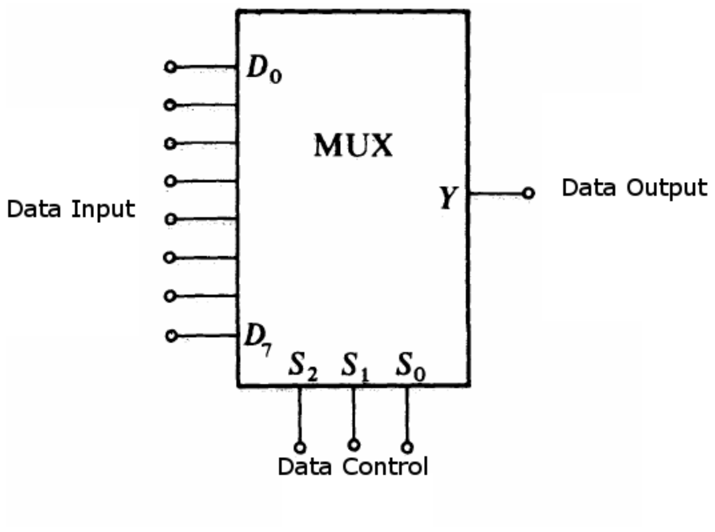


Figure 1.1: *Multiplexer block diagram.*

table. When the selection input is *S = 0*, the upper AND gate is selected (*G0*) and the digital data (0 or 1) present at the input data *D0* is transferred to the output. Conversely when *S = 1*, the upper AND gate is enabled (*G1*) and the value *D1* it is transferred at the output. The logic function that expresses the output is

$$Y = D_0\bar{S} + D_1S. \tag{1.1}$$

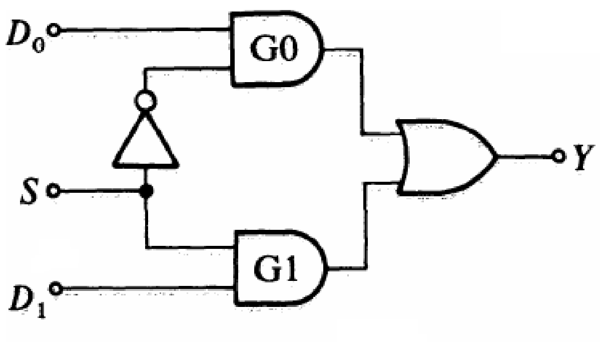
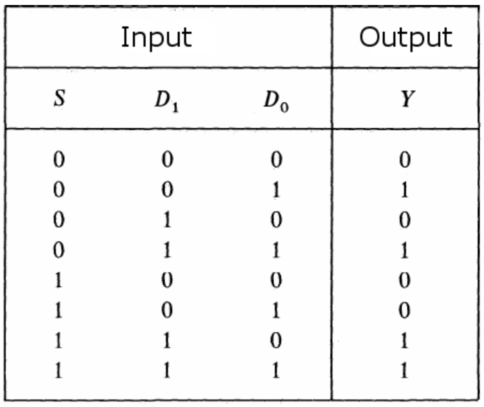


Figure 1.2: *Logic circuit implementation of a multiplexer with two inputs.*



Input			Output
S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 1.3: *Truth table.*

1.2 Multiplexer with NAND gates

In principle, any combinatorial logic function can be realized with enough NAND gates through the two De Morgan theorems:

$$\overline{A \cdot B} = \overline{A} + \overline{B} \tag{1.2}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B} \tag{1.3}$$

The first 1.2 one mentions that the negated product of two variables is equal to the sum of negated variables.
The second 1.3 theorem is stating that the negated sum of two variables is equal to the product of the single negated variables.
Applying to the figure 1.2 the De Morgan's theorems allows to redefine the entire circuit using only NAND gates.

$$Y = \overline{\overline{D_0 + \overline{S + S}} + \overline{D_1 + S}} \tag{1.4}$$

The inverter can be easily imagined as a NAND port with the inputs connected together. By implementing these properties the logical scheme becomes: We choose to use basic NAND configurations as they allow a simpler approach, namely to evaluate the area and power consumption of a single NAND gate and multiply it by the number of those that make up the circuit.
With a single 2 to 1 mux unit is possible to build more complex multiplexers, showing a tree circuit structure. In figure 1.7 is reported a multiplexer 8 to 1 with a word of 1 bit, using 24 NAND gates.

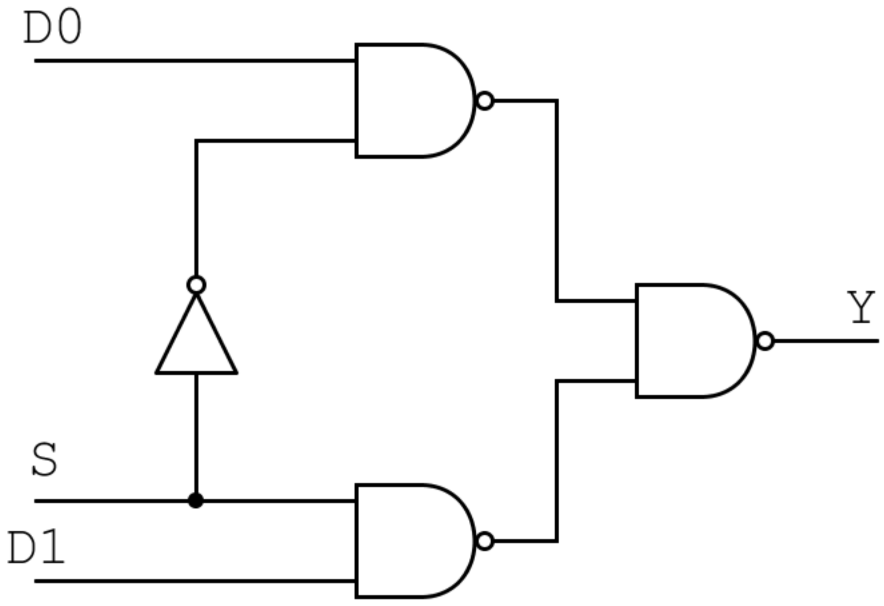
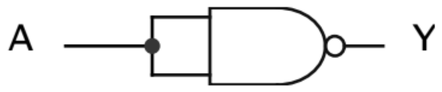


Figure 1.4: *Circuit implementation of a multiplexer with two inputs optimized.*



$$Y = \overline{A \bullet A} = \overline{A}$$

Figure 1.5: *Inverter made with a NAND*

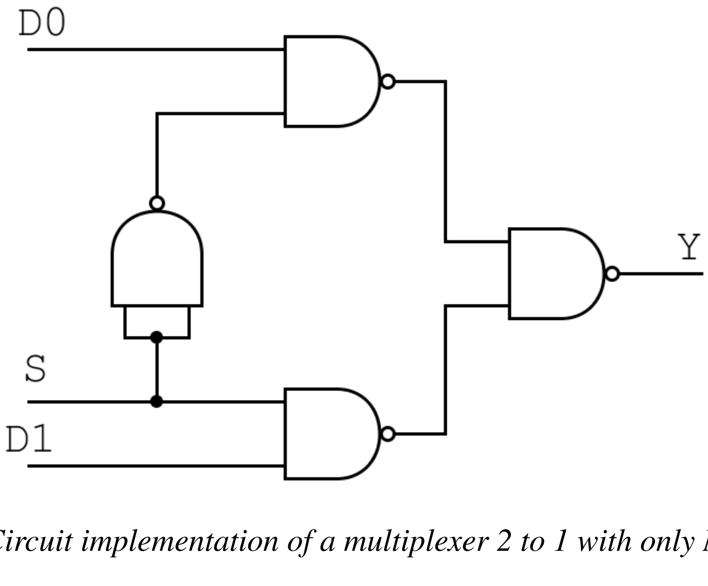


Figure 1.6: *Circuit implementation of a multiplexer 2 to 1 with only NAND gates.*

.

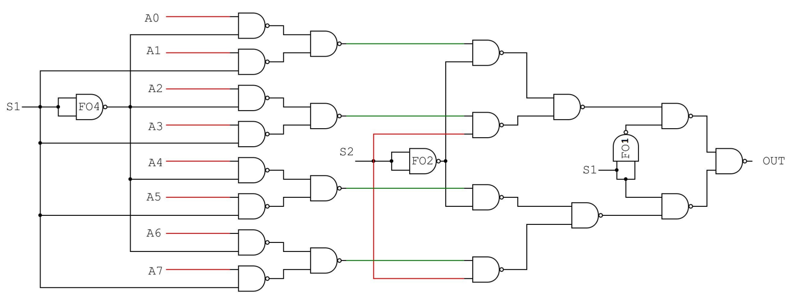


Figure 1.7: *Circuit implementation of a multiplexer 8 to 1 with only NAND gates.*

1.3 CMOS realization of multiplexers

The next step is to realize a MUX circuit with transistors, by implementing CMOS technology. The building unit is reported in figure 1.8 where a NAND gate is showed.

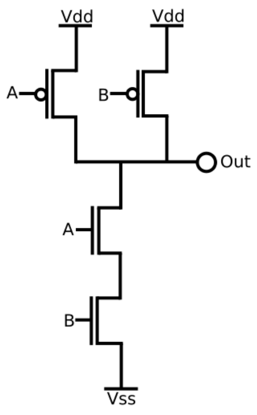


Figure 1.8: *CMOS circuit for a NAND gate.*

An example of CMOS multiplexer realization is reported in the figure below.

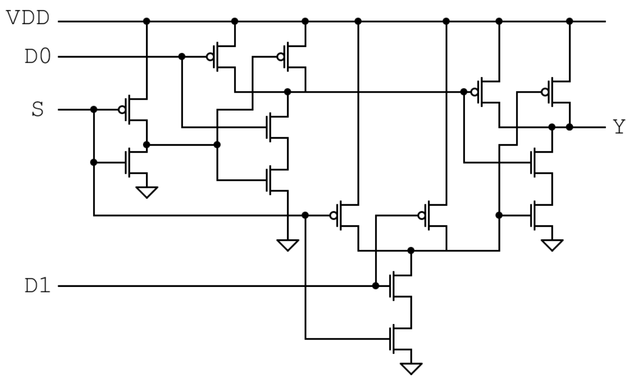


Figure 1.9: *CMOS realization of a MUX 2x1.*

1.4 Types of Multiplexer

Here are summarized the different typologies of mutiplexer analysed. The various multiplexers vary by the number of input signals and the parallelism.

- Multiplexer 2 to 1 parallelism 16 bit
- Multiplexer 2 to 1 parallelism 32 bit
- Multiplexer 2 to 1 parallelism 64 bit
- Multiplexer 8 to 1 parallelism 8 bit
- Multiplexer 16 to 1 parallelism 8 bit
- Multiplexer 16 to 1 parallelism 16 bit
- Multiplexer 16 to 1 parallelism 32 bit
- Multiplexer 16 to 1 parallelism 64 bit
- Multiplexer 32 to 1 parallelism 8 bit
- Multiplexer 32 to 1 parallelism 16 bit
- Multiplexer 32 to 1 parallelism 32 bit
- Multiplexer 32 to 1 parallelism 64 bit
- Multiplexer 64 to 1 parallelism 8 bit
- Multiplexer 64 to 1 parallelism 16 bit
- Multiplexer 64 to 1 parallelism 32 bit
- Multiplexer 64 to 1 parallelism 64 bit
- Multiplexer 128 to 1 parallelism 8 bit
- Multiplexer 128 to 1 parallelism 16 bit
- Multiplexer 128 to 1 parallelism 32 bit
- Multiplexer 128 to 1 parallelism 64 bit

1.5 NAND with TAMTAMS

By using the application TAMTAMS web, it was possible to make a first analysis at system level of several parameters concerning CMOS NAND2 gates employed. The input data used for the analysis are:

- Year: 2010;
- Technology branch: HP, LOP, LSTP;
- model used for I_{gate}: Mastar4.

From the analysis done we've obtained the following results:

```
Output analysis of BULK - HP 2010 - CMOS NAND2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cox = 5.3124e-06
Vtlong = 0.56919
SCE = 0.18999
DIBL = 0.21532
mueff_n = 159.99
mueff_p = 51.647
mueff_n_over_mueff_p = 3,098
%%%%%%%%%%%%%%%%%%%%%%% Fan Out = 1 %%%%%%%%%%%%%%%%%%%%%%%
Igate_FO1 = 1.0147e+05          %%[nA]
Ioff_FO1 = 1.2499e+02           %%[nA]
P_stat_FO1 = 2.0318e-04         %%[W]
P_dyn_FO1 = 2.3776e-06          %%[W]
%%%%%%%%%%%%%%%%%%%%%%% Fan Out = 2 %%%%%%%%%%%%%%%%%%%%%%%
Igate_FO2 = 9.6394e+04          %%[nA]
Ioff_FO2 = 1.1874e+02           %%[nA]
P_stat_FO2 = 1.9302e-04         %%[W]
P_dyn_FO2 = 2.2560e-06          %%[W]
%%%%%%%%%%%%%%%%%%%%%%% Fan Out = 3 %%%%%%%%%%%%%%%%%%%%%%%
Igate_FO3= 9.1320e+04           %%[nA]
Ioff_FO3 = 1.1249e+02           %%[nA]
P_stat_FO3 = 1.8287e-04         %%[W]
P_dyn_FO3 = 2.1343e-06          %%[W]
%%%%%%%%%%%%%%%%%%%%%%% Fan Out = 4 %%%%%%%%%%%%%%%%%%%%%%%
Igate_FO4 = 9.1320e+04          %%[nA]
Ioff_FO4 = 1.1249e+02           %%[nA]
P_stat_FO4 = 1.8287e-04         %%[W]
```

```
P_dyn_FO4 = 2.1343e-06          %%[W]
```

```
Output analysis of BULK - LOP 2010 - CMOS NAND2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cox = 3.8367e-06
Vtlong = 0.59616
SCE = 0.20934
DIBL = 0.16757
mueff_n = 288.52
mueff_p = 67.284
mueff_n_over_mueff_p = 4,288
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 1 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO1 = 1.1652e+03          %%[nA]
Ioff_FO1 = 2.2361e+01           %%[nA]
P_stat_FO1 = 2.0189e-06         %%[W]
P_dyn_FO1 = 9.0581e-07          %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 2 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO2 = 1.1039e+03          %%[nA]
Ioff_FO2 = 2.1184e+01           %%[nA]
P_stat_FO2 = 1.9127e-06         %%[W]
P_dyn_FO2 = 8.5967e-07          %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 3 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO3= 1.1039e+03           %%[nA]
Ioff_FO3 = 2.1184e+01           %%[nA]
P_stat_FO3 = 1.9127e-06         %%[W]
P_dyn_FO3 = 8.5967e-07          %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 4 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO4 = 1.0426e+03          %%[nA]
Ioff_FO4 = 2.0007e+01           %%[nA]
P_stat_FO4 = 1.8064e-06         %%[W]
P_dyn_FO4 = 8.1353e-07          %%[W]
```

```
Output analysis of BULK - LSTP 2010 - CMOS NAND2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cox = 2.4665e-06
Vtlong = 0.84227
SCE = 0.14859
DIBL = 0.18625
mueff_n = 208.33
mueff_p = 51.329
mueff_n_over_mueff_p = 4,059
```

```
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 1 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO1 = 54.85995          %%[nA]
Ioff_FO1 = 0.00846                    %%[nA]
P_stat_FO1 = 1.152237e-7      %%[W]
P_dyn_FO1 = 1.309179e-6       %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 2 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO2 = 51.81217          %%[nA]
Ioff_FO2 = 0.00799                    %%[nA]
P_stat_FO2 = 1.088223e-7      %%[W]
P_dyn_FO2 = 1.309179e-6       %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 3 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO3= 48.76440                   %%[nA]
Ioff_FO3 = 0.00752                    %%[nA]
P_stat_FO3 = 1.02421e-7       %%[W]
P_dyn_FO3 = 1.242597e-6       %%[W]
%%%%%%%%%%%%%%%%%%%%%% Fan Out = 4 %%%%%%%%%%%%%%%%%%%%%%
Igate_FO4 = 48.76440          %%[nA]
Ioff_FO4 = 0.00752                    %%[nA]
P_stat_FO4 = 1.02421e-7       %%[W]
P_dyn_FO4 = 1.242597e-6       %%[W]
```

As can be seen from the previous analysis, different fan out type of NAND2 are available with the respective values of static and dynamic power consumption. This will allow us to built our multiplexers by using different type of gates, reducing the overall occupation area. In particular, to build our multiplexers, we will use NAND gates with fan-out equal to 1, 2 and 4.

Chapter 2

Power consumption

In this chapter are presented the calculations performed for the power consumption in CMOS multiplexers. By using the application TAMTAMS web, it was possible to make a first analysis at system level of several parameters concerning CMOS NAND2 gates employed.

2.1 Theoretical analysis

Static Power is generated by the contribution of two different leakage currents: subthreshold current and gate current. Static power is considered as the product between the supply voltage (V_{dd}) and the leakage currents probabilistic mean value.

Dynamic Power is described by the following expression

$$P_{DYN} = f_{ck} V_{DD}^2 E_{SW} C_L,$$

where contributes are:

V_{DD} voltage supply. Dynamic power consumption has a quadratic relation with it, so to reach a further power dissipation decreasing we should act on it;

f_{ck} clock frequency. Faster is the device behaviour, higher is the consumption;

E_{SW} switching activity. It is the number of transition per period;

C_L load capacitance. Is the load that the device has to drive.

To compute the dynamic power consumption, the basic MUX unit has been studied and the worst case has been selected: the maximum switching activity has been obtained analyzing the following truth table where the starting configuration is assumed as $A = B = 0$ and $S = 1$:

Table 2.1: Truth table for transition analysis

A	**B**	**S**	**Y**	**Transitions**
0	0	0	0	1
0	0	1	0	0
0	1	0	0	2
0	1	1	1	2
1	0	0	1	3
1	0	1	0	0
1	1	0	1	3
1	1	1	1	2

We have maximum switching activity when $A = 1$, B doesn't care, $S = 0$. This configuration determines that three gates (2 NAND2 and 1 inverter) output commute at the same time (see figure 2.1). In order to compute the dynamic power consumption we have taken into account only the worst case condition, considering for every elementary mux structure the consumption associated to the commutation of 2 NAND2 and 1 inverter.

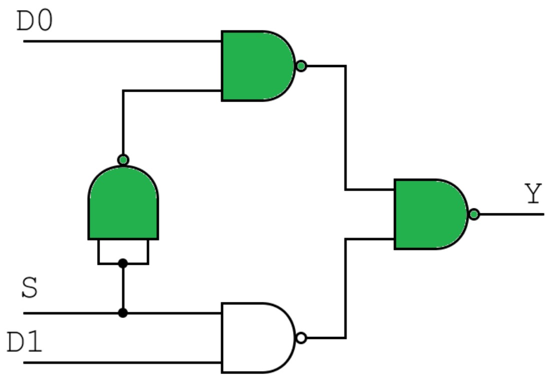


Figure 2.1: Basic MUX structure. In green are represented gates defining the maximum switching activity.

2.1.1 Models

Here is presented the models employed in the power evaluation. To represent the characteristics of each mux circuit, we used the following notation:

X stands for configuration number input;

W describes the parallelism, number of bit, that compose a single mux input information.

Static power model To compute the total amount of static power, it is necessary to consider the single NAND2 static power consumption, obtained with TAMTAMS analysis. This value has to be multiplied by the total number of gates in the configuration under exam. For instance,a basic 2-to-1 mux with $W = 1$, is obtained from a total amount of 4 NAND2 gates with fan out of 1 (FO1), according to figure 2.1. Generalizing for a mux with number of input X and parallelism W, it is possible to express a general equation describing the amount N_{gates} of NAND2 needed.

$$N_{gates} = \underbrace{3W\left(X-1\right)}_{\text{number of NAND2}} + \underbrace{\frac{W}{4}\left(X-1\right).}_{\text{number of NAND2 to build inverters}} \tag{2.1}$$

Moreover, the term expressing the number of NAND inverter, gives information about the fan-out of inverters in the circuit. Let's see three examples of how this equation work.
Example 1
Consider a mux 4-to-1 with W=2.
The amount of logic NAND2 needed are: $3W\left(X-1\right) = 18$. The amount of NAND2 inverters are: $W/4\left(X-1\right) = 1.5 = \underbrace{1}_{\text{one NAND2 FO4}} + \underbrace{0.5}_{\text{one NAND2 FO2}}$
Example 2
Consider a mux 8-to-1 with W=2.
The amount of logic NAND2 needed are: $3W\left(X-1\right) = 42$. The amount of NAND2 inverters are: $W/4\left(X-1\right) = 3.5 = \underbrace{3}_{\text{tree NAND2 FO4}} + \underbrace{0.5}_{\text{one NAND2 FO2}}$
Example 3
Consider a mux 8-to-1 with W=1.
The amount of logic NAND2 needed are: $3W\left(X-1\right) = 21$. The amount of NAND2 inverters are: $W/4\left(X-1\right) = 1.75 = \underbrace{1}_{\text{one NAND2 FO4}} + \underbrace{0.5}_{\text{one NAND2 FO2}} + \underbrace{0.25}_{\text{one NAND2 FO1}}$
In the end, to obtain the power values for static consumptions, numbers obtained with this model have been multiplied with the NAND2 static power dissipation values taken from TAMTAMS analysis.

Dynamic power model From considerations derived from the truth table with the analysis of transitions in a single 2-to-1 mux, presented at the beginning of this chapter, the mathematical expression representing maximum number of gate commutations in a generic X-to-1 mux with parallelism W is

$$N_{transitions} = \underbrace{2W\left(X-1\right)}_{\text{wort case NAND2 transitions}} + \underbrace{\frac{W}{4}\left(X-1\right).}_{\text{wort case inverter transitions}} \tag{2.2}$$

As always, the term expressing the number of NAND inverter, gives information about the fan-out of inverters commuting in the circuit. In order to obtain the power values for the dynamic consumptions, numbers obtained with this model have been multiplied with the NAND2 dynamic power dissipation values taken from TAMTAMS analysis.

2.2 Simulation results

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
2	1	0.00081W	7.133e-6W	8.076e-6 W	2.717e-6W	4.609e-7W	3.927e-6W

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
2	8	0.00524W	4.231e-5W	5.207e-5W	1.612e-5W	2.970e-6W	2.343e-5W
2	16	0.0105W	8.462e-5W	0.00010W	3.224e-5W	5.940e-6W	4.686e-5W
2	32	0.02097W	0.00017W	0.00021W	6.448e-5W	1.188e-5W	9.373e-5W
2	64	0.04194W	0.00034W	0.00042W	0.00013W	2.376e-5W	0.00019W

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
16	8	0.0786W	0.0006W	0.0008W	0.0002W	4.455e-5W	0.0003W
16	16	0.1573W	0.0013W	0.0016W	0.0005W	8.9106e-5W	0.0007W
16	32	0.3145W	0.0025W	0.0031W	0.0010W	0.0002W	0.0014W
16	64	0.6290W	0.0051W	0.0062W	0.0019W	0.0004W	0.0028W

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
32	8	0.1625W	0.0013W	0.0016W	0.0005W	9.208e-5W	0.0007W
32	16	0.3250W	0.0026W	0.0032W	0.0010W	0.0002W	0.0015W
32	32	0.6500W	0.0052W	0.0065W	0.0020W	0.0004W	0.0029W
32	64	1.3000W	0.0105W	0.0129W	0.0040W	0.0007W	0.0058W

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
64	8	0.3302W	0.0027W	0.0033W	0.0010W	0.0002W	0.0015W
64	16	0.6605W	0.0053W	0.0066W	0.0020W	0.0004W	0.0029W
64	32	1.3210W	0.0107W	0.0131W	0.0041W	0.0007W	0.0059W
64	64	2.6420W	0.0213W	0.0262W	0.0081W	0.0015W	0.01181W

Configuration		**HP_2010**		**LOP_2010**		**LSTP_2010**	
Input	*Word*	*Static*	*Dynamic*	*Static*	*Dynamic*	*Static*	*Dynamic*
128	8	0.6657W	0.0054W	0.0066W	0.0020W	0.0004W	0.0030W
128	16	1.3315W	0.0107W	0.0132W	0.0041W	0.0007W	0.0060W
128	32	2.6630W	0.0215W	0.0264W	0.0082W	0.0015W	0.0119W
128	64	5.3259W	0.0430W	0.0529W	0.0164W	0.0030W	0.0238W

2.3 Discussion

Static Power: *HP* technology has high static power value consumption due to the necessity to reach faster commutation speed compared to the others.
LSTP technology, instead, has can be seen from the results, has been designed to reach the opposite situation (lowest static power consumption).
In the end, *LOP* technology is a middle way solution because to reach lowest dynamic power consumption the commutation can't be quite fast and the threshold voltage can't be so high (because it needs higher supply voltage).

Dynamic Power: From the results can be seen that *HP* technology reaches highest dynamic power consumption, this is due to their purpose to achieve the fastest commutation velocity.
LOP technology results shows that the theoretical assumption is correct, so its values are the lowest ones.
LSTP technology is a middle way solution because to activate the gate the supply voltage needed is the higher one, due to its higher threshold voltage.

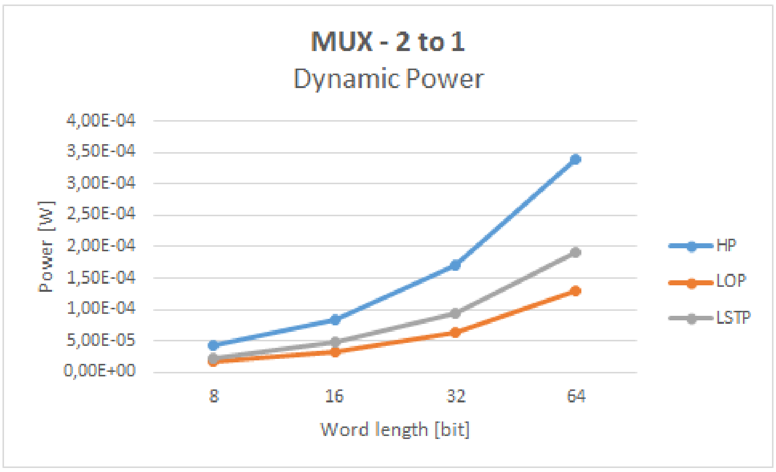


Figure 2.2:

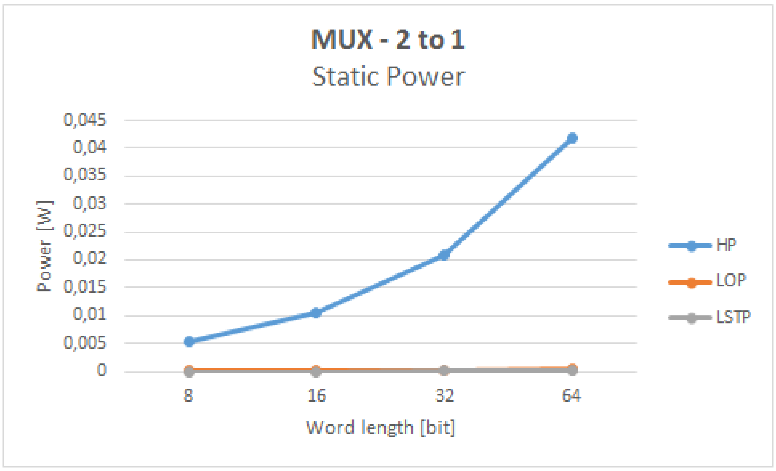


Figure 2.3:

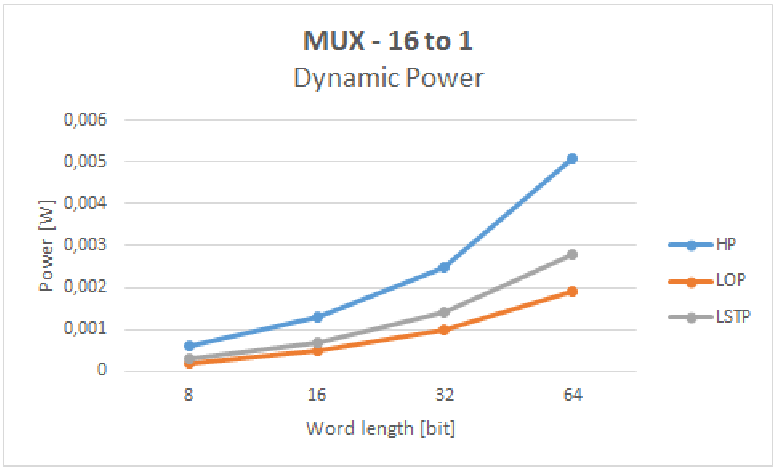


Figure 2.4:

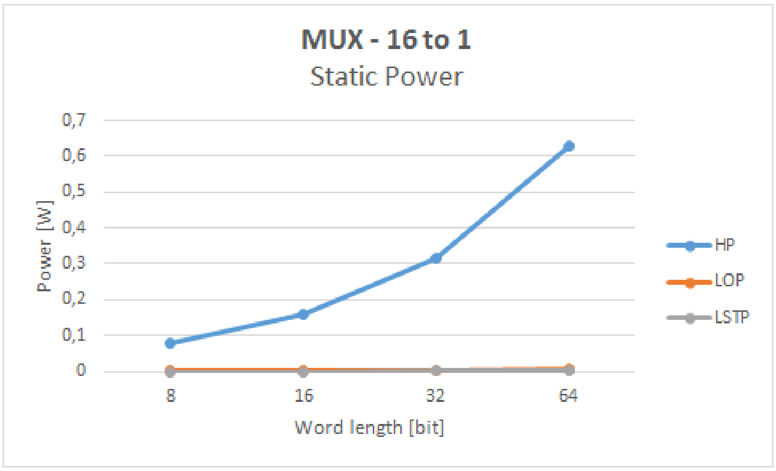


Figure 2.5:

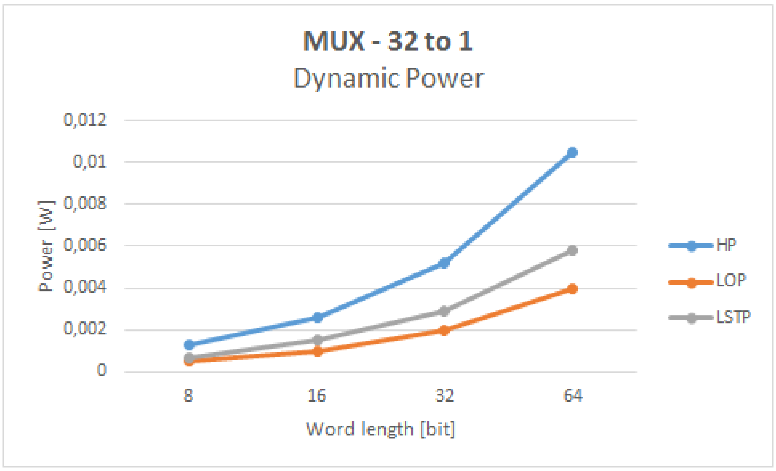


Figure 2.6:

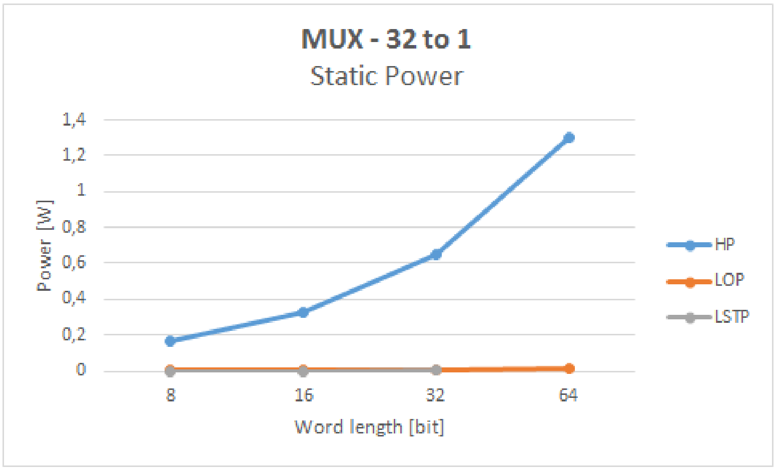


Figure 2.7:

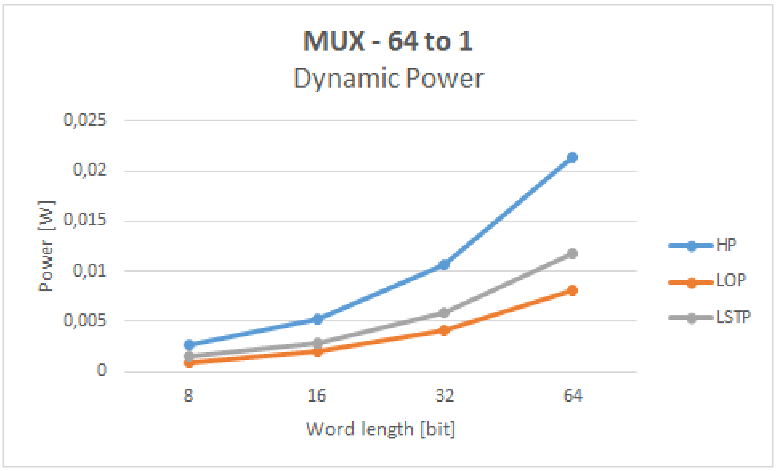


Figure 2.8:

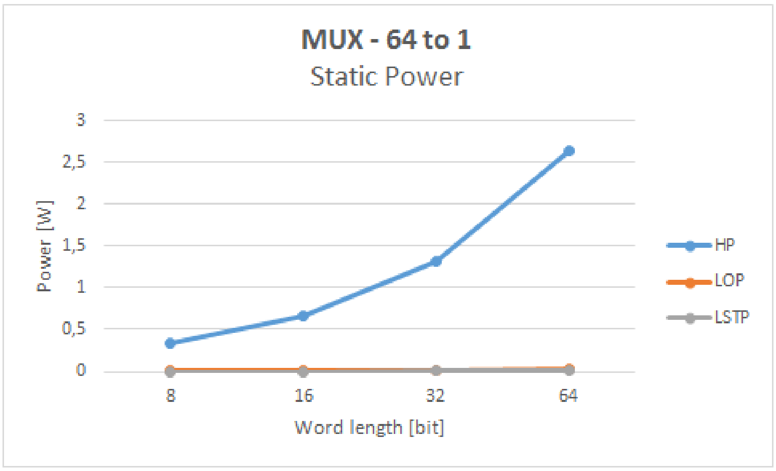


Figure 2.9:

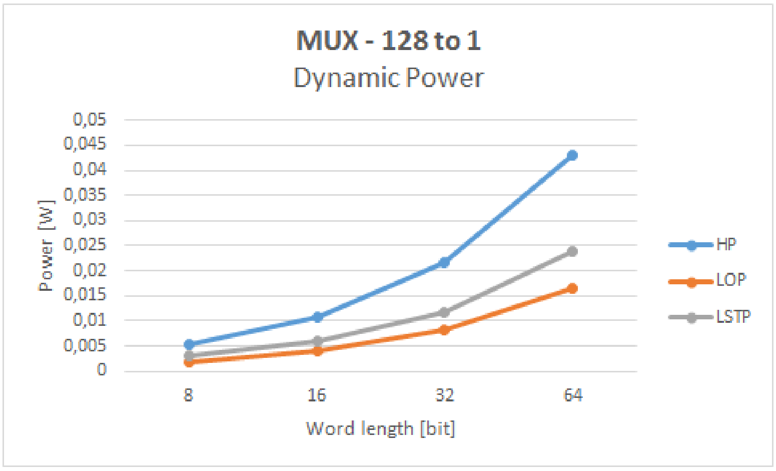


Figure 2.10:

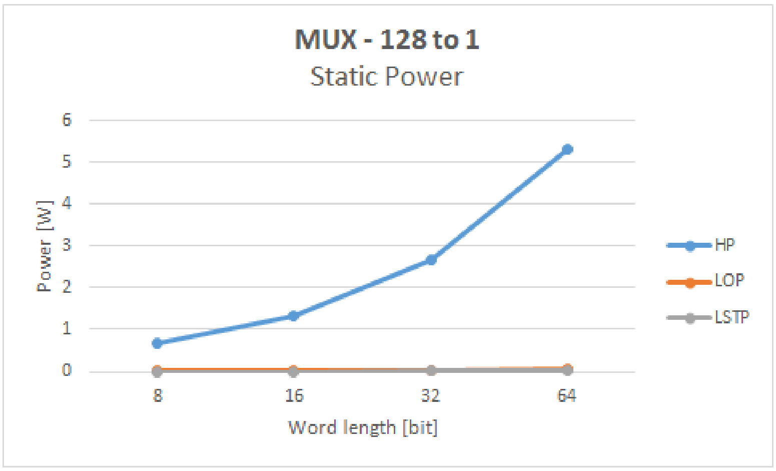


Figure 2.11:

2.4 Matlab implementation

In the table below are reported the input data.

Quantity name	Description	u.m. (S.I.)	Variable name
X	Number of input (is a power of 2)	/	X
W	Parallelism (is a power of 2)	/	Word
$P_{static}NANDFO1$	Static power NAND FO1	W	P_stat_FO1
$P_{static}NANDFO2$	Static power NAND FO2	W	P_stat_FO2
$P_{static}NANDFO4$	Static power NAND FO4	W	P_stat_FO4
$P_{dyn}NANDFO1$	Dynamic power NAND FO1	W	P_dyn_FO1
$P_{dyn}NANDFO2$	Dynamic power NAND FO2	W	P_dyn_FO2
$P_{dyn}NANDFO4$	Dynamic power NAND FO4	W	P_dyn_FO4

Table 2.2: Input data

In the table below are reported the output data.

Quantity name	Description	u.m. (S.I.)	Variable name
P_{static}	Static power MUX	W	P_stat
$P_{dynamic}$	Dynamic power MUX	W	P_dyn

Table 2.3: Input data

2.4.1 Main code

The program asks the user to select the technology workspace needed and requires to specify the number of inputs and the parallelism of the MUX. The programs needs that both number of inputs and parallelism are power of two.

```
%MUX static and dynamic power consumption
close all
clear all
clc

%open workspace of the selected technology
prompt = 'Select␣"1"␣if␣HP2010,␣"2"␣if␣LOP2010,␣"3"␣if␣LSTP2010:␣␣␣';
selection = input(prompt)
if selection ==1
    load('HP_2010.mat')
elseif selection == 2
    load('LOP_2010.mat')
```

```
elseif selection == 3
    load('LSTP_2010.mat')
end

prompt = 'Select␣number␣of␣mux␣input:␣␣␣␣␣␣';
%you can select only power of 2
X = input(prompt)

prompt = 'Select␣parallelism:␣␣␣␣␣␣';
%you can select only power of 2
Word = input(prompt)


N_gates1 = 3*Word*(X-1);
B = Word*(X-1)/4;
cont_FO4 = floor(B); %number of NAND2 FO4
decimal = B - floor(B);

cont_FO2= 0; %number of NAND2 FO2
cont_FO1= 0; %number of NAND2 FO1

if (decimal == 0.75)
    cont_FO2= 1;
    cont_FO1= 1;
elseif (decimal == 0.50)
    cont_FO2= 1;
elseif (decimal == 0.25)
    cont_FO1= 1;
end

%Number of Nand gates
N_NAND2_FO1 = N_gates1 + cont_FO1;
N_NAND2_FO2 = cont_FO2;
N_NAND2_FO4 = cont_FO4;

%Power evalutaion [W]
P_stat = N_NAND2_FO1*P_stat_FO1 + N_NAND2_FO2*P_stat_FO2 + ...
    N_NAND2_FO4*P_stat_FO4

N_NAND2_FO1 = 2*Word*(X-1) + cont_FO1;
```

```
P_dyn = N_NAND2_FO1*P_dyn_FO1 + N_NAND2_FO2*P_dyn_FO2 + ...
    N_NAND2_FO4*P_dyn_FO4
```

Chapter 3

MUX time propagation delay

In this chapter are presented the time delay calculations for all the previously showed multiplexers. Calculations have been performed using a ”dividi et impera” approach, relying on the possibility to subdivide advanced multiplexer structures in 2-to-1 NAND2 based mux units. An RC delay model have been exploited, where the time constant of a single NAND gate is given by the total capacitance in the output node times an effective resistance depending from the voltage source V_{DD} and a ON current I_{ON}. In order to carry out quick estimations, we have considered a smart approach presented during lectures, referring on NAND2 statistical loads. Calculations have been done for CMOS MUX in HP, LOP and LSTP 2010 technologies.

3.1 Theoretical analysis

According to the Roadmap, the average output of a NAND2 gate is fan-out of 4 made by inverters. It is therefore convenient to consider the total load capacitance of a NAND2 gate, as a sum of three different contributions

$$C_{ND2} = C_{FO4} + C_{metal} + C_J, \tag{3.1}$$

where C_{FO4} is an fan-out of 4 ideal statistical load, given by four inverters. The term C_{metal} accounts for capacitive contributions due to interconnects and C_J accounts for junction capacitances. Overhead contributions of metallic interconnects and junctions have been estimated to be around 15% of the ideal C_{FO4}. It is possible to rewrite the previous expression in a more compact form

$$C_{ND2} = M\ C_{FO4}, \tag{3.2}$$

where M has been fixed to 1.5 and plays the role of parasitic multiplier. According to theory, C_{FO4} is given by four times the value assumed by a simpler C_{FO1} that can be written as $C_{FO1} = C_{gate_n}\,(1 + W_p/W_n) = 2.29\;C_{gate_n}$. The nMOS gate capacitance can be expressed at first order by three different contributions involving the oxide capacitance, drain and source extension capacitances and a capacitance term modelling fringe effect. The capacitive contribution duo to the extensions has been considered 20% of the oxide capacitance. It is possible therefore to rewrite a plausible representation of the NAND2 load capacitance as a function of several CMOS technological parameters, taking into account all the three previously presented physical effects

$$C_{ND2} = 13.74\left(\frac{\epsilon_{OX}}{t_{OX}} l_{ch} + 0.2\frac{\epsilon_{OX}}{t_{OX}} l_{ch} + C_{fringe}\right) \tag{3.3}$$

Once the capacitances are known, it is necessary to evaluate the effective resistance. It can be represented as the ratio between the voltage source V_{DD} and the ON current of a NAND2 gate ($I_{ND2_{ON}}$). In order to evaluate a worst-case situation, we consider the ON current of a NAND gate equal to the I_{DS} current of a single pMOS transistor. This value has been evaluated using the Matlab MASTAR4 model reported in Tamtams. The effective resistance is given by the ratio $R = \frac{V_{DD}}{I_{ON}}$. The final value of the time constant of NAND2 gate is the simple RC product

$$\tau = R\,C_{ND2}. \tag{3.4}$$

3.1.1 MUX time delay

The time delay evaluation in a multiplexer composed by a NAND tree structure follows simple rules. In order to explain in a rigorous way the employed evaluation methodology, it is necessary to focus first the attention on a elementary block MUX 2x1. The critical path in a simple 2x1 mux is composed by three nand gates (see figure 3.1) each one contributing for a unit of constant time. If we want to find the time delay of a 4x1 mux, it is necessary to consider a critical path defined by two 2x1 multiplexers connected in series. By extending this analysis to more advanced multiplexers, it is possible to write a generic expression for the total time delay, valid only in the simplified case of multiplexers showing a nand tree structure

$$t = \tau\left[2\log_2(X) + 1\right] \tag{3.5}$$

Where X is the total number of input and is a power of two. Please notice that this general expression is unaffected by the parallelism W of the system. In fact, increasing the parallelism will just affect the number of logic gates connected in parallel, without changing

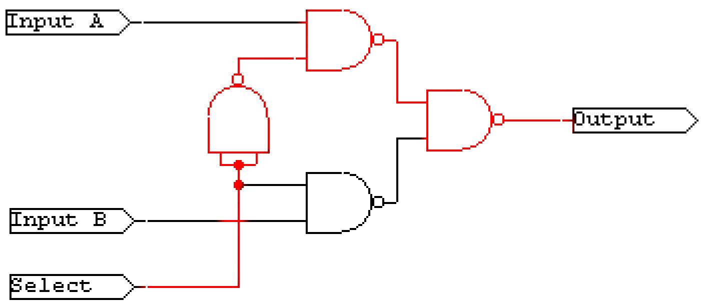


Figure 3.1: Representation of a 2x1 MUX. In red is highlighted the critical path.

the critical path. Moreover, the simplicity of the expression is direct consequence of the choice to treat each nand gate in a statistical way, according to the roadmap recommendation.

3.2 Simulation results

Here are reported the time propagation delay numerical results of several multiplexers built according to the technologies HP, LSTP, LOP 2010. The simulations have been done for an arbitrary channel width $W_n = 270nm$.

Number input	HP	LOP	LSTP
2	35.58 ps	59.95 ps	59.96 ps
8	83 ps	130.6 ps	177.2ps
16	106.73 ps	167.9 ps	227.87 ps
32	130.44 ps	205.15 ps	278.5 ps
64	154.16 ps	242.44 ps	329.14 ps
128	177.88 ps	279.74 ps	379.78 ps

3.2.1 Discussion

In figure 3.2 are reported the simulation values together in a unique dispersion plot. As expected, the time delay of a high performance multiplexer shows smallest value with respect the other technologies. LSTP multiplexers are the slowest devices.

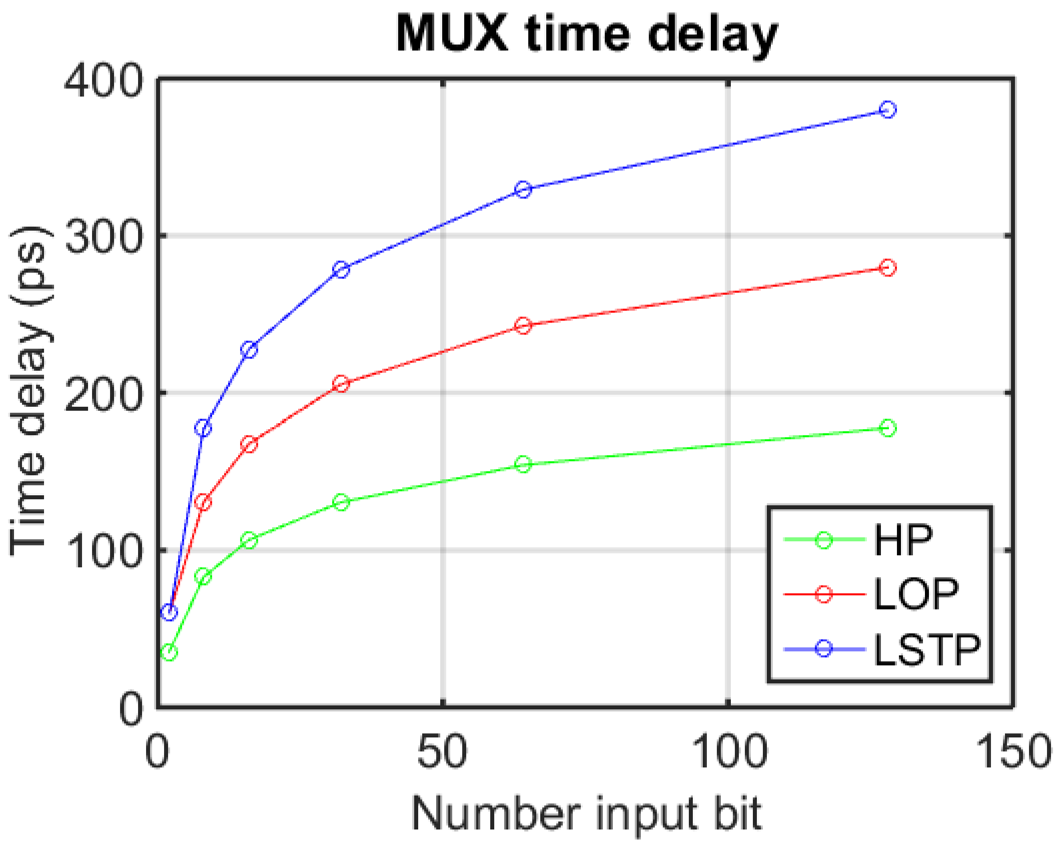


Figure 3.2: Time delay simulation results for 8, 16, 24, 32, 64, 128 inputs. Lines connecting each dot are just a graphical support.

3.3 Matlab implementation

In the table below are reported the input data of the main script.

Quantity name	Description	u.m. (S.I.)	Variable name
X	Number of input of MUX(is a power of 2)	/	X
$C_{fringe}\ HP$	Fringe capacitance MOS HP	F/nm	fring_capHP2010
$C_{fringe}\ LOP$	Fringe capacitance MOS LOP	F/nm	fring_capLOP2010
$C_{fringe}\ LSTP$	Fringe capacitance MOS LSTP	F/nm	fring_capLSTP2010
$T_{ox}\ HP2010$	Oxide thickness MOS HP2010	nm	Tox_HP2010
$T_{ox}\ LOP2010$	Oxide thickness MOS LOP2010	nm	Tox_LOP2010
$T_{ox}\ LSTP2010$	Oxide thickness MOS LSTP2010	nm	Tox_LSTP2010
$l_{ch}\ HP2010$	channel length MOS HP2010	nm	L_HP2010
$l_{ch}\ LOP2010$	channel length MOS LOP2010	nm	L_LOP2010
$l_{ch}\ LSTP2010$	channel length	nm	L_LSTP2010
$V_{DD}\ HP$	Voltage source HP technology	V	Vdd_HP2010
$V_{DD}\ LOP$	Voltage source LOP technology	V	Vdd_LOP2010
$V_{DD}\ LSTP$	Voltage source LSTP technology	V	Vdd_LSTP2010

Table 3.1: Input data

In the table below are reported the output data of the main script.

Quantity name	Description	u.m. (S.I.)	Variable name
$\tau\ HP2010$	Time constant NAND2 HP2010	s	Tsp_HP_nd2
$\tau\ LOP2010$	Time constant NAND2 LOP2010	s	Tsp_LOP_nd2
$\tau\ LSTP2010$	Time constant NAND2 LSTP2010	s	Tsp_LSTP_nd2
$t\ HP2010$	Time delay mux HP2010	s	tautotalHP
$t\ LOP2010$	Time delay mux LOP2010	s	tautotalLOP
$t\ LSTP2010$	Time delay mux LSTP2010	s	tautotalLSTP

Table 3.2: Output data

3.3.1 Main time delay

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  MAIN Nand2 time delay
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

```

```
%Permittivity
eps_0 = 8.854187e-12 *1e-9; %F/nm
eps_SiO2 = 3.9*eps_0; %F/nm

%Fringe capacitances
fring_capHP2010 = 1.5e-16*1e-3; %F/nm
fring_capLOP2010 = 2.2e-16*1e-3;
fring_capLSTP2010 = 2.4e-16*1e-3;

%Oxide thickness
Tox_HP2010 = 0.65; %nm
Tox_LOP2010 = 0.9;
Tox_LSTP2010 = 1.4;

%Channel length
L_HP2010 = 18; %nm
L_LOP2010 = 22;
L_LSTP2010 = 28;

%Voltage source
VDD_HP2010 = 1; %V
VDD_LOP2010 = 0.7;
VDD_LSTP2010 = 1.1;

prompt = 'Specify␣the␣nMOS␣channel␣width␣Wn␣(nm):␣␣';
Wn = input(prompt)

%Total capacitance on output node of the Nand2 gate
C_nd2_HP = 13.74*(eps_SiO2*L_HP2010/Tox_HP2010 + fring_capHP2010 + ...
 0.2*eps_SiO2*L_HP2010/Tox_HP2010)*Wn;
C_nd2_LOP = 13.74*(eps_SiO2*L_LOP2010/Tox_LOP2010+fring_capLOP2010+ ...
0.2*eps_SiO2*L_LOP2010/Tox_LOP2010)*Wn;
C_nd2_LSTP=13.74*(eps_SiO2*L_LSTP2010/Tox_LSTP2010+fring_capLSTP2010 + ...
0.2*eps_SiO2*L_LSTP2010/Tox_LSTP2010)*Wn;

%Mastar model implementation
[Vth_nHP,Vth_pHP, Ioff_nHP, Ioff_pHP, Igate_nHP,...
Igate_pHP]= Mastar4_Vth_Ioff_IgHP2010();
[I_nMOSHP, I_pMOSHP]= Ion_Mastar_modelHP2010(Vth_nHP); %uA/um

```

```
[Vth_nLOP,Vth_pLOP, Ioff_nLOP, Ioff_pLOP, ...
Igate_nLOP, Igate_pLOP]= Mastar4_Vth_Ioff_IgLOP2010();
[I_nMOSLOP, I_pMOSLOP]= Ion_Mastar_modelLOP2010(Vth_nLOP); %uA/um

[Vth_nLSTP,Vth_pLSTP, Ioff_nLSTP, Ioff_pLSTP, ...
Igate_nLSTP, Igate_pLSTP]= Mastar4_Vth_Ioff_IgLSTP2010();
[I_nMOSLSTP, I_pMOSLSTP]= Ion_Mastar_modelLSTP2010(Vth_nLSTP); %uA/um

%NAND2 delay evaluation
Wp = 1.29*Wn;
Tdp_HP_nd2 =C_nd2_HP*VDD_HP2010/(I_pMOSHP*Wp*1e-3*1e-6); %s
Tdp_LOP_nd2 =C_nd2_LOP*VDD_LOP2010/(I_pMOSLOP*Wp*1e-3*1e-6); %s
Tdp_LSTP_nd2 =C_nd2_LSTP*VDD_LSTP2010/(I_pMOSLSTP*Wp*1e-3*1e-6); %s

prompt = 'Specify␣the␣number␣of␣channel␣inputs:␣␣';

X = input(prompt);
%Ninput has a mean only if is 2,4,8,16,32,64

%Time delay evaluation, s
tautotalHP =log2(X)*2*Tdp_HP_nd2+Tdp_HP_nd2 %s
tautotalLOP =log2(X)*2*Tdp_LOP_nd2+Tdp_LOP_nd2 %s
tautotalLSTP =log2(X)*2*Tdp_LSTP_nd2+Tdp_LSTP_nd2 %s
```

3.3.2 Mastar4 Vth Ioff Ig HP2010()

This is a function needed in order to evaluate the threshold voltages, the off and the gate current of MOS transistors. The most part of the code has been taken from the Matlab modules of TAMTAMS.

```
function [Vth_n,Vth_p, Ioff_n, Ioff_p, Igate_n,...
Igate_p]= Mastar4_Vth_Ioff_IgHP2010()
%Please see: https://tamtams.vlsilab.polito.it/...
%Documentation/TechnologyHTML/bulk/HP2010_dev_bul.html

%PARAMETERS
%Please, select "Model" value for the technology type
%BULK = 0; FINFET = 1; SOI = 2; GAA = 3; CNT = 4;
Model = 0;

```

```
%Physical parameters
q = 1.6021766208e-19;       % elementary charge (C)
kB = 1.3806488e-23;         % Boltzmann constant (J/K)
hbar = 1.054571800e-34;     % reduced Planck cons tant (Js)
m0 = 9.10938291e-31;        % electron mass (kg)
clight = 2.99792458e8 ;     % speed of light (m/s)
mu0 = (4* pi )*1e-7;        % vacuum magnetic permeability (H/m)
e0 = 1e-2/( clight ^2* mu0 );% vacuum electric permittivity (F/cm)
es = 11.8;                  % Silicon relative dielectric constant
esio2 = 3.9;                % Silicon oxide relative dielectric constant
Eg0 = 1.166;                % Silicon energy gap at 300K (eV)
Alpha = 4.73e-4; Beta = 636;
Temp = 300;                 % Absolute working temperature (K)
Eg = Eg0 - Alpha * Temp^2/(Temp + Beta);%Silicon energy gap (eV)
Vt = kB.*Temp./q;           %Termal voltage (V)
ni = sqrt(5.85e31*Temp^3*exp(-Eg/Vt));%Intrinsic carrier concentration(cm^-3)
Xs = 4.003;                 % Silicon affinity (eV)

%Geometrical parameters (HP 2012)
Lgate =18; %nm, The effective length of the gate.
Xj = 6.5; %nm, Source/Drain extensions length.
LDDW = 50; %nm, Doping width of source/drain extension, value used only...
%for transistors image generation.
Tox = 0.65; %nm, Physical gate oxide thickness.
Darks = 0.27; %nm, 2.5 A=0.1nm Dark space length, used in...
%Mastar threshold voltage model.
Polyd = 0; %Poly depletion length, used in Mastar threshold voltage model.
Wt = 1000; %nm, Gate width of n-MOS.
beta =1.29; %is the ratio between Wp/Wn.
gateW_p = beta*Wt; %nm, Gate width of p-MOS
gateH = 40; %nm, Gate thickness,
DopH = 30; %nm, Doping depth of source/drain
DopW = 100; %nm, Doping width of source/drain
ContW = 30; %nm, Contact width
Diff_ray = 5;%nm, Ray of annealing diffusion
Angle = 25; %deg, Pocket implantation angle

%Doping parameters (HP 2010)
Nbulk = 7.14e18; %cm^-3, Channel Doping
n_sub = 3.2e18; %cm^-3, Substrate Doping
```

```
n_plus_poly = 1e20; %cm^-3, Polysilicon n+ doping
Next = 7.819e18; %cm^-3, Extensions doping
n_source_drain = 1e20; %cm^-3, Source/Drain doping

%Electrical parameters (HP 2010)
rconst_n = 254.61; %Ohm, N-MOS access resistance
rconst_p = 254.61; %Ohm, P-MOS access resistance
Fring_cap = 1.5e-16; %F/um, Fringe capacitance per unit length
Ith = 5e-7; %A, Id at the threshold voltage (Vth_off)
Phi_m = 4.15; %V Built-in potential
Kfield = 1; %Effective electric field reduction

%Scaling factors (HP 2010)
Gamma = 0.7; % Scaling factor for lateral diffusion
Zeta = 0.8; %Scaling factor for drain induced lowering barrier (DIBL)
Zeta2 = 0.64; %Scaling factor for short channel effect (SCE)

%Power supply parameters (HP 2010)
Vdd = 1; %V Operation voltage (gate and drain voltages)

%Other parameters(HP 2010)
Cpoches = 5.2e13;
Rp = 10; %nm
Delta_rp = 6; %nm
Delta_rl = 3; %nm
ActivePkt = 0;


%THRESHOLD VOLTAGE, MASTAR 4 MODEL
% https://tamtams.vlsilab.polito.it/Documentation/ ...
ModulesHTML/vth/vth_mas_complete_bul.html

ni_temp = ni /3.79; %Fitting from Mastar
Phi_f = Vt*log(Nbulk/ni_temp); %Surface potential (V)
Vfb= Phi_m -(Xs + Eg/2 + Phi_f);
Vbi=Vt*log(Nbulk*Next/ni_temp^2);
Qdep=sqrt(2*es*e0*q*Nbulk*(2*Phi_f));
Tox_el=Tox*3.9/esio2 + Darks/10 + Polyd/10; %nm, Electrical oxide thickness
% %considering the effect of Dark Space and Polysilicon depletion
Cox=esio2*e0/(Tox_el*1e-7); %Oxide capacitance evaluation [F/cm^2]
```

```matlab
Vtlong=Vfb+2*Phi_f+Qdep/Cox; %Long channel threshold voltage
Lel=Lgate-Gamma*Xj; %Channel electric length [nm]

%Depletion depth calculation:
Tdepbulk=1e7*Qdep/(q*Nbulk);
% for BULK transistors
if(Model==0)
Tdep=Tdepbulk;
% for Double Gate transistors
elseif(Model==1)
Tdep=Xj/2;
Xj=Xj/2;
% for SOI transistors
else if (Tdepbulk<Xj)
Tdep=Tdepbulk;
else
Tdep=Xj;
end
end

EI=(1+(Xj/Lel)^2)*Tox_el*Tdep/(Lel)^2;
% Short Channel Effect (SCE) [V]
SCE=es/esio2*Zeta2*Vbi*EI;
% Drain Induced Barrier Lowering (DIBL) [V]
DIBL=es/esio2*Zeta*EI*Vdd;
Vth=Vtlong-SCE-DIBL;
Vth_n = Vth;
Vth_p = -Vth;


%SUBTHRESHOLD CURRENT, MASTAR 4 MODEL
% https://tamtams.vlsilab.polito.it/Documentation/...
ModulesHTML/ioff/Ioff_mas4_bul.html


Npocket=0.5*(Cpoches/((Rp+2*Delta_rp)*10^-7));
Lpocket=(Rp+2*Delta_rp)*sin(Angle*pi/180)+ ...
2*Delta_rl*cos(Angle*pi/180)-(Gamma*Xj)/2;
Lel = Lgate - Gamma * Xj; %Electrical channel length [nm]
Lmin=min(Lel,Lpocket); %Effective channel length
```

```
Nch = Nbulk + 2 * Npocket * (Lmin/Lel); %Channel doping
if (ActivePkt==0)
Ith_new = Ith*Wt/Lel;
else
Ith_new = Ith*Wt/Lel* 8 * 10^8 * Nch^(-0.4865);
end;

Qdep = sqrt(2*es*e0*q*Nbulk*(2*Phi_f)); %Depletion charge evaluation

%Depletion depth calculation
Tdepbulk=1e7*Qdep/(q*Nbulk);
% for BULK transistors
if(Model==0)
Tdep=Tdepbulk;
% for Double Gate transistors
elseif(Model==1)
Tdep=Xj/2;
Xj=Xj/2;
% for SOI transistors
else if (Tdepbulk<Xj)
Tdep=Tdepbulk;
else
Tdep=Xj;
end
end

SS=Vt*log(10)*(1+((es/esio2)*Kfield*(Tox_el/Tdepbulk))); %V/dec
Ioff_n=Ith_new*10^(-Vth/SS)*10^9;  %nA/um
Ioff_p=Ith_new*10^(-Vth/SS)*10^9;  %nA/um


% GATE CURRENT, MASTAR 4 MODEL
% https://tamtams.vlsilab.polito.it/Documentation/...
ModulesHTML/igate/Igate_Mastar4_bul.html

%a, b, c, d parameters:
ag = 1.44e5;    % A/cm^2
bg = -4.02;           % V^-2
cg = 13.05;           % V^-1
dg = 1.17;            % Angstrom^-1
```

```
%Gate current density
Jgate_n = ag*exp(bg*Vdd^2 + cg*Vdd)*exp(-dg*Tox*10)*10; % nA/um^2
Jgate_p = ag*exp(bg*Vdd^2 + cg*Vdd)*exp(-dg*Tox*10)*10;

Igate_n= Jgate_n*Lgate*1e-3; % [nA/um]
Igate_p= Jgate_p*Lgate*1e-3; % [nA/um]

if (Model==1)
Igate_n=2*Igate_n;
Igate_p=2*Igate_p;
end
return
```

The other two functions Mastar4_Vth_Ioff_IgLOP2010() and Mastar4_Vth_Ioff_IgLSTP2010() are not reported here, since they change just for what concern constants. For more details please see the original Matlab scripts.

3.3.3 Ion Master model HP 2010()

This is a function needed in order to evaluate the I_{ON} of MOS transistors. The most part of the code has been taken from the Matlab modules of TAMTAMS.

```
function [Ion_n, Ion_p]= Ion_Mastar_modelHP2010(Vth)

% PARAMETERS
%Please, select "Model" value for the technology type
%BULK = 0; FINFET = 1; SOI = 2; GAA = 3; CNT = 4;
Model = 0;

%Physical parameters
q = 1.6021766208e-19;      % elementary charge (C)
kB = 1.3806488e-23;        % Boltzmann constant (J/K)
hbar = 1.054571800e-34;    % reduced Planck cons tant (Js)
m0 = 9.10938291e-31;       % electron mass (kg)
clight = 2.99792458e8 ;    % speed of light (m/s)
mu0 = (4* pi )*1e-7;       % vacuum magnetic permeability (H/m)
e0 = 1e-2/( clight ^2* mu0 ); % vacuum electric permittivity (F/cm)
es = 11.8;                 % Silicon relative dielectric constant
esio2 = 3.9;               % Silicon oxide relative dielectric constant
Eg0 = 1.166;               % Silicon energy gap at 300K (eV)
```

```
Alpha = 4.73e-4; Beta = 636;
Temp = 300;                 % Absolute working temperature (K)
Eg = Eg0 - Alpha * Temp^2/(Temp + Beta);%Silicon energy gap (eV)
Vt = kB.*Temp./q;           %Termal voltage (V)
ni = sqrt(5.85e31*Temp^3*exp(-Eg/Vt)); %Intrinsic carrier concentration(cm^-3)
Xs = 4.003;                 % Silicon affinity (eV)

%Geometrical parameters (HP 2010)
Lgate =18; %nm, The effective length of the gate.
Xj = 6.5; %nm, Source/Drain extensions length.
LDDW = 50; %nm, Doping width of source/drain extension
Tox = 0.65; %nm, Physical gate oxide thickness.
Darks = 0.27; %nm, 2.5 A=0.1nm Dark space length
Polyd = 0; %Poly depletion length
Wt = 1000; %nm, Gate width of n-MOS.
beta =1.29; %is the ratio between Wp/Wn.
gateW_p = beta*Wt; %nm, Gate width of p-MOS
gateH = 40; %nm, Gate thickness
DopH = 30; %nm, Doping depth of source/drain
DopW = 100; %nm, Doping width of source/drain
ContW = 30; %nm, Contact width
Diff_ray = 5;%nm, Ray of annealing diffusion
Angle = 25; %deg, Pocket implantation angle

%Doping parameters (HP 2010)
Nbulk = 7.14e18; %cm^-3, Channel Doping
n_sub = 3.2e18; %cm^-3, Substrate Doping
n_plus_poly = 1e20; %cm^-3, Polysilicon n+ doping
Next = 7.819e18; %cm^-3, Extensions doping
n_source_drain = 1e20; %cm^-3, Source/Drain doping

%Electrical parameters (HP 2010)
rconst_n = 254.61; %Ohm, N-MOS access resistance
rconst_p = 254.61; %Ohm, P-MOS access resistance
Fring_cap = 1.5e-16; %F/um, Fringe capacitance per unit length
Ith = 5e-7; %A, Id at the threshold voltage (Vth_off)
Phi_m = 4.15; %V Built-in potential
Kfield = 1; %Effective electric field reduction

%Scaling factors (HP 2010)
```

```
Gamma = 0.7; % Scaling factor for lateral diffusion
Zeta = 0.8; %Scaling factor for drain induced lowering barrier (DIBL)
Zeta2 = 0.64; %Scaling factor for short channel effect (SCE)

%Power supply parameters (HP 2010)
Vdd = 1; %V Operation voltage (gate and drain voltages)

%Other parameters(HP 2010)
Cpoches = 5.2e13;
Rp = 10; %nm
Delta_rp = 6; %nm
Delta_rl = 3; %nm
ActivePkt = 0;

ni_temp = ni /3.79; %Fitting from Mastar
Phi_f = Vt*log(Nbulk/ni_temp); %Surface potential (V)
Vfb= Phi_m -(Xs + Eg/2 + Phi_f);
Vbi=Vt*log(Nbulk*Next/ni_temp^2);
Qdep=sqrt(2*es*e0*q*Nbulk*(2*Phi_f));
Tox_el=Tox*3.9/esio2 + Darks/10 + Polyd/10; %nm, Electrical oxide thickness
% %considering the effect of Dark Space and Polysilicon depletion
Cox=esio2*e0/(Tox_el*1e-7); %Oxide capacitance evaluation [F/cm^2]
Vtlong=Vfb+2*Phi_f+Qdep/Cox; %Long channel threshold voltage
Lel=Lgate-Gamma*Xj; %Channel electric length [nm]

%Depletion depth calculation:
Tdepbulk=1e7*Qdep/(q*Nbulk);
% for BULK transistors
Model = 0;
if(Model==0)
Tdep=Tdepbulk;
% for Double Gate transistors
elseif(Model==1)
Tdep=Xj/2;
Xj=Xj/2;
% for SOI transistors
else if (Tdepbulk<Xj)
Tdep=Tdepbulk;
else
Tdep=Xj;
```

```
end
end
Sigma = 1;
Gamma = 0.7;
Xj = 6.5; %nm
d=Sigma*q*Nbulk*Tdep/(2*Cox*(2*Phi_f))*1e-7;

%Electrical channel length [nm]
Lel = Lgate - Gamma * Xj;

%Evaluation of the value of E critic field for...
%the electrons and holes with the approximation of...
%saturation velocity of carriers
Kvs = 1.1;
vsat = 1e7; %cm/s
%Effective electric field
%https://tamtams.vlsilab.polito.it/Documentation/ModulesHTML/...
mobility/mobility_mas_bul.html
Kfield = 1;
Eeff_n=Kfield*(1e-6*(Vdd+Vth-2*(Vfb+2*Phi_f)))/(6*Tox_el*1e-7);
Eeff_p=Kfield*(1e-6*(Vdd+2*Vth-3*(Vfb+2*Phi_f)))/(9*Tox_el*1e-7);
%Mobility in normal conditions
muac_n=330*Eeff_n^(-0.3);
muac_p=90*Eeff_p^(-0.3);
%Mobility in high electric field conditions
musr_n=1450*Eeff_n^(-2.9);
musr_p=140*Eeff_p^(-1);
%Effective mobility calculation [cm2 V-1 s-1]
Kp = 1;
mueff_n=Kp*muac_n*musr_n/(muac_n+musr_n)
mueff_p=Kp*muac_p*musr_p/(muac_p+musr_p)

%Evaluation of the value of E critic field for the electrons and holes...
 with the approximation of saturation velocity of carriers
Ecrit_n=2*Kvs*vsat/mueff_n;
Ecrit_p=2*Kvs*vsat/mueff_p;
%Evaluation of Vdsat with approssimation of high field and body ...
linearization of the Qn charge
Vdsat_n=(1/(Lel*1e-7*Ecrit_n)+(1+d)/(Vdd-Vth))^-1;
Vdsat_p=(1/(Lel*1e-7*Ecrit_p)+(1+d)/(Vdd-Vth))^-1;
```

```

%Evaluation of the saturation current without the channel resistance
Idsat0_n=0.5*mueff_n*Cox*(Wt/Lel)*(Vdd-Vth)*Vdsat_n;
Idsat0_p=0.5*mueff_p*Cox*(Wt/Lel)*(Vdd-Vth)*Vdsat_p;

%Evaluation of the Ion current by adding the contribute of channel resistance
Rconst_n = 256.61; %Ohm
Rconst_p = 256.61; %N-MOS access resistance
Ion_n=Idsat0_n/(1+(2*Rconst_n*Idsat0_n/(Vdd-Vth))- ...
Rconst_n*Idsat0_n/(Vdd-Vth+Lel*1e-7*Ecrit_n*(1+d)))*1e6; %uA/um
Ion_p=Idsat0_p/(1+(2*Rconst_p*Idsat0_p/(Vdd-Vth))- ...
Rconst_p*Idsat0_p/(Vdd-Vth+Lel*1e-7*Ecrit_p*(1+d)))*1e6; %uA/um
```

The other two functions Ion_Mastar_modelLSTP2010(Vth) and Ion_Mastar_modelLOP2010(Vth) are not reported here, since they change just for what concern constants. For more details please see the original Matlab scripts.

Chapter 4

Multiplexer area

4.1 Theoretical analysis

The area of a single transistor is estimated as the product between the length (L) and the width (W). The overall length L is the sum of the the gate and the source and drain lengths. For the symmetry of the MOS structure, we can imagine that both source and drain lengths are equal and depend from the gate length according to arbitrary design rules $L_{S/D} = \lambda L_{gate}$.

$$L = L_{gate} + 2\ L_{S/D}. \tag{4.1}$$

The width is different between the two transistors, but the ratio remains always the same ($W_p/W_n = 1.29$). In order to evaluate the occupied area of a mux, it is necessary to obtain first the area of a single Nand made with the CMOS technology. Another important aspect to take into account is the interconnections override that has been estimated to be the 15% of the logic gate area ($I_O = 0.15$). As results, the area of a single Nand2 (A_{ND2}) is:

$$A_{ND2} = 2L\ (Wn + Wp)(1 + I_O) \tag{4.2}$$

Once the occupied area of a single Nand2 is known, it is necessary to know the overall number of gates in the circuit. To do so, we can use the equation 2.1 in chapter 2, that allows to do a smart and full estimation of the number of gates in our system.

4.2 Simulation results

Here are reported several tables with the occupation area expressed in μm^2, for a nMOS with a channel width of 270nm. The program of course allows to make calculations for a customized value of channel width.

Hp Technology

	2x1	8x1	16x1	32x1	64x1	128x1
8 bit	1.9966	13.9763	29.9492	61.8950	125.7867	253.578
16 bit	3.9932	27.9526	59.8984	123.7901	251.5734	507.1400
32 bit	7.9865	55.9052	119.7569	247.5802	503.1468	1014.3
64 bit	15.9729	111.8104	239.5937	495.1604	1006.3	2028.6

Lop Technology

	2x1	8x1	16x1	32x1	64x1	128x1
8 bit	2.4403	17.0821	36.6046	75.6495	153.7393	309.9189
16 bit	4.8806	34.1643	73.2092	151.2990	307.4786	619.8378
32 bit	9.7612	68.3286	146.4184	302.5980	614.9572	1239.7
64 bit	19.5225	136.6572	292.8368	605.1960	1229.9	2479.4

Lstp Technology

	2x1	8x1	16x1	32x1	64x1	128x1
8 bit	3.1058	21.7409	64.5877	96.2812	195.6682	394.4423
16 bit	6.2117	43.4818	93.1753	192.5424	391.3364	788.8845
32 bit	12.4234	86.9634	186.3507	385.1247	782.6728	1577.8
64 bit	24.8468	173.9273	372.7013	770.2495	1565.3	3155.5

4.3 Discussion

As expected the occupied area shows a net tendency to increase as a function of the parallelism. However, the occupied area of a HP technology is generally smaller than the other technologies. And the LSTP technology shows the greatest occupied area as a function of number of the parallelism.

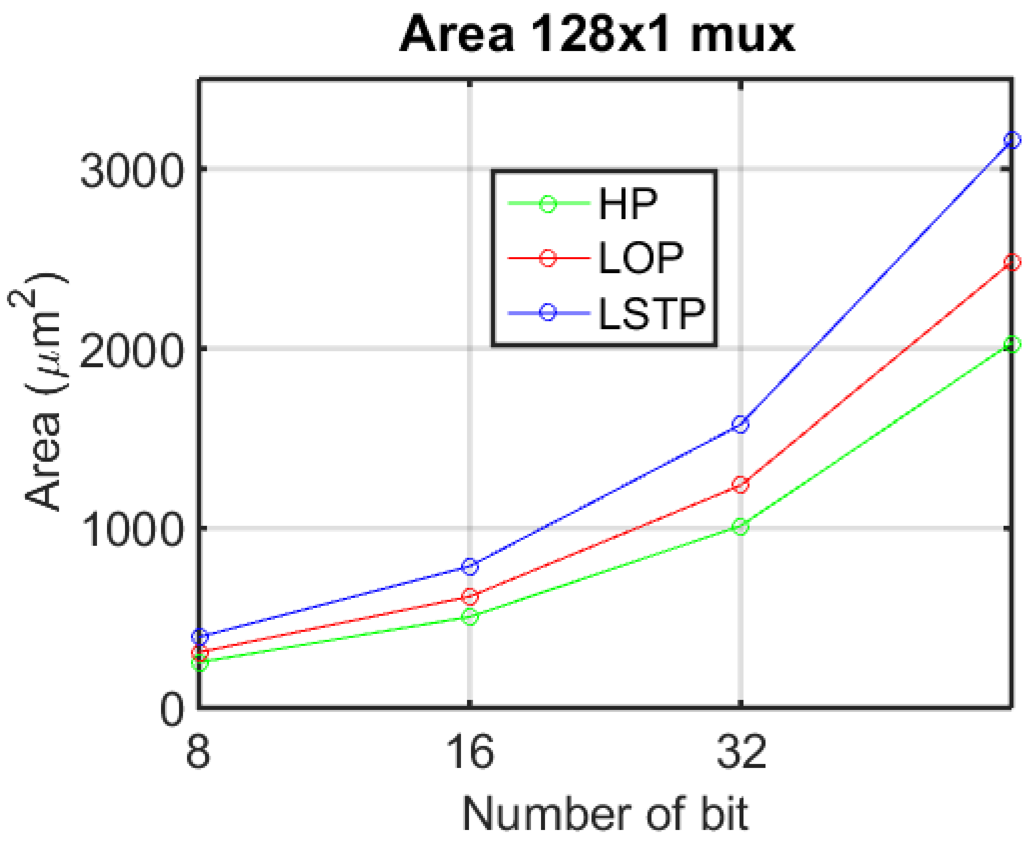


Figure 4.1: Occupied area of a mux 128x1 as a function of the parallelism

4.4 Matlab implementation

In the table below are reported the input data of the main script.

Quantity name	Description	u.m. (S.I.)	Variable name
X	Number of input (is a power of 2)	/	X
W	Parallelism (is a power of 2)	/	Word
W_n	Gate width	um	Wn
I_O	Interconnections override	/	interc_override
λ	design parameter to determine $L_{S/D}$	/	lambda
L_{gate}	Gate length HP, LOP, LSTP 2010	um	Lgate

Table 4.1: Input data

In the table below are reported the output data of the main script.

Quantity name	Description	u.m. (S.I.)	Variable name
$A_{MUX}\ HP2010$	Area MUX HP 2010	μm^2	AreaHP
$A_{MUX}\ LOP2010$	Area MUX LOP 2010	μm^2	AreaLOP
$A_{MUX}\ LSTP2010$	Area MUX LSTP 2010	μm^2	AreaLSTP

Table 4.2: Output data

4.4.1 Main code

```
%MUX area calculation

clear all;
close all;
clc
prompt = 'Specify␣the␣nMOS␣channel␣width␣Wn␣(um):␣␣';
Wn = input(prompt)%um, Gate width of n-MOS.
interc_override=0.15; %we estimate 15% of override
beta =1.29; %is the ratio between Wp/Wn.
gateW_p = beta*Wn; %um, Gate width of p-MOS

lambda = 1; % referred to the design rules

%area MOS HP 2010
Lgate =0.018; %um, The length of the gate.
L_s_d = lambda*Lgate;
```

```
L=Lgate+2*L_s_d; %um total length.
areaNand2HP=2*( Wn+gateW_p )*L *(1+ interc_override ); % um^2

%area MOS LOP 2010
Lgate =0.022; %um, The length of the gate.
L_s_d = lambda*Lgate;
L=Lgate+2*L_s_d; %um total length.
areaNand2LOP=2*( Wn+gateW_p )*L *(1+ interc_override ); % um^2

%area MOS LSTP 2010
Lgate =0.028; %um, The length of the gate.
L_s_d = lambda*Lgate;
L=Lgate+2*L_s_d; %um total length.
areaNand2LSTP=2*( Wn+gateW_p )*L *(1+ interc_override ); % um^2

prompt = 'Specify␣the␣number␣of␣inputs:␣␣';
X = input(prompt);

prompt = 'Specify␣the␣size␣in␣bit␣of␣the␣word:␣␣';
%Nbit has a mean only if is 1,2,4,8,16,32,64
Word = input(prompt);

%Number of Nand gates
N_gates1 = 3*Word*(X-1);
B = Word*(X-1)/4;
N_gates2 = floor(B); %number of NAND2 FO4
decimal = B - floor(B);

cont_FO2= 0; %number of NAND2 FO2
cont_FO1= 0; %number of NAND2 FO1

if (decimal == 0.75)
     cont_FO2= 1;
     cont_FO1= 1;
elseif (decimal == 0.50)
     cont_FO2= 1;
elseif (decimal == 0.25)
     cont_FO1= 1;
end

```

```
%%
N_gate = N_gates1 + N_gates2 + cont_FO2 +cont_FO1;

%Area evaluation
AreaHP = areaNand2HP*N_gate
AreaLOP = areaNand2LOP*N_gate
AreaLSTP = areaNand2LSTP*N_gate
```