Politecnico di Torino

III Facoltà di Ingegneria

Area, power and timing analysis of half adder and full adder Integrated system technology

GROUP: 08

Jacopo Abramo

July 24, 2017

Contents

1 Introduction

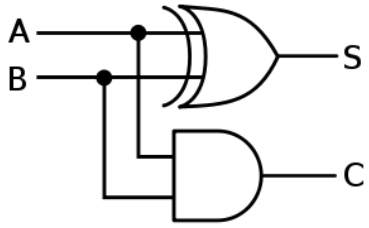
The aim of this project is to create a parametrical model to describe the behaviour of the basic components used to realize arithmetical circuits, the half adder and the full adder. In the Theoretical analysis is discussed the general description of both components above mentioned and then the calculation of power consumption, occupied area and timing. The calculations have been executed transforming the logic gates inside the structure in NAND gates; this is a choice made in order to simplify calculations of power and timing. In the Octave implementation is shown the Matlab code used to realize the models. Moreover, two tables, which list the legend of the parameters, have been created in order to help the reader during the reading of this document. The last section concerns the results for each analysis indicated above. The used technological parameters refer to Bulk Technology HP 2009.

2 Theoretical analysis

2.1 Half Adder

The half adder circuit is shown in figure 1: As shown in the figure, the half adder generates two outputs starting from two

Figure 1: Half adder



input bits. The S bit represents the sum bit, and has the logic function:

$$S = A \oplus B \tag{1}$$

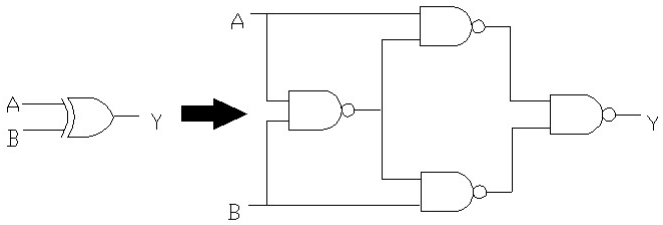
The C bit represents the carry bit, and the logic function is:

$$C = AB \tag{2}$$

2.1.1 Area

As said in chapter 1, all logic gates will be considered as NAND gates. The XOR gate in NAND technology becomes as shown in figure 2 [1]. The area occupied by the XOR gate in NAND technology is four times the area occupied by a single

Figure 2: XOR gate realized with NAND gates



NAND gate; in the calculations, the area override due to interconnections has been taken into account. The CMOS structure of the NAND gate is shown in figure 3. By theory, the area of a single NAND gate is computed as:

$$area_{NAND} = 2 \cdot (W_n + W_p) \cdot (1 + override) \cdot L \tag{3}$$

- W_n is the channel width for nMOS;
- W_p is the channel width for pMOS;
- $L = L_{eff} + 2L_{s,d}$ is the total channel length; L_{eff} is the effective gate length, while $L_{s,d}$ is the source length, equal to the drain length, hence the factor 2;
- *override* is the area override due to interconnections.

Figure 3: NAND gate in CMOS technology

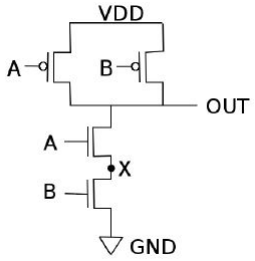
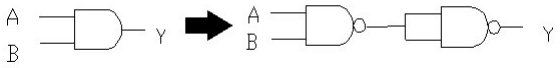


Figure 4: AND gate realized with NAND gates



From design rules, it is also known that $W_p = \beta W_n$, with $\beta = 1.29$ as a choice. The area of the XOR gate is then:

$$area_{XOR} = 4 \cdot area_{NAND} \tag{4}$$

The AND gate can be seen in NAND technology as in figure 4 [1]. The area of the AND gate is then:

$$area_{AND} = 2 \cdot area_{NAND} \tag{5}$$

Finally, the total area of the half adder is:

$$area_{HA} = area_{AND} + area_{XOR} = 6 \cdot area_{NAND} \tag{6}$$

2.1.2 Timing

Timing analysis of the half adder takes into consideration the critical path of the combinational circuit. In this case, two paths have to be analyzed: the path generating the sum bit, and the path generating the carry bit. As said, the starting assumption (all gates represented in NAND technology) gives the possibility to use the roadmap modelization for the delay. The delay of a single 2-input NAND gate is computed as:

$$\tau_{NAND} = \frac{C_{NAND} \cdot V_{DD}}{I_{NAND}} \tag{7}$$

V_{DD} is the supply voltage, I_{NAND} is the current of the NAND gate and C_{NAND} is the NAND gate capacitance, equal to

$$C_{NAND} = C_{FO4} + C_{jNAND2} \tag{8}$$

C_{jNAND2} is the junction capacitance, while C_{FO4} is the fan-out capacitance, which is the capacitance that it is possible to see at the input multiplied for the fan-out:

$$C_{FO4} = 4 \cdot C_{in_{tot}} \tag{9}$$

$C_{in_{tot}}$ is expressed as:

$$C_{in_{tot}} = C_{in_{nMOS}} + C_{in_{pMOS}} \tag{10}$$

Where:

$$C_{in_{nMOS}} = 2C_{ox}L_{eff}W_n + 2C_{overlap_n}W_n \tag{11}$$

$$C_{in_{pMOS}} = 2C_{ox}L_{eff}W_p + 2C_{overlap_p}W_p \tag{12}$$

$C_{overlap}$ are the capacitances due to the fact that the gate oxide overlaps the source and drain region. The junction capacitance C_{jNAND2} is computed instead as follows:

$$C_{jNAND2} = C_{jn_n} + 2C_{jn_p} \tag{13}$$

Where

$$C_{jn_n} = C_{bottom_n} Wn + C_{sidewall_n} \cdot perimeter_n \tag{14}$$

$$C_{jn_p} = C_{bottom_p} Wp + C_{sidewall_p} \cdot perimeter_p \tag{15}$$

For the sum bit, the critical path for corresponds to the delay of the XOR gate in NAND configuration, shown in figure 5. For the carry bit, the critical path corresponds to the delay of the AND gate in NAND configuration, shown in figure 6. It is

Figure 5: XOR gate critical path

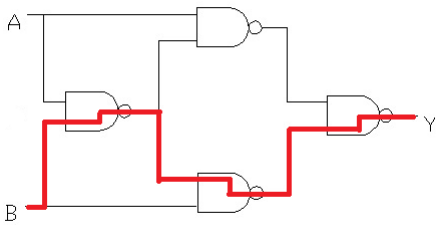


Figure 6: AND gate critical path

clear that the critical path of the half adder corresponds to the longest chain of NAND gates, which is the one corresponding to the XOR gate. Therefore, the critical path of the half adder is equal to the delay of the XOR gate, which is - in NAND technology - equal to:

$$\tau_{HA} = 3\tau_{NAND} \tag{16}$$

This allows a maximum operating frequency equal to:

$$f_{max_{HA}} = \frac{1}{\tau_{HA}} = \frac{1}{3\tau_{NAND}} \tag{17}$$

2.1.3 Power consumption: dynamic power

The total power consumption of a logic gate is computed as

$$P_{tot} = P_{dynamic} + P_{static} \tag{18}$$

Focusing on the dynamic power, this is computed as:

$$P_{dynamic} = \alpha \cdot V_{DD}^2 \cdot C \cdot f \tag{19}$$

- α is the switching activity of the gate;
- V_{DD} is the power supply;
- C is the capacitance of the logic gate (in the case of the NAND gate is equal to C_{NAND});
- f is the operating frequency.

For the half adder, C is equal to the total capacitance of the XOR gate, which is equal to $4C_{NAND}$ (since the XOR gate in NAND technology is composed of 4 NAND gates, singularly contribuiting to the total switching capacitance), while the operating frequency f is equal to $f_{max_{HA}}$.

2.1.4 Power consumption: static power

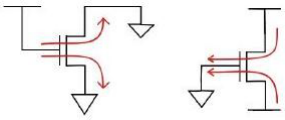
The static power of a logic gate is computed as:

$$P_{static} = V_{DD} \cdot I_{static} \tag{20}$$

The contributions to I_{static} are given by:

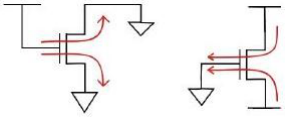
- subthreshold current: is the I_{off} current that flows in source node of the nMOS when the gate is connected to ground and the drain to the power supply;
- gate leakage: is the gate current that flows from the gate to the channel when the gate is connected to V_{DD} while the drain and the source are connected to the ground (figure 7, case a), or it is the current that flows in opposite direction when the source and drain are connected to V_{DD} and the gate is connected to the ground (figure 7, case b).

Figure 7: Gate leakage, case a and b



There are two intermediate cases to find the gate leakage: when the drain and source are connected to different voltages and the gate is connected to the ground or to the power supply (Figure 6 case c and case d). In these two cases there is a contribution of current but is considerably smaller than previous cases; therefore these two contributions are neglected.

Figure 8: Gate leakage, case c and d



A	B	OUT	Static current
0	0	1	$I_{off_n} + 2I_{gate_p}$
0	1	1	$I_{off_n} + I_{gate_p}$
1	0	1	$I_{off_n} + I_{gate_n} + Igate_p$
1	1	0	$2I_{off_p} + 2I_{gate_n}$

In the table above all 4 possible input combinations of a 2-input NAND gate are taken into account, providing a combination of gate and leakage current for each case. Assuming that each combination has a 25% probability, the average leakage current of the NAND gate is the sum of all previous currents multiplied by a factor 0.25:

$$I_{static_{NAND}} = \frac{1}{4}(3I_{off_n} + 4I_{gate_p} + I_{gate_n} + 2I_{off_p} + 2I_{gate_n}) \tag{21}$$

Focusing on the NAND equivalent of the AND gate, it is worth noticing that from the schematic (figure 4), the second NAND gate considers only two possible sets of inputs: 00 and 11, because both inputs are connected to a single line. This gate is, in fact, equivalent to an inverter gate. Therefore, the average current of this inverter-like NAND gate is equal to:

$$I_{static_{NANDasINV}} = \frac{1}{2}(I_{off_n} + 2I_{gate_p} + 2I_{off_p} + 2I_{gate_n}) \tag{22}$$

The factor $\frac{1}{2}$ takes into account the 50% probability of each input combination. Finally, the average static current of the XOR gate is:

$$I_{static_{XOR}} = 4 \cdot I_{static_{NAND}} \tag{23}$$

Instead, the average static current of the AND gate is:

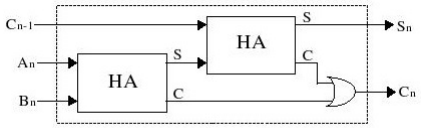
$$I_{static_{AND}} = I_{static_{NAND}} + I_{static_{NANDasINV}} \tag{24}$$

Overall, the static current inside an half adder structure is:

$$I_{static_{HA}} = I_{static_{AND}} + I_{static_{XOR}} \tag{25}$$

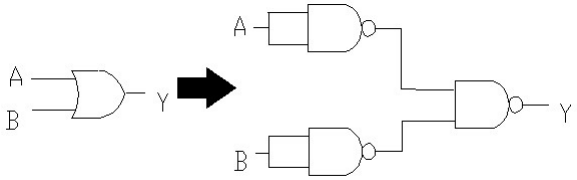
2.2 Full adder

Figure 9: Full adder circuit



The full adder circuit is shown in figure 9. It is formed by 2 half adders and a final OR gate which generates the carry output. Considering an implementation using NAND gates technology, it is possible to make use of the previous analysis and extent it to this case. First of all, the NAND gate version of the OR gate has to be used. This is shown in figure 10. It is worth noticing that both A and B input in the figure above make use of the inverter-like NAND gate, previously used to realize the AND gate generating the carry bit in the half adder (section 2.1.1).

Figure 10: OR gate realized with NAND gates



2.2.1 Area

The overall area of the full adder can be easily computed taking into account the previous computations made for the half adder (see 2.1.1): since the full adder is formed by a cascade of 2 half adders and an OR gate, the overall area is:

$$area_{FA} = 2 \cdot area_{HA} + area_{OR} \tag{26}$$

With:

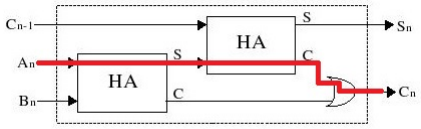
$$area_{OR} = 3 \cdot area_{NAND} \tag{27}$$

2.2.2 Timing

The critical path of the full adder is shown in figure 11. As shown, the path goes from input to output passing from the first half adder (the S output corresponds to the output of the XOR gate), then from the second half adder (the C output corresponds to the output of the AND gate), and finally from the OR gate. The overall delay is:

$$\tau_{FA} = \tau_{HA1} + \tau_{HA2} + \tau_{OR} \tag{28}$$

Figure 11: Full adder critical path



τ_{HA1} is the delay corresponding to the XOR gate of the first half adder, and is equal to $3 \cdot \tau_{NAND}$; τ_{HA2} is the delay corresponding to the AND gate of the second half adder, and is equal to $2 \cdot \tau_{NAND}$. τ_{OR} is the delay of the OR gate. From figure 10, it is clear that the delay of the OR gate in NAND technology is:

$$\tau_{OR} = 2 \cdot \tau_{NAND} \tag{29}$$

Which means that the overall delay of the full adder is:

$$\tau_{FA} = 7 \cdot \tau_{NAND} \tag{30}$$

And the maximum working frequency is:

$$f_{max_{FA}} = \frac{1}{\tau_{FA}} = \frac{1}{7\tau_{NAND}} \tag{31}$$

2.2.3 Power consumption: dynamic power

As seen in section 2.1.3, the dynamic power of a digital circuit is computed as:

$$P_{dynamic} = \alpha \cdot V_{DD}^2 \cdot C \cdot f \tag{32}$$

In case of the full adder, the maximum working frequency corresponds to $f_{max_{FA}}$.

2.2.4 Power consumption: static power

As seen in section 2.1.4, the static power of a digital circuit in CMOS technology is computed as:

$$P_{static} = V_{DD} I_{static} \tag{33}$$

For the full adder, the I_{static} can be considered as seen in section 2.1.4, with a surplus term given by the presence of the OR gate.

$$I_{static_{FA}} = 2 \cdot I_{static_{HA}} + I_{static_{OR}} \tag{34}$$

The term $I_{static_{OR}}$ can be computed taking into consideration the NAND version of the OR gate. It is possible to see in figure 10 that the NAND gates connected to the 2 inputs only consider two possible sets of inputs: 00 and 11, as it was before with the AND gate in NAND technology. These two gates give a static current contribution equal to $I_{static_{NANDasINV}}$; therefore:

$$I_{static_{OR}} = 2 \cdot I_{static_{NANDasINV}} + I_{static_{NAND}} \tag{35}$$

Finally, the overall power consumption of the full adder is the sum of $P_{dynamic}$ and P_{static}, as seen before.

3 Octave implemetation

Below is reported the MATLAB code and the tables of input (table 1) parameters and output parameters (table 2) for simpler reading.

```
clc;
close all;
clear all;

%% PARAMETER
%the default values of parameter are referred to the roadmap 2009 , years
%2011
lambda =1;                  %referred to the rules design
L_eff =27*10^(-3);          %um %the effective length of the gate
L_s_d = lambda * L_eff ;    %source length equal to drain length
L = L_eff + 2* L_s_d ;      %total length
interc_override = 0.15;         %15% of interconnetion override
alpha = 0.15; %activity factor
vdd = 1;                    %V
Wgate = [10*L 10*L*1.2 10*L*1.4 10*L*1.6 10*L*1.8 10*L*2];
                            %um width of NMOS W_n =10* L;
W_n = Wgate ;               %um
beta =1.29;
W_p = beta * Wgate ;        %um width of PMOS W_p =1.29* W_n ;

Cj0n =2.7*10^(-3);        %pf/um ^2 N- MOS junction capacitance per unit ...
                          % area under zero - bias conditions
Cj0p =3.3*10^(-3);        %pF/um ^2 P- MOS junction capacitance per unit ...
                          % area under zero - bias conditions

Cjswn = 9.2*10^(-10);     %F/m N-MOS sidewall junction capacitance
Cjswp =8*10^(-10);        %F/m P-MOS sidewall junction capacitance
Cgd0n =1.35*10^(-10);     %F/m N-MOS overlap capacitance between gate ...
                          %and drain per unit transistor width

Cgd0p =1*10^(-10);        %F/m P-MOS overlap capacitance between gate and ...
                          % drain per unit transistor width
Mjn = 0.38;
Mjp = 0.45;
Mswn = 0.22;
Mswp = 0.265;
Pbn = 0.85;
Pbp = 0.87;
Pbswn = 0.67;
Pbswp = 0.76;
k = 10;
Cox = 4.6041e-06; %pf/um ^2
n_fan_out = 4; %Nand with two input is loaded with four identical inverter
               %Source and drain diffusion length
lungh_diff =2.5* L_eff ; %um
```

```
46
47 % nMOS source and drain diffusion capacitance
48 Cbottom_n = lungh_diff * Cj0n *(1+ vdd /(2* Pbn ))^( - Mjn ); %pF/um
49
50 % nMOS source and drain sidewall capacitance
51 Csidewall_n =1e6* Cjswn *(1+ vdd /(2* Pbswn ))^( - Mswn );    %pF/um
52
53 % pMOS source and drain diffusion capacitance
54 Cbottom_p = lungh_diff * Cj0p *(1+ vdd /(2* Pbp ))^( - Mjp ); %pF/um
55
56 % pMOS source and drain sidewall capacitance
57 Csidewall_p =1e6* Cjswp *(1+ vdd /(2* Pbswp ))^( - Mswp );    %pF/um
58
59 WD=k* L_eff ;
60 perim_N =2* lungh_diff +WD;
61 perim_P =2* lungh_diff +2* WD;
62 Cjn= Cbottom_n *WD+ Csidewall_n * perim_N ; %pF
63 Cjp= Cbottom_p *2* WD+ Csidewall_p * perim_P ; %pF
64 Coverlap_n =1e6* Cgd0n ; %pF/um
65 Coverlap_p =1e6* Cgd0p ; %pF/um
66 Cin = Cox* L_eff*1e-3*2* W_n + Cox * L_eff*1e-3* W_p + ...
67 2* Coverlap_n +2* Coverlap_p; %pF
68
69 Cout = Cjn + 2* Cjp;                          %pF
70 Cl_nand = Cin* n_fan_out ;                    %pF   %NAND load capacitance
71 cap_nand = ( Cout +Cl_nand )*10^(3);    %fF   %capacitance of NAND2
72 ioff_n =  26*10^( -1)* W_n;     %uA
73 ioff_p =  7.9*10^( -1)* W_p;    %uA
74 igate_n = 22*10^(-2)* W_n ;     %uA
75 igate_p = 14*10^(-2)* W_p ;     %uA
76 i_NAND =  4.24* W_n;            %mA
77
78 %% HALF ADDER
79 %% HALF ADDER AREA ANALYSIS
80
81 %%%%%%%%%%%% SINGLE GATES AREA %%%%%%%%%%%%
82 %the 2 input NAND gate in CMOS technology comprehends:
83 %2 nMOS and 2 pMOS
84 one_nand_area = 2*L.*(W_n + W_p)*(1 + interc_override); %um^2
85
86 %the XOR gate - whose output corresponds to the SUM bit
87 %is equivalent to 4 NAND gates
88 one_xor_area = 4*one_nand_area; %um^2
89
90 %the AND gate - whose output corresponds to the CARRY bit - can be seen as
91 %2 NAND gates
92 one_and_area = 2*one_nand_area; %um^2
93
94 %%%%%%%%%%%% HALF ADDER AREA %%%%%%%%%%%%
```

```
%the total area of the half adder with this configuration corresponds to:
area_half_adder = one_xor_area + one_and_area; %um^2

%% HALF ADDER TIMING ANALYSIS

%%%%%%%%%%%%% SUM BIT %%%%%%%%%%%%%

%the critical path of the sum corresponds to the delay of the
%XOR gate (represented as 4 NAND gates)
%with this configuration, the critical path of the XOR gate is the delay
%of 3 NAND gates

%the delay of a single NAND gate is
t_nand = (cap_nand.*vdd)./i_NAND; %ps

%that means that the delay of the XOR gate is
t_xor = 3*t_nand;                 %ps
t_half_adder_sum_bit = t_xor;     %ps

%%%%%%%%%%%%% CARRY BIT %%%%%%%%%%%%%

%the delay of the AND gate is then
t_and = 2*t_nand;                 %ps
t_half_adder_carry_bit = t_and; %ps

%we assume that the delay of the critical path (which is the inverse of the
%maximum allowed frequency) is the maximum delay between the sum bit path
%and the carry bit path

%%%%%%%%%%%%% MAXIMUM FREQUENCY %%%%%%%%%%%%%
t_max_half_adder = max(t_half_adder_carry_bit, t_half_adder_sum_bit); %ps
freq_allowed_half_adder = 1./t_max_half_adder;

%% HALF ADDER POWER ANALYSIS

%%%%%%%%%%%%% DYNAMIC POWER %%%%%%%%%%%%%

%the power consumption is estimated as the sum of the static power and the
%dynamic power of the circuit

%the dynamic power of the half adder can be estimated as the sum of the
%dynamic power of the XOR gate and the dynamic power of the AND gate

%the dynamic power is defined as the product of the switching activity
%of the gate times the square of the supply voltage, times the
%capacitance of the gate, times the frequency

%the dynamic power of the xor gate is
cap_xor = 4*cap_nand;                                              %pF
```

```
144 P_dyn_xor_half_adder = alpha*vdd^2.*cap_xor.*freq_allowed_half_adder; %uW
145
146 %the dynamic power of the and gate is
147 cap_and = 2*cap_nand;                                                  %pF
148 P_dyn_and_half_adder = alpha*vdd^2.*cap_and.*freq_allowed_half_adder; %uW
149
150 %the overall dynamic power of the half adder is
151 P_dyn_half_adder = P_dyn_xor_half_adder + P_dyn_and_half_adder; %uW
152
153 %%%%%%%%%%%% STATIC POWER %%%%%%%%%%%%
154
155 %the static power is given by two contributions: the subthreshold
156 %leakage (Ioff) and the gate leakage (Igate).
157 %assuming that the probability associated to the gate NAND is 25% for each
158 %case, the static current is given by:
159 a = 0.25;
160 i_stat_nand = a*(ioff_n +2*igate_p )+a *(ioff_n + igate_p)+...
161     a*(ioff_n + igate_n+igate_p )+a *(2*ioff_p+2*igate_n ); %uA
162 %the overall static current of the XOR gate is computed considering 4 NAND
163 %gates
164 i_stat_xor = 4*i_stat_nand; %uA
165
166 %instead, the overall static current of the AND gate is given by the
167 %static current of the first NAND gate and the static current of the second
168 %NAND gate; this one though as only two possible input configuration: 00
169 %and 11 (it behaves like an inverter gate);
170 %therefore, the overall current will be:
171 b = 0.5;                    %assuming 50% probability for each input
172                             %configuration
173 i_stat_nand_as_inv = b*(ioff_n + 2*igate_p + 2*ioff_p + 2*igate_n); %uA
174 i_stat_and = i_stat_nand + i_stat_nand_as_inv;
175
176 %the overall static power is computed as the supply voltage times the sum
177 %of the overall static currents
178 i_stat_half_adder = i_stat_xor + i_stat_and;
179 P_stat_half_adder = vdd*i_stat_half_adder; %uW
180
181 %the overall power dissipated by the half adder is, as a consequence:
182 P_tot_half_adder = P_dyn_half_adder + P_stat_half_adder; %uW
183
184 %% FULL ADDER
185 %% FULL ADDER AREA ANALYSIS
186
187 %the full adder circuit includes two full adders and an OR gate for the
188 %computation of the carry bit
189 %the OR gate is equivalent to three NAND gates
190 %the area of an OR gate with NAND configuration is
191 one_or_area = 3*one_nand_area; %um^2
192
```

```
%while the overall area of the full adder is equal to
area_full_adder = 2*area_half_adder + one_or_area; %um^2

%% FULL ADDER TIMING ANALYSIS

%%%%%%%%%%%% SUM BIT %%%%%%%%%%%%

%the overall critical path for the computation of the sum bit in a full
%adder is two times the critical path of the half adder for the computation
%of the sum bit
t_full_adder_sum_bit = 2*t_half_adder_sum_bit;     %ps

%%%%%%%%%%%% CARRY BIT %%%%%%%%%%%%

%instead, the overall critical path for the computation of the carry bit in
%a full adder is the sum of the delay for the computation of the sum bit
%(first stage), the delay for the computation of the carry bit (second
%stage) and the delay of the OR gate

%the delay of the OR gate with NAND configuration is the delay of 2 NAND
%gates
t_or = 2*t_nand; %ps
t_full_adder_carry_bit = t_half_adder_sum_bit + t_half_adder_carry_bit +...
    t_or;         %ps

%%%%%%%%%%%% MAXIMUM FREQUENCY %%%%%%%%%%%%

%the maximum allowed frequency is the inverse of the maximum of the two
%computed delays
t_max_full_adder = max(t_full_adder_carry_bit, t_full_adder_sum_bit); %ps
freq_allowed_full_adder = 1./t_max_full_adder;

%% FULL ADDER POWER ANALYSIS

%%%%%%%%%%%% DYNAMIC POWER %%%%%%%%%%%%

%the dynamic power of the full adder is the sum of the dynamic power of the
%two half adders plus the dynamic power of the OR gate

%since the OR gate is formed by 3 NAND gates, the overall capacitance is
cap_or = 3*cap_nand; %pF

%the dynamic power of the OR gate is
P_dyn_or = alpha.*vdd^2.*cap_or.*freq_allowed_full_adder; %uW

%overall, the dynamic power of the full adder is
P_dyn_xor_full_adder = alpha.*vdd^2.*cap_xor.*freq_allowed_full_adder; %uW
P_dyn_and_full_adder = alpha.*vdd^2.*cap_and.*freq_allowed_half_adder; %uW

```

```
P_dyn_full_adder = 2*(P_dyn_xor_full_adder + P_dyn_and_full_adder) + ...
                       P_dyn_or; %uW

%%%%%%%%%%%% STATIC POWER %%%%%%%%%%%%

%the static current of the full adder is 2 times the static current of a
%single full adder plus the static current of the OR gate with NAND
%configuration

%the OR gate with NAND configuration shows that the first 2 NAND gates
%behave as an inverter (allowing only a 00 or 11 input configuration);
%therefore, the overall static current of the OR gate is
i_stat_or = 2*i_stat_nand_as_inv + i_stat_nand;

%the overall static current is
i_stat_full_adder = 2*i_stat_half_adder + i_stat_or;

%the overall static power is:
P_stat_full_adder = vdd*i_stat_full_adder;

%finally, the overall power dissipated by the full adder is the sum of
%static and dynamic power
P_tot_full_adder = P_dyn_full_adder + P_stat_full_adder;

%% PLOTTING RESULTS

%%%%%%%%%%%% AREA %%%%%%%%%%%%

figure;
plot(Wgate, area_half_adder, 'b+:');
hold on;
plot(Wgate, area_full_adder, 'r+:');
grid on;

legend ({'Half adder','Full adder'})
title ('Area Half Adder vs Area Full Adder')
xlabel ('Gate width [um]')
ylabel ('Area [um^2]')

%%%%%%%%%%%% TIMING %%%%%%%%%%%%

figure;
plot(Wgate, t_max_half_adder, 'b+:');
hold on;
plot(Wgate, t_max_full_adder, 'r+:');
grid on;

legend ({'Half adder','Full adder'})
title ('Delay Half Adder vs Delay Full Adder')
```

```
xlabel (' Gate width [um]')
ylabel (' Delay [ps] ')


figure;
plot(Wgate, freq_allowed_half_adder, 'b+:');
hold on;
plot(Wgate, freq_allowed_full_adder, 'r+:');
grid on;

legend ({'Half adder','Full adder'})
title (' Frequency Half Adder vs Frequency Full Adder ')
xlabel (' Gate width [um]')
ylabel (' Frequency [GHz] ')

%%%%%%%%%%%% POWER %%%%%%%%%%%%
%%%%%%%%%%%% POWER - Fixed alpha %%%%%%%%%%%%
%%%%%%%%%%%% Using alpha = 0.15 %%%%%%%%%%%%
figure;
plot(Wgate, P_tot_half_adder, 'b+:');
hold on;
plot(Wgate, P_tot_full_adder, 'r+:');
grid on;

legend ({'Half adder','Full adder'})
title (' Power Half Adder vs Power Full Adder ')
xlabel (' Gate width [um], alpha = 15%')
ylabel (' Power [uW] ')

%%%%%%%%%%%% POWER - Fixed width %%%%%%%%%%%%
%%%%%%%%%%%% Using Wgate = 10*L %%%%%%%%%%%%
alpha_array = [0.15 0.3 0.45 0.6 0.75 0.9];
P_dyn_half_adder_new = P_dyn_half_adder(1)/alpha.*alpha_array;
P_dyn_full_adder_new = P_dyn_full_adder(1)/alpha.*alpha_array;

P_tot_half_adder_new = P_dyn_half_adder_new + P_stat_half_adder;
P_tot_full_adder_new = P_dyn_full_adder_new + P_stat_full_adder;

figure;
plot(alpha_array*100, P_tot_half_adder_new, 'b+:');
hold on;
plot(alpha_array*100, P_tot_full_adder_new, 'r+:');
grid on;

legend ({'Half adder','Full adder'})
title (' Power Half Adder vs Power Full Adder ')
xlabel (' Switching activity [%], Gate width = 10*L [um]')
ylabel (' Power [uW] ')
```

Quantity name	Description	u.m. (S.I.)	Variable name
L_{eff}	Effective gate length	μm	L_eff
$L_{s,d}$	Source and drain length	μm	L_s_d
$inter_{ovr}$	Area override due to interconnections	%	interc_override
V_{DD}	Power supply	V	vdd
α	Switching activity	%	alpha
W_{gate}	nMOS width	μm	Wgate
β	beta factor between nMOS and pMOS	/	beta
C_{j0n}	nMOS junction capacitance per unit area	$\frac{pF}{\mu m^2}$	Cj0n
C_{j0p}	pMOS junction capacitance per unit area	$\frac{pF}{\mu m^2}$	Cj0p
C_{jswn}	nMOS sidewall junction capacitance per unit area	$\frac{pF}{\mu m^2}$	Cjswn
C_{jswp}	pMOS sidewall junction capacitance per unit area	$\frac{pF}{\mu m^2}$	Cjswp
C_{gd0n}	nMOS overlap capacitance between gate and drain per unit area	$\frac{pF}{\mu m^2}$	Cgd0n
C_{gd0p}	pMOS overlap capacitance between gate and drain per unit area	$\frac{pF}{\mu m^2}$	Cgd0p
C_{ox}	Oxide capacitance per unit area	$\frac{pF}{\mu m^2}$	Cox
L_{diff}	Source and drain diffusion length	μm	lungh_diff
C_{NAND}	NAND2 capacitance	fF	cap_nand
I_{off_n}	subthreshold current for nMOS	μA	ioff_n
I_{off_p}	subthreshold current for pMOS	μA	ioff_p
I_{gate_n}	gate current for nMOS	μA	igate_n
I_{gate_p}	gate current for pMOS	μA	igate_p
I_{NAND}	output current for NAND gate	μA	i_NAND

Table 1: Input data

Quantity name	Description	u.m. (S.I.)	Variable name
$area_{HA}$	Area half adder	μm^2	area_half_adder
$area_{FA}$	Area full adder	μm^2	area_full_adder
$t_{cp,HA}$	Critical path half adder	ps	t_max_half_adder
$t_{cp,FA}$	Critical path full adder	ps	t_max_full_adder
$f_{max,HA}$	Maximum allowed frequency for half adder	GHz	freq_allowed_half_adder
$f_{max,FA}$	Maximum allowed frequency for full adder	GHz	freq_allowed_full_adder
$P_{tot_{HA,W}}$	Half adder total power consumption with fixed α	μW	P_tot_half_adder
$P_{tot_{FA,W}}$	Full adder total power consumption with fixed α	μW	P_tot_full_adder
$P_{tot_{HA,\alpha}}$	Half adder total power consumption with fixed gate width	μW	P_tot_half_adder_new
$P_{tot_{FA,\alpha}}$	Full adder total power consumption with fixed gate width	μW	P_tot_full_adder_new

Table 2: Output data

4 Results

Figure 12 shows the variation of the area of both half adder and full adder as the gate width increases by 20% for each node. It is clear that the area is linearly dependent with the width of the transistor.

Figure 12: MATLAB area plot

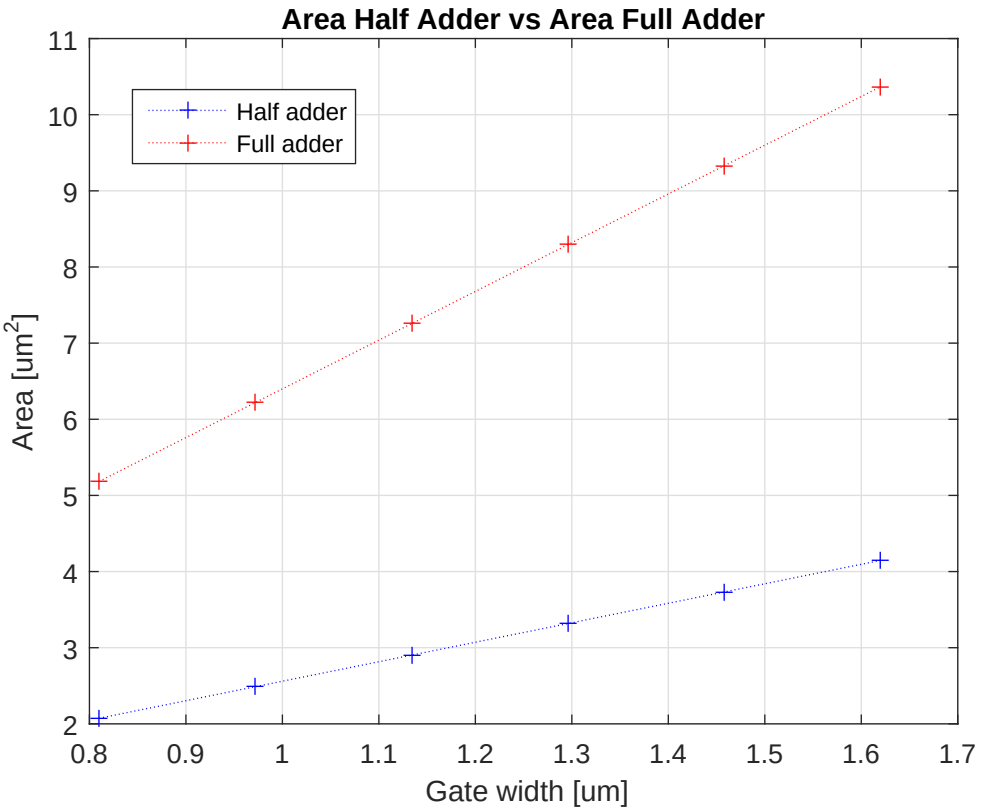


Figure 13 shows the dependency between the delay of both half adder and full adder to the gate width. The plot shows an asintotic behavior, explained by the fact that the NAND capacitance is not affected by the variation of the gate width (almost remaining constant), while instead having the driving current of the NAND gate linearly increasing. Since the delay is inversely proportional to the driving current, this explains the obtained results.

Figure 13: MATLAB delay plot

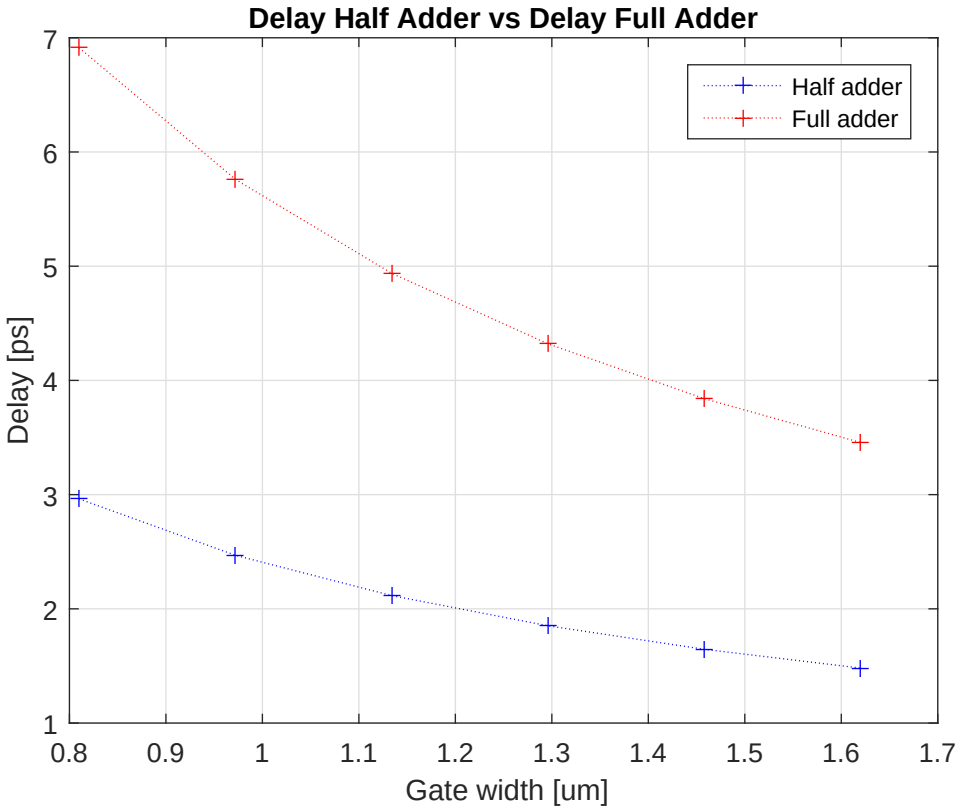


Figure 14 shows the dependency between frequency of half adder and full adder to the gate width. Since it is the inverse of the critical path, the behavior is simply linear.

Figure 14: MATLAB freq plot

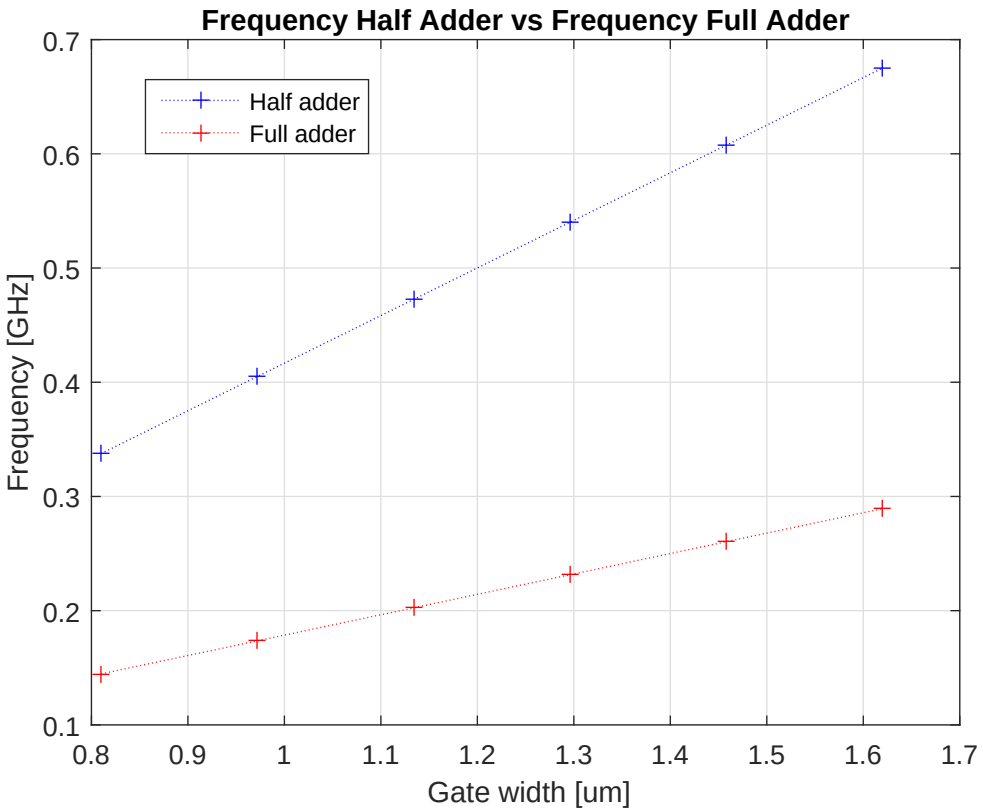


Figure 15 shows the dependency of the power consumption of both half adder and full adder in respect to the gate width, considering a fixed switching activity. The dynamic power is linearly dependent to the frequency (which has a linear tendence) and to the NAND capacity. Therefore, the result will be a linear plot as well.

Figure 15: MATLAB power plot

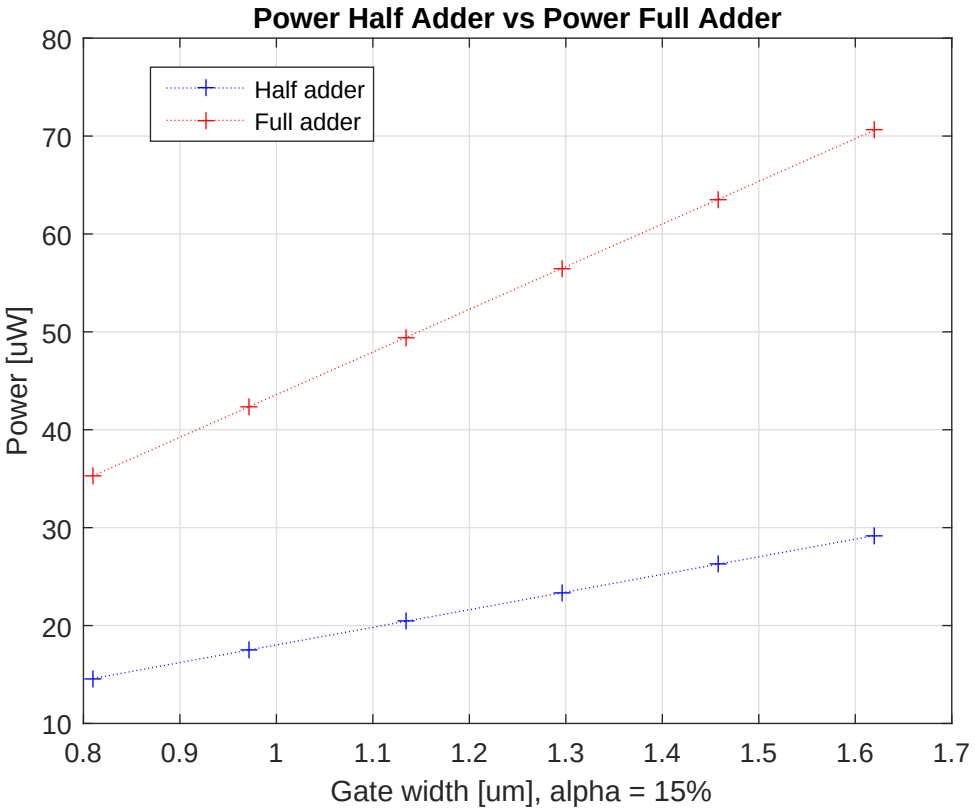
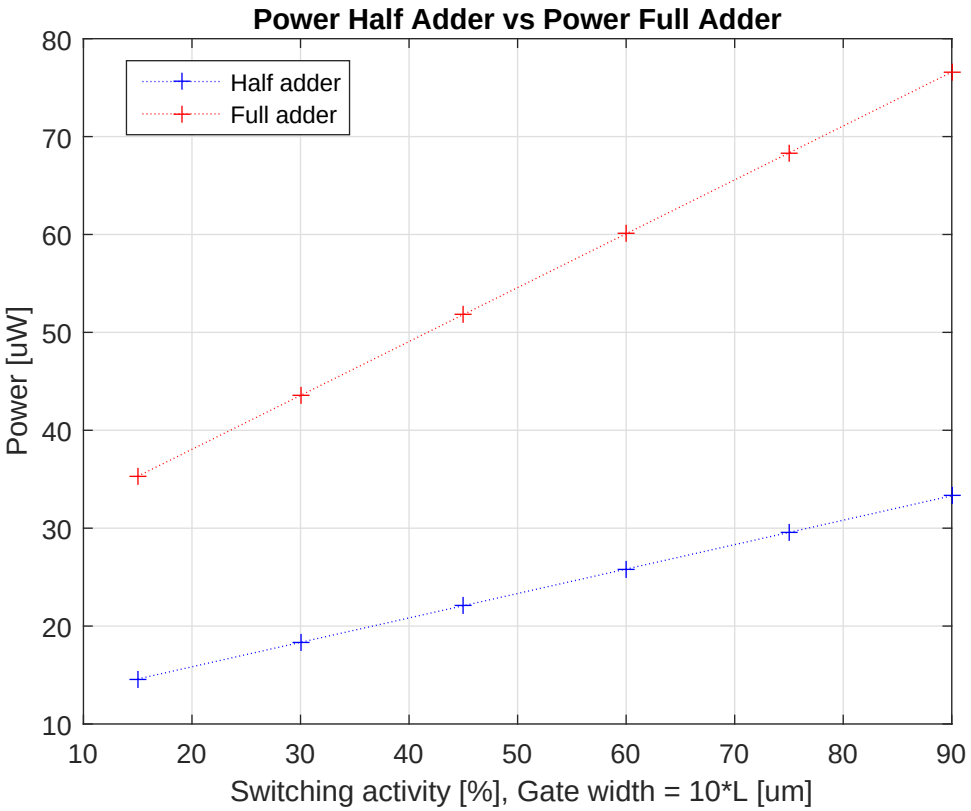


Figure 16 shows the dependency of the variation of the switching activity of the system to the overall power consumption, considering a fixed gate width. The trend of the power consumption is still linear, but higher α show an higher power consumption in respecto to higher values of the gate width.

Figure 16: MATLAB power plot



References

[1] Weste, Neil H. E.; Money Harris, David, *CMOS VLSI Design - A circuit and systems perspective*, fourth edition