UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUÍS HENRIQUE REINICKE

Construction of a Low-Voltage Standard Cell Library for Ultra-low Power Applications

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engeneering

Advisor: Prof. Dr. Renato Perez Ribas

Porto Alegre
December 2014

“All animals are equal, but some animals are more equal than others.”

— GEORGE ORWELL, ANIMAL FARM

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Prof. Renato Ribas, with whom I have been working for two years, for all the knowledge he shared with me. I also thank my co-workers at the laboratory (Eduardo, Cícero, Lauro) for their help and suggestions for this work.

Next, I thank my close family (father, mother and sister) and my brothers Lad, Meirense, Glenn, Mizu, Bigodi, Jão, Barba and Incerto for the wonderful times that we spend together. I thank the good friends that I made at the university, Alan, Alessandra and Jeckson, for their support and friendship in all times.

Finally, I thank the CI Brasil program, whose structure and personnel allowed me to finish this work. This research was partially funded by the UFRGS-CEITEC S.A partnership through CNPq and FINEP, project "Programa Nacional de Formação de Projetistas de Circuitos Integrados" under the number 455067/2013-9, advised by Prof. Dr. Eric Ericson Fabris.

ABSTRACT

In digital IC design area, low-power computation has been a necessity since applications that required both performance and lower energy operation (such as cellphones and laptops) became popular. Among several low-power techniques, reducing the supply voltage is arguably the most efficient way to reduce power consumption, as they share a quadratic relationship. However, working at lower voltages bring new difficulties and challenges that need to be overcame in order to have significant power consumption gains. This work presents guidelines to construct a standard cell library aimed to be used at a low voltage range, specifically the near-threshold voltage regime, which was chosen over the sub-threshold voltage regime for a series of reasons explained in the text.

Keywords: CMOS. Low-power. Low-voltage. Near-threshold. Standard Cell Library.

RESUMO

Na área de projeto de circuitos integrados digitais, computação de baixo consumo tem sido uma necessidade desde que aplicações que requerem tanto alta performance quanto baixo consumo de potência (como celulares e laptops) tornaram-se populares. Entre inúmeras técnicas de baixo consumo, diminuir a tensão de alimentação é possivelmente a maneira mais eficiente de reduzir o consumo de potência, visto que a tensão possui uma relação quadrática com o consumo de potência. No entanto, operar circuitos digitais em baixos níveis de tensão acarreta novas dificuldades e desafios que devem ser superados para que os ganhos em economia de potência sejam relevantes. Esse trabalho apresenta diretrizes para construir uma biblioteca de células voltadas para baixos níveis de tensão, especificamente no regime de tensão *near-threshold*, que foi escolhido sobre o regime de tensão *sub-threshold* por uma série de motivos abordados no texto.

Palavras-chave: ASIC, CMOS, Baixo consumo, Baixa tensão, Near-threshold, Biblioteca de Células.

LIST OF ABBREVIATIONS AND ACRONYMS

UFRGS	Universidade Federal do Rio Grande do Sul
CEITEC	Centro Nacional de Tecnologia Eletrônica Avançada
IC	Integrated Circuit
ASIC	Application-Specific Integrated Circuit
NMOS	N-Channel Metal-Oxide Semiconductor
PMOS	P-Channel Metal-Oxide Semiconductor
CMOS	Complementary Metal-Oxide Semiconductor
EDA	Electronic Design Automation
CAD	Computer-Aided Design
HDL	Hardware Description Language
PnR	Place-and-Route
PDK	Process Design Kit
PN ratio	PMOS width / NMOS width ratio

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

In the early days of IC design, circuits were manually designed and laid out. This trend began to change with the emergence of EDA tools, which allowed to design circuits with more density and complexity. These tools also dramatically reduced circuits' production time, hence allowing designers to focus on higher-level aspects of the design. Early EDA tools were very primitive, and a significant part of IC design was still done manually by the designers. The adaptation of graph theory and algorithms to the IC area was another important step, as it led to the appearance of PnR (place-and-route) tools which also cut design time significantly. In the mid-80s, hardware description languages (HDLs) such as VHDL and Verilog were developed. Logic synthesis tools enabled designers to describe the design into a high-level language and obtain a description in terms of gates and transistors (netlist). Logic synthesis further deepened the automation process of IC design. Currently, there is a vast array of EDA tools, presenting a large range of functions such as logical synthesis, timing and power consumption estimation, PnR algorithms, behavioral and electrical simulation, electrical characterization and automated testbench generator.

The first ASICs (Application-Specific Integrated Circuit) began to be developed in the 80s. They were produced using gate array technology, which consists of unconnected array of transistors. Devices were connected by mapping the design logic to the gate array, using PnR tools sold by gate array vendors. Such tools would try to determine the optimal pathing to produce the logic within the gate array. Since gate arrays had a fixed number of transistors, its usage was simple but very inefficient. If a customer had a small logic, and the vendor only offered chips with a large number of transistors (much more than the customer's logic required, for example), the customer would have to acquire an unnecessarily large solution. Therefore, gate array ASICs had a fairly high cost per unit. The next important technique in ASIC design was the standard cell approach.

A standard cell is a network of transistors and interconnections arranged to yield a Boolean logic function of its inputs, e.g., inverter, and, or, exclusive or, addition and multiplication. The synthesis is done by translating the desired logic (e.g., described through a HDL such as VHDL or Verilog) to an intermediate description (the netlist) of the present standard cell functions. This step is called technology mapping. This netlist is then used to decide the physical position of standard cells and its interconnect wires, using place-and-route tools. The standard cell approach allowed to manufacture smaller, more efficient chips and it is still the most used methodology in ASIC design. A standard cell library is an ensemble of standard cells designed in a standard way, with one of its dimensions fixed, in order to ease the automation task of PnR. Besides having the layout and schematic files of the standard cells, a standard cell library generally features several other components such as parasitic extraction models, timing abstract and electrical characterization files.

1.1 Low-Power IC Design

Initial efforts on VLSI research were focused on the construction of increasingly faster circuits, and power consumption was considered only a secondary factor. Designers then realized that NMOS-logic was very power hungry and that digital circuits could not evolve for a much longer time if power consumption kept growing at a high rate, so IC started to be designed using CMOS logic, which has lower static power consumption and combines the advantages of both NMOS and PMOS logic. This power saving trend continued when battery-based applications which demanded high performance (like cell phones and laptops) became popular, beginning in the late-80s and early-90s. Using standard, off-the-shelf components to build such devices would require high power consumption, making it necessary to carry a large quantity of battery, which would make the use of those devices impracticable. The interest on low power circuits continued to increase as applications that required ultra-low power but had no or little concern on performance emerged, such as sensor networks and environmental monitoring equipment. Techniques for saving power vary from layout to circuit and architectural level. Low-power techniques will be further explored in the next chapter. Detailed related work of articles about low and ultra-low power are addressed in Chapter 3.

Among those techniques, voltage scaling is clearly the most efficient to obtain immediate results due to its quadratic relationship with dynamic power. Reducing supply voltage comes at a cost, though. In (CHANDRAKASAN; SHENG; BRODERSEN, 1992) and (CHANDRAKASAN; BRODERSEN, 1995) it is shown that a speed penalty is paid when using a reduced supply voltage. These penalties become even harsher when the supply voltage approaches or even surpasses the threshold voltage. Several other issues arise when devices are operating under the sub-threshold or near-threshold regimes, as circuits become dramatically sensitive to process, temperature and voltage variations (WANG; CHANDRAKASAN; KOSONOCKY, 2002). Issues introduced by operating in low voltage regime are discussed in more detail in the following chapters.

1.2 CTC LIB Project

Designers at CEITEC S.A. realized that the company would benefit from having its own standard cell library collection, rather than continue using the XFab 0.6μ standard cell libraries (XC06). The argument was that developing a standard cell library would give the designers more autonomy and flexibility, since they could modify or add new functionalities to the library at any time, which would not be possible if the company kept using XFab standard cell libraries. The project was then created and a partnership between CEITEC and UFRGS was established, and a group of engineers and students was assembled to execute the task, which would be funded by federal government agencies (CNPq and FINEP). The project had initial 1-year duration (from December 2013 to December 2014) but was extended to end in March 2016. The project

was named CTC LIB.

The project is also intertwined with the CI Brasil program. The goal of the program, which was approved in 2005, is to develop the semiconductors industry in Brazil. Newly graduated engineers receive training on the most used tools in the IC design market and at the end of the course, the students would work full-time at a design house involved with the project. The program would then produce a surplus of engineers prepared to work in the semicondutors industry, increasing national competitiveness in the area.

The standard cell libraries would be produced using XFab 0.6μ technology node, satisfying the demands of CEITEC S.A. engineers. As a first step, the main features the chosen technology, including voltage transfer characteristic (VTC), design rules and timing parameters, were studied and documented. Then, a primitive version of the library with no timing, area or power constraints was developed, which was finished by August 2014. Finally, in the last project phase, a series of libraries with different characteristics, namely, a low area cell library, a low-voltage cell library (which is the focus of this work) and a high speed cell library began to be developed. The final versions should be ready by the end of the project.

1.3 Work Proposal and Structure

Considering the scope of the CTC LIB project, the objective of this work is to discuss the construction of a low-voltage standard cell library to achieve reduced power consumption. For this purpose, challenges and advantages of working at very low voltages are considered. Then, an analysis of sub-threshold and near-threshold voltage operation characteristics is presented, as well as the differences between them and our motivation to have chosen near-threshold over sub-threshold logic. Next, we discuss guidelines and methodologies to design a standard cell library in the $0.6\mu\mathrm{m}$ technology, constructed to operate at the near-threshold voltage ($1.0\mathrm{V}$-$1.5\mathrm{V}$ for this technology) range, considering that the nominal voltage for this technology is $3.3\mathrm{V}$. Finally, the library is used to synthesize a series of different circuits and the results are shown and commented.

The work is organized as follows: in Chapter 2, a brief description of power dissipation components in CMOS devices and some of the strategies used to reduce power consumption with a special focus on low voltage operation, particularly sub-threshold and near-threshold voltage digital design. In Chapter 3, ASIC design flow (logical and physical synthesis) is addressed, as well as general guidelines to construct a standard cell library. In Chapter 4, related work is discussed and analyzed. In Chapter 5, work proposal, methodologies and implementation of the standard cell library are presented. Finally, Chapter 6 brings the conclusion and points future extension of this work.

2 POWER CONSUMPTION IN CMOS DEVICES AND LOW-POWER TECHNIQUES

Power consumption in CMOS devices has two major components: dynamic power and static power. They can be combined to yield the following equation:

$$P_{total} = P_{switching} + P_{short-circuit} + P_{leakage} \tag{2.1}$$

The switching and short-circuit components comprise the dynamic power consumption. The leakage component constitutes the majority of static power consumption and is composed by a rather large number of subcomponents. Both dynamic and static power will be discussed in the following subsections.

2.1 Dynamic Power

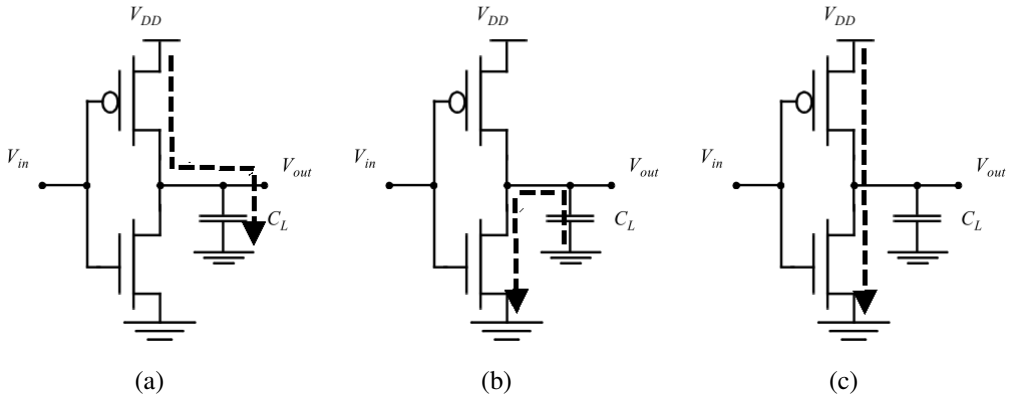
The dynamic power consumption has two main components: switching activity and dissipation due to short-circuit currents. Switching power dissipation happens during logic transitions, when the load capacitance is charged (from the supply voltage rail through the PMOS circuit to the load capacitance) or discharged (from the load capacitance through the NMOS circuit to ground). It is given by the following equation:

$$P_{switching} = p_{sw} \cdot C_L \cdot V_{dd}^2 \cdot f_{clk} \tag{2.2}$$

Where p_{sw} is the probability that a power-consuming transition (where the load capacitance is being charged, i.e., the output is shifting from '0' to '1') occurs in a clock period, C_L is the load capacitance, V_{dd} is the supply voltage and f_{clk} is the clock frequency. The product $p_{sw} \cdot C_L$ is also known as C_{eff} (effective capacitance), the average capacitance being switched per clock cycle. The p_{sw} factor is dependant of the logic function. Take a two-input NAND gate as an example. Considering four possible states ('00', '01', '10', '11') as a combination of the logic level of the inputs, the only combination that yields '0' as an output is '11', so the probability that the output is '0' (p_0) is $1/4$ and the probability that the output is '1' (p_1) is $3/4$. The probability that a power-consuming transition occurs is the product between p_0 and p_1. For the NAND gate, the p_{sw} factor is $1/4 \cdot 3/4 = 3/16$ and for a XOR gate, it is $1/2 \cdot 1/2 = 1/4$. The p_{sw} factor is also dependent on the logic style (for example, dynamic CMOS logic gates will have different p_{sw} values due to necessary pre-charge).

The second component of dynamic power consumption is dissipation due to the short-circuit current. The short-circuit current arises in the brief moment both the NMOS and PMOS networks are active. This fact occurs specifically when $V_{tn} < V_{in} < V_{dd} - |V_{tp}|$, where V_{tn} is the NMOS threshold voltage, V_{tp} is the PMOS threshold voltage, V_{in} is the input signal voltage and V_{dd} is the supply voltage. The interval of time in which this conductive path exists depends on transistor sizing, output load capacitance and specially on input and output rise and

Figure 2.1 – Currents that contribute to dynamic power dissipation. a) Capacitance being charged through PMOS network and b) discharged through NMOS network. c) Short-circuit current flowing through the direct path between supply and ground.



Source: Modified from (RABAEY; CHANDRAKASAN; NIKOLIC, 2002)

fall times. If input rise and fall delays are much larger than the output rise and fall delays, the conductive path will persist for a longer interval. Thus, it is important that input and output rise and fall times are as balanced as possible. In Figure 2.1 we can observe both switching and short-circuit components of dynamic power consumption. In the next subsection, static power will be addressed.

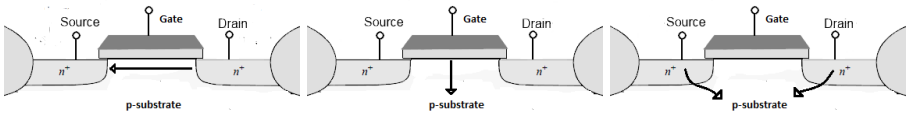
2.2 Static Power

Static power consumption is caused by the leakage component of power. Several factors contribute to leakage current, and their occurrence and intensity depends on the chosen technology node. The most important factors of leakage current are sub-threshold currents, gate-oxide tunneling and reverse-bias diode leakage, which are briefly described below.

The sub-threshold leakage current occurs when the gate-to-source voltage V_{gs} exceeds the weak inversion point (usually a very small voltage value) but does not exceed the threshold voltage V_t. In older technologies, sub-threshold current were considered negligible, or even zero in some cases, since the supply voltage was much higher and the strong inversion currents were many orders of magnitude higher than the sub-threshold leakage currents. In newer technologies, with reduced supply voltage and smaller currents, sub-threshold leakage is an important source of power dissipation and cannot be ignored.

The reverse-bias diode leakage current occurs when a transistor is not conducting but there is a voltage difference between drain and substrate. Consider a CMOS inverter with low input. The NMOS transistor will be turned off and the PMOS transistor will conduct current from supply to output, whose voltage will be driven low. Since the NMOS bulk is connected to the ground rail, and the drain is directly connected to the output, there will be a voltage difference

Figure 2.2 – Leakage current factors. a) Sub-threshold leakage current. b) Gate tunneling current c) Reverse bias junction current.



(a) (b) (c)

Source: Modified from (RABAEY; CHANDRAKASAN; NIKOLIC, 2002)

between drain and bulk, and a conductive path will arise between these terminals. The same phenomenon occurs in the PMOS transistor when it is turned off and the NMOS transistor is turned on. The reverse-bias diode leakage current depends on the drain diffusion area, leakage current density (which is dependent on the chosen technology) and supply voltage.

The third factor of leakage current is the gate-oxide tunneling current which is caused by aggressive gate oxide thickness downscaling to compensate supply voltage and transistor length reduction in newer technologies. The gate-oxide tunneling current flows from gate to substrate through the gate-oxide layer and it depends on the supply voltage, gate-oxide thickness and temperature. This effect can be mitigated by using a gate material with a high dielectric constant (also known as *high-K*). Since we are using a relatively old technology (0.6μ), gate-oxide tunneling current is not a significant cause for concern. The leakage current factors can be observed in Figure 2.2. Leakage power consumption can be summarized in the following equation:

$$P_{leakage} = I_{leakage} \cdot V_{dd} \tag{2.3}$$

2.3 Low-Power Techniques

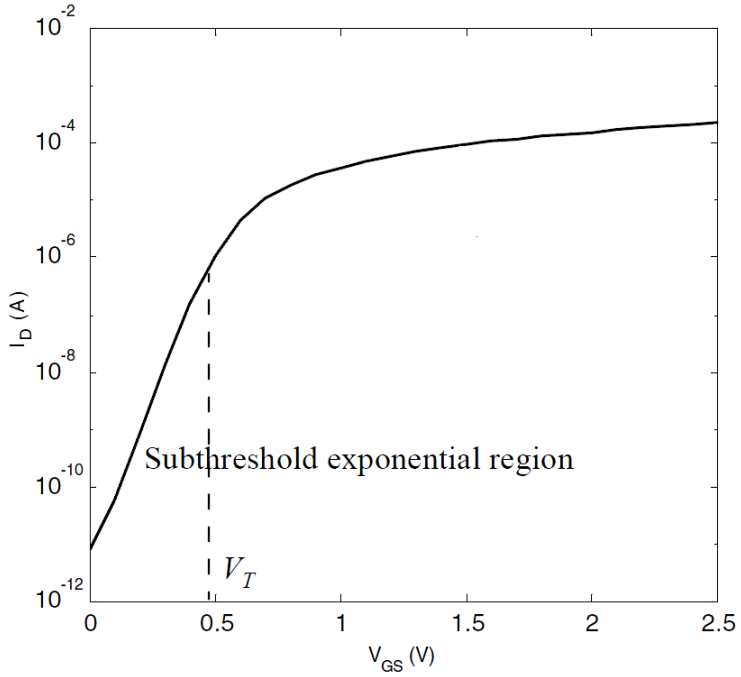
Techniques employed to reduce power consumption in digital IC design attempt to change one or more design parameters (capacitance, voltage and frequency) or add extra components to the circuit. Low-power techniques also differ from the level they are implemented. For example, clock gating, power gating and operand isolation can be employed by the designer in the RTL description of the circuit. Capacitance reduction and changes in logic style should be considered when the standard cell library is being developed. A brief analysis of low-power techniques (CHANDRAKASAN; SHENG; BRODERSEN, 1992; CHANDRAKASAN; BRODERSEN, 1995; PEDRAM, 1996) is presented below.

- Logic style. Using dynamic over static logic could reduce parasitic capacitance (less transistors) and the number of spurious transitions (also known as glitches), which are transitions that happen before the node settles down, mainly caused due to imbalanced paths in the network. However, circuits built using dynamic logic would have increased switch-

ing activity due to pre-charging and would consume extra clock power. Dynamic logic is also free of short-circuit currents. Alternatives such as pass-transistor logic (PTL) and complementary pass-transistor logic (CPL) have the advantage of using fewer transistors in some logic function implementations like XOR, adder and multiplexer gates, consequently reducing parasitic capacitance (CHANDRAKASAN; BRODERSEN, 1995).

- Implementation of logic functions. There are various ways to implement a given logic equation. For example, to implement a N-bit adder, we could simply use a ripple-carry adder, which is simple to build but could have long delays due to slow carry propagation (last 1-bit adder has to wait for the carry to pass through every intermediate 1-bit adder) and present a significant number of glitches (due to delays on carry propagation) which could affect the functionality of the circuit and cause several power consuming transitions. On the other hand, using carry-lookahead architecture would be faster, almost glitch-free but much more complex.
- Operand isolation. In operand isolation, certain parts of the circuit are kept from seeing its inputs which otherwise would produce unneeded, power-consuming logic swings. Let us suppose a circuit with an ALU composed of an adder and a multiplier. Even if we want to only one of them, both components would calculate the operation result as a function of its inputs, and the appropriate operation would be chosen. With operand isolation, a latch or multiplexer or another cell that can prevent a signal from propagating is added, and the unused operation is not realized.
- Power gating. Power gating is achieved by adding a switch that prevents the supply voltage from reaching certain parts of the circuit that will not be used for extended periods of time. The switch is usually a very large PMOS or NMOS transistor. Many types of power gating exist, and it is arguably the best technique to reduce leakage power.
- Clock gating. In circuits where there is little clock activity, it is interesting to turn registers off whenever they are not loading new data values. Since clock toggles in registers happen at every clock cycle, a significant part of dynamic power consumed comes from clock tree buffers. Clock gating adds extra circuitry to prune a part of the clock tree and prevents the clock signal from propagating so that the desired group of registers is disabled and does not switch states. Clock gating is not, however, an interesting technique to employ in high clock activity circuits, since registers would seldom be disabled.
- Capacitance reduction. From equation 2.2, it is observed that cell capacitance is an important source of dynamic power consumption. Therefore, capacitance minimization techniques may be employed if low-power cells are being constructed. The designer should aim to produce more compact cells, using less metal wires and minimum sized devices. Such measures will surely affect the resulting power consumption. However, certain precautions must be taken, since minimum sized devices can yield unbalanced rise and fall delays (the NMOS device generally drives more current than the PMOS device), degrad-

Figure 2.3 – I_d versus V_{gs} curve (in logarithmic scale) for a MOS device.



Source: Modified from (RABAEY; CHANDRAKASAN; NIKOLIC, 2002)

ing performance, so this trade-off must be considered.

- Voltage scaling. Reducing the supply voltage is the most effective way of minimizing power consumption, since it affects both dynamic and static power dissipation and has a quadratic relationship to dynamic power, but this reduction comes with a performance penalty. Also, a number of other issues arise when voltage is scaled near or under the threshold voltage, as transistors become more sensitive to process, environmental and aging variability.

Since the voltage scaling technique is one of the main points of our work, it will be discussed in more detail in the next section.

2.4 Low Voltage Operation

To understand how voltage scaling affects MOS device behavior, observe the graph in Figure 2.3, where the transistor drain current (I_d) is plotted against device gate voltage (V_{gs}) in a $0.25\mu\text{m}$ technology and a 2.5V maximum supply voltage.

Three different operating regions can be noticed. The first region, named weak inversion region, ranges from 0V to 0.45V (the device threshold voltage). In this region, the drain current has an exponential dependence on the supply voltage, and any small voltage variation could cause a large variation in the drain current. The second region ranges from 0.45V to approximately 0.9V. Here, the current decays rapidly (approximately in a quadratic fashion) but not as

fast as the drain current in the weak inversion region. Thus, we will call it moderate inversion region. In the last region (strong inversion region), which varies from $0.9\mathrm{V}$ onwards, variation in the supply voltage causes only a small change in drain current.

Such concepts are useful to define sub-threshold and near-threshold device operation, which will be discussed in the next subsections.

2.4.1 Sub-Threshold Voltage Operation

Sub-threshold operation is achieved when the supply voltage (V_{dd}) is reduced to below the threshold voltage (V_t), meaning the device is operating in the weak inversion region. As remarked above, the drain current in this region have an exponential relationship to the supply voltage. The current in the sub-threshold region is given by the following equation, as seen in (KWONG; CHANDRAKASAN, 2006):

$$I_d = I_o e^{\frac{V_{gs}-V_t+\eta V_{ds}}{nV_{th}}}(1 - e^{\frac{-V_{ds}}{V_{th}}}) \tag{2.4}$$

Where V_{gs} is the gate-source voltage, V_{ds} is the drain-source voltage, n is the sub-threshold slope factor, which measures by how much V_{gs} has to be reduced for the drain current to drop by a factor of 10, η is the DIBL (Drain-induced Barrier Lowering) coefficient and V_{th} is the thermal voltage. The current exponencial dependence on supply voltage implies that delays rapidly increase when the supply voltage approaches and goes below the threshold voltage. Hence, any deep voltage reduction must be accompanied by a frequency reduction, or the circuit will not function properly. This will limit sub-threshold voltage usage to applications in the low-to-moderate performance range. For example, in (WANG; CHANDRAKASAN; KOSONOCKY, 2002), the authors demonstrate that circuits designed using a $0.18\mu\mathrm{m}$ technology process operating in the sub-threshold voltage regime could operate in the $100\mathrm{kHz} - 10\mathrm{MHz}$ range, whilst a typical commercial microprocessor using this technology usually operates in the $800\mathrm{MHz}$-$1\mathrm{GHz}$ range.

Another important issue that arises when devices are operating in the sub-threshold region is parameter variability. Devices operating under these conditions are more susceptible to process, environmental (voltage and temperature) and aging variability, according to (WANG; CHANDRAKASAN; KOSONOCKY, 2002; CALHOUN; CHANDRAKASAN, 2004; KWONG; CHANDRAKASAN, 2006). In (STANGHERLIN, 2013) it is demonstrated that, for a device operating at $250\mathrm{mV}$ supply voltage in a $65\mathrm{nm}$ technology process, a 10% variation in the voltage supply, a $3 - \sigma$ process variation (where σ is the standard deviation from the mean) and temperature changing from $25°\mathrm{C}$ to $-40°$ could induce up to 99 times increased cycle time. For the same device operating at $150\mathrm{mV}$ voltage, going from the fast case with $125°\mathrm{C}$ temperature to the slow case with $-40°$ temperature could present a 1076 times increased cycle time. Also in (STANGHERLIN, 2013), agressive device upscaling is shown as a measure to mitigate such

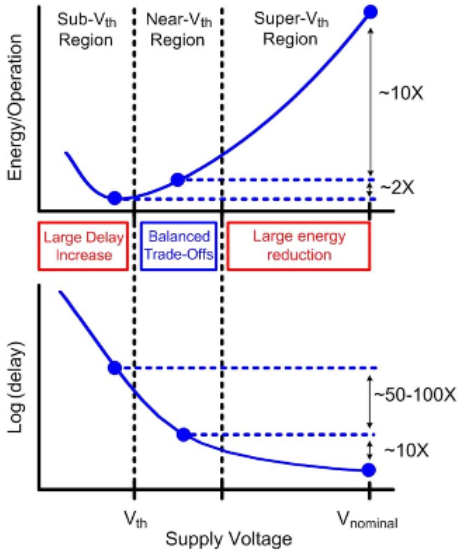
variability issues, as the author shows that threshold voltage variation is dependant on the area of the device. Therefore, increasing device width or even length is a possible solution to the variability problem.

Degraded logic swing due to reduced I_{on}/I_{off} ratio also has to be considered. The I_{on}/I_{off} ratio can be used to measure the functionality of a logic gate. The I_{on} current is the drive current of the device and the I_{off} is the current when the device is in idle mode. For some minimalist architectures, the pull-up and pull-down networks have an unbalanced number of devices, which can in some situations result in high parallel leakage current that reduces the I_{on}/I_{off} ratio, resulting in the propagation of weak zeroes or ones. In (WANG; CHANDRAKASAN, 2005), the authors give the tiny XOR architecture as an example. The architecture has three parallel devices in the pull-down network and one device in the pull-up network. When the three parallel pull-down devices are not operating and the pull-up device is driving current, the devices present parallel leakage currents that results in a degraded signal in the output. A feasible solution is to use a transmission gate XOR, which has balanced pull-up and pull-down parallel devices.

2.4.2 Near-Threshold Voltage Operation

Considering the challenges that sub-threshold circuits impose, only a very limited range of products will truly benefit from the ultra-low voltage operation regime, and the designer has to consider if it is really worth to operate digital IC under such harsh conditions. According to (DRESLINSKI et al., 2010), the moderate inversion region offers balanced trade-offs between delay and energy, as observed in Figure 2.4.

Figure 2.4 – Energy and delay in different voltage operationg regions.



Source: (DRESLINSKI et al., 2010)

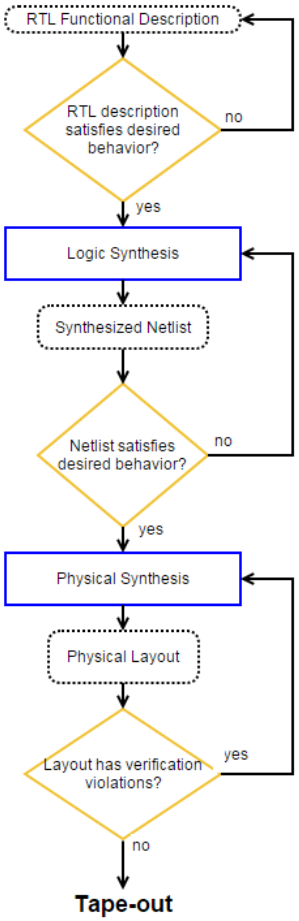
The authors argue that delay increases about 50 to 100 times when voltage is scaled from the

near-threshold region to the sub-threshold region, while consumed energy is reduced by only 2 times. Near-threshold circuits are also far more robust than sub-threshold circuits, presenting fewer variability issues (though they are still significant) and almost no swing degradation due to leakage currents. Thus, mitigation variability techniques such as device upscaling do not have to be so aggressive, resulting in capacitance reduction and consequently, less power consumption. Having higher frequency also means that they can be used in a broader set of applications. Also in (DRESLINSKI et al., 2010), the authors proposes the use of parallelization and device optimization methods in near-threshold circuits to obtain operating frequencies as high as when operating in nominal supply voltage while retaining very low power consumption. Using parallelization techniques in the sub-threshold to obtain the same results would require a higher density of cells, increasing area and capacitance of the final circuit. Hence, in many cases, operating in the near-threshold voltage is more efficient than in the sub-threshold voltage. In the case of the chosen XFab $0.6\mu\mathrm{m}$ technology process, the sub-threshold region is defined as the 0-$0.9\mathrm{V}$ range and the near-threshold region is defined as the $0.9\mathrm{V}$-$1.5\mathrm{V}$ interval.

3 STANDARD CELL DIGITAL DESIGN

Modern standard cell-based digital IC is constructed using a series of steps called the ASIC design flow, which can be seen on Figure 3.1. The ASIC design flow can be divided into three major parts: specification and RTL description, logic synthesis and physical synthesis. The first part is fairly straightforward. First, the designers have to understand the proposed problem and elaborate a high-level model that satisfies the imposed conditions. At this step, the designer is generally not interested into transistor-level logic or electrical characteristics, so it is common to use HDL languages (such as Verilog or VHDL) to describe the circuit. These languages allow the designer to build small combinational or sequential modules, define its internal logic (using either Boolean logic or loop statements like for and switch/case) and specify how the modules' inputs and outputs will be connected in other modules. This type of description is called RTL (Register-Transfer Level) description. Logic and physical synthesis are discussed in sections 3.1 and 3.2, respectively. In 3.3, we present an overview of the general structure and design flow of a standard cell library.

Figure 3.1 – Simplified ASIC design flow.



3.1 Logic Synthesis

In logic synthesis, the RTL description previously constructed is converted into an optimized netlist (in terms of standard cells and design constraints) description. The logic synthesis can be divided into three steps: generic mapping, technology mapping and optimization phase. Generic mapping is a step in which the RTL description is translated into a system of Boolean equations or generic leaf cells which contains no information about power, timing or area, just the logic function (usually simple and primitive, like NAND or NOR gates). Generic mapping is important to perform constraint-independent logic optimization and minimization and is independent of technology. In technology mapping, the simplified logic produced by generic mapping is transformed to match the components of the standard cell library. This transformation is carefully made to satisfy the given constraints. Optimizations are then made to further adjust the design. Logic synthesis has generally three inputs:

- RTL description. The designer should provide the HDL files that describe the desired logic.
- Constraints file. This file specifies the clock period, arbitrary clock delays, and other parameters that indicate how the synthesis will be performed. The most used constraint file extension is the .sdc (Synopsys Design Constraint) format.
- Timing and power models of the standard cell library. Logic synthesis requires the .lib (Liberty Timing Format) file, which is generated in electrical characterization and contains information about timing and power of the library cells when several device parameters are allowed to vary. Both electrical characterization and the .lib file will be discussed in Section 3.3.

The output is generally the synthesized netlist, in which the original logic is written using the functions available in the chosen standard cell library.

3.2 Physical Synthesis

Physical Synthesis is the stage in which the synthesized netlist is transformed into a physical description of the logic circuit, including information about the location of the input and output pads, interconnect wires and standard cells. It is divided into several small steps, and the most important of them are listed below.

- Floorplanning and Powerplanning. In the floorplanning phase, parts of the circuit that should be closely placed in order to meet timing constraints are identified. An estimation of the circuit area is also calculated, taking into account the space needed for routing wires and macro blocks (such as memory and other IP blocks). The placement of the input and output pads as well as the aspect ratio is also decided here. A well-done floorplanning is essential to yield good results in the placement and routing phases. In powerplanning, the

number and direction of the rings, stripes and rails are determined. The rings carry power supply around the circuit. The stripes carry the power from the rings across the chip and the rails carry the power from stripes to standard cells.

- Placement. In this step, the physical location of each cell is determined. The placement must be done respecting the connectivity between cells and thus yield a routable chip that satisfies timing and area constraints. Placement is a particularly challenging step since there is an enormous number of cells to place in a limited area.
- Routing. To perform routing, metal wires are placed between the cells. The terminals that should be connected are indicated in the synthesized netlist. The quality of the routing depends on the resulting placement. For example, if two terminals that should be connected are not directly facing each other, the metal wire might have to go around the cell, wasting area. Routing is divided in global and local routing. In global routing, the routing regions are defined, and in local routing the layout and exact path of the wires is decided. Routing has to respect timing constraints, so the addition of buffers along the path might be needed.
- Clock Tree Synthesis. Digital ICs are usually synchronous, which means that the operating frequency is defined by one or more clock sources (e.g., external clock, oscillators). To improve the synchronization and ensure that timing constraints are satisfied, the clock signal should arrive in each sequential cell at approximately the same time, i.e., the clock skew (the difference between the maximum and minimum clock delay) should be the closest possible to zero. In Clock Tree Synthesis (CTS), buffers of different drive strengths are added along the path from the clock source to the sinks (cells that receive a clock signal) to reduce clock skew.

The final steps involve passing through a series of verification routines such as signal integrity analysis, IR drop analysis, sign-off analysis, formal verification and physical verification (DRC and LVS tests, which will be discussed in section 3.3). The design is then ready to be taped-out (sent to the foundry to be produced).

3.3 Standard Cell Library

In this section, we discuss the general structure and development of a standard cell library. First, if the designer is interested in creating a standard cell library, generally certain elements are needed, namely:

- Process design kit (PDK). The PDK is an ensemble of files containing information about a certain technology process (for example, IBM $130\mathrm{nm}$ and AMS $0.35\mu\mathrm{m}$). It usually comes with the design rules check (DRC) file (which contains a set of geometrical rules regarding minimum or maximum distance between layers of different materials within a cell layout), and mathematical models of MOS transistors (the BSIM3 model is one of

the most popular), diodes, capacitors and resistors, among others. The PDK also comes with a standard cell library designed by the foundry.

- CAD software to design custom IC. Tools for developing custom layout (such as Cadence Virtuoso and Synopsys Galaxy Custom Designer) are used in every development phase (schematic, layout, extraction and simulation). Usage of such tools is usually not intuitive, and designers must often receive special training to be able to use them succesfully.

Second, the designer should familiarize itself with aspects of the technology, such as maximum supply voltage, device threshold voltage, number of metal layers available and NMOS to PMOS relative drive strength (to calculate the optimal PN ratio for the chosen technology node). The NMOS device conducts a larger quantity of current than the PMOS device due to a difference of mobility. The NMOS device uses electrons as its signal carrier, while the PMOS device used holes (lack of electrons) as its carrier, which are slower than electrons (RABAEY; CHANDRAKASAN; NIKOLIC, 2002).

The next step is the selection of which logic functions will be present in the library. As seen in (SULISTYO, 2000), the library must have a minimum number of functions to be able to synthesize any design. This minimum is the concept of functional completeness. The required functions depend on the software vendor (for example, Cadence tools need different functions to realize logic synthesis in comparison to Synopsys tools). Generally, the library requires the three basic Boolean logic operations: negation (inverter gate), conjunction (AND or NAND gate) and disjunction (OR or NOR gate). Since most tools are unable to build complex sequential and tri-state logic from basic combinational gates, the designer must include some kind of tri-state element, a D-type latch and a D-type register with at least a set or reset trigger.

The designer might also include more complex cells, such as adders, multipliers, decoders or multiplexers or let the tool synthesize them by itself. A library with a very large number of functions may have redundant, unnecessary cells. On the other hand, a minimalist cell library might lack optimized complex logic which the tools could not be able to produce as well as a designer, by hand. It is common for designers to include exclusive disjunction logic (XOR and XNOR) functions, and-or-invert (AOI) and or-and-invert (OAI) logic functions in the library, among others, not to mention several types of latches and registers, and cells with different drive strengths. In addition to logic functions, the library needs cells that are used to satisfy certain power or timing constraints or in the fabrication process, such as input and output pads, filler and antenna cells and clock buffers.

The next step is netlist or schematic design. A netlist can be represented by a directed graph where each node is an instance of a circuit element (e.g., transistors, resistors, capacitors) and the edges represent the connections between the elements. Netlists also include input and output pins. Using netlists to implement logic functions involves the description of a transistor network, its interconnections and the input and output pins. To perform netlist simulation, the designer can build a small circuit in which he describes the type of stimulus (for digital design,

square waves are the most common) and the associated delays. Netlist simulation is fairly accurate, since it employs precise mathematical transistor models but does not consider wire delay and capacitance. Netlists can be described textually (for example, SPICE and Spectre languages) or using a GUI editor (for instance, Cadence Virtuoso Schematic Editor).

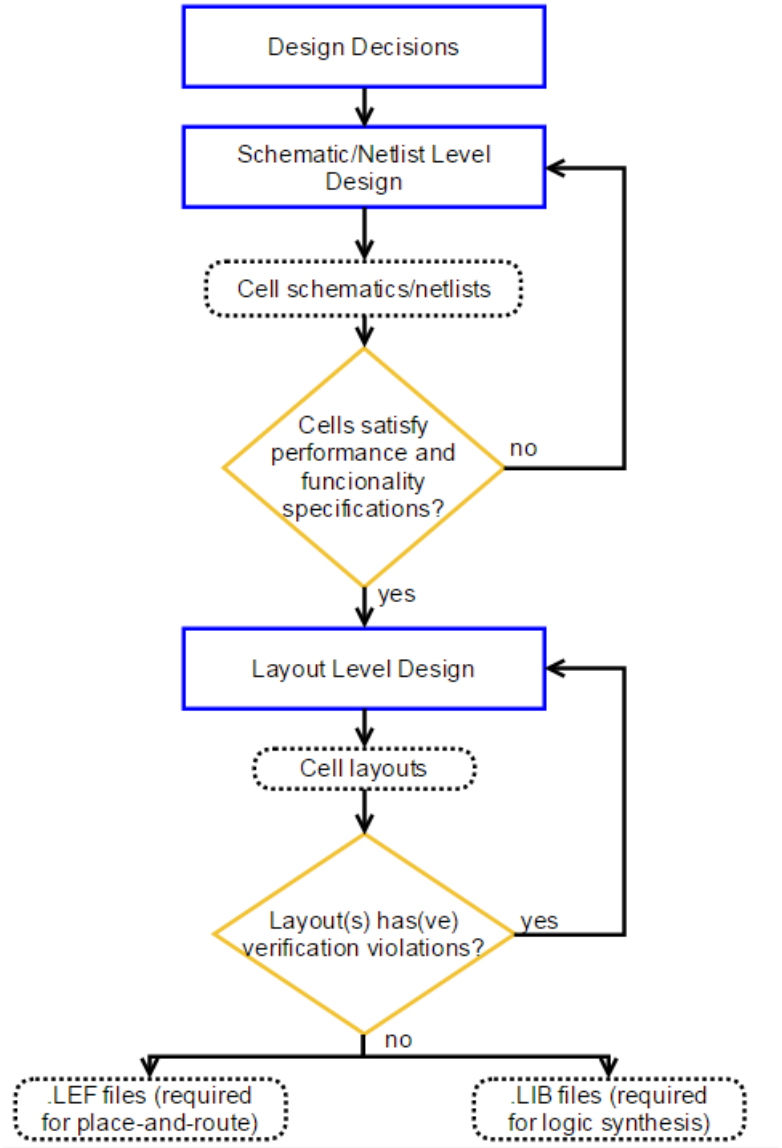
The final design step is the layout level design, meaning the designer must utilize custom layout tools to draw standard cells. It is important that the designer is familiarized with the electrical and physical characteristics of materials used by the technology process. He should be able to use different metal layers, polysilicon, n and p-type implants, n and p-type wells and diffusion to construct logic equivalent to that in the netlist.

The layout should comply with the rules specified in the Design Rules Check (DRC) file. This file is a list of geometric constraints such as minimum shape width, minimum spacing between the same or different materials, minimum transistor length, contact dimensions. The exact values vary depending on the fabrication process and the foundry. CAD software frequently provide options to notify or even enforce design rules during the layout design. To facilitate placement and routing, cells must be designed in a regular fashion. Usually, one of the cell dimensions is fixed and the other is flexible. Both width and height are defined in terms of the routing metal layer pitch. Routing rails are set down with even spaces between them. For example, if the vertical routing rails have a width of $0.8\mu\mathrm{m}$ and a spacing of $1.5\mu\mathrm{m}$, the distance from the center of a rail to the one immediately beside it is $2.3\mu\mathrm{m}$. This value is the vertical metal pitch. Metal layers usually run in perpendicular directions (for instance, METAL1 layer runs horizontally and METAL2 layer runs vertically). Pins should be placed upon the grid (in a 2-layer technology, it is common to place pins where the METAL1 track crosses the METAL2 track). Power supply rails should also have the same width or height (depending on which dimension is fixed) and run in the same direction.

After a cell layout is done, it has to go through a series of verification steps. In the DRC phase, the design is checked against the Design Rules Check (DRC) file. If no geometric rule is violated, the LVS (Layout versus Schematic) test is launched. The LVS compares the layout to the previously described netlist to check if the logic is equivalent. Then, the design goes through layout extraction to obtain a detailed and accurate model that takes into consideration the parasitic devices (parasitic resistances, capacitances and inductances) of the cell. Finally, the layout is exported to a .lef (Layout Exchange Format) file. The .lef file contains partial information about the layout, specifically pin and metal layer coordinates. This simplified model is helpful so that PnR tools do not need to load the whole layout, just what is relevant to routing.

The extracted netlists obtained in the last step are used in the library characterization, the last development phase. Characterization tools are employed to verificate the functionality of a design with different input parameters such as temperature, supply and device threshold voltage. For instance, in a circuit designed to operate at $3.3\mathrm{V}$ supply voltage, we could simulate a typical case ($3.3\mathrm{V}$ voltage, $25^{\circ}\mathrm{C}$ temperature and nominal threshold voltage), a fast case ($3.6\mathrm{V}$ voltage, $-40^{\circ}\mathrm{C}$ temperature and 10% decreased threshold voltage on both devices) and a slow

Figure 3.2 – Simplified standard cell library design flow.



case ($3.0\mathrm{V}$ voltage, $125°\mathrm{C}$ temperature and 10% increased threshold voltage on both devices). Extreme test cases are called corners. Examples of corners are SS (slow PMOS and NMOS devices), FF (both devices fast), SF (fast PMOS and slow NMOS), FS (slow PMOS and fast NMOS) and TT (typical). Library characterization can be used to simulate a great variety of situations that the circuit might go through in real circunstances. Usually, the tool generates a .lib (Liberty Timing Format) file which contains the delay, power and capacitance estimation of every gate in the standard cell library. The .lib file is an ASCII file which has detailed information about the realized simulation and is used in the logic synthesis phase to obtain an accurate initial estimation of the characteristics of the circuit.

Finally, the library has every element (electrical netlists, layouts, .lef and .lib files, among others) required to synthesize digital circuits. In Figure 3.2 a simplified design flow for a standard cell library can be observed.

4 RELATED WORK

In (CHANDRAKASAN; SHENG; BRODERSEN, 1992) and (CHANDRAKASAN; BRODERSEN, 1995), the authors present a compilation of low power design aspects. They discuss about areas in which consuming less power could be very useful and briefly describes power consumption in CMOS circuits. Then, some of the most important low power techniques are presented and analyzed. Finally, they suggest that the negative performance effects of voltage scaling can be neutralized in arithmetical circuits (e.g., adders and multipliers) using a mixed architecture consisting of parallel and pipelined components which would introduce an area overhead but would have the same throughput while using a reduced supply voltage. It is argued that this architecture can be extremely useful in DSP and circuits with heavy mathematical activity. Power consumption and low power techniques are also discussed in (PEDRAM, 1996). The article focus is on power estimation at various levels (circuit, logic and behavioral). He divides those techniques into simulation based and non-simulation based. In the low power section, the focus is on techniques that can be implemented by CAD software.

Generic guidelines to build a standard cell library are presented in (SULISTYO, 2000). The author introduces the concept of functional completeness covered in Chapter 3 and talks about the three steps of development of a cell library, namely netlist level design and simulation, layout level design and simulation and porting to synthesis/simulation libraries and PnR libraries. He also provides a review of standard cell based ASIC design. He then presents a series of techniques to build regular standard cells. For example, metal tracks of the same layer must have the same width in all cells of the library; supply rails should run in the same direction; pins should be placed over the routing grid, among other strategies. Next, the author discusses characterization of delay and capacitance of logic cells. For simplicity, he utilizes the linear delay model (which consists of simple equations, is a good approximation of the real delay model and is used by most commercial synthesis tools). He presents a list of methods to calculate intrinsic, transition and slope delays, setup time in sequential cells, input/output capacitance in both combinational and sequential cells and techniques to calculate the previous parameters in tri-state cells. Every definition is accompanied by a SPICE example. Then, power characterization is discussed. The author introduces the basics of power dissipation (static and dynamic power) in CMOS devices and the Synopsys model of power dissipation to estimate static and dynamic power consumption. He finishes the chapter by talking about his own method to estimate power consumption in combinational, sequential and tri-state cells (again, SPICE examples are given in each definition). Finally, in the last chapter, he provides an example RTL description (using VHDL) of a twin 4-bit counter that has been synthesized using a standard cell library designed according to the development flow proposed in the article.

Sub-threshold voltage operation is discussed in (SOELEMAN; ROY, 1999), (SOELEMAN; ROY, 2000), (SOELEMAN; ROY; PAUL, 2001), (**??**), (WANG; CHANDRAKASAN; KOSONOCKY, 2002), (CALHOUN; CHANDRAKASAN, 2004), (WANG; CHANDRAKASAN, 2005) and

(KWONG; CHANDRAKASAN, 2006). In (SOELEMAN; ROY, 1999) and (SOELEMAN; ROY, 2000), the authors present a sub-threshold voltage strategy to implement ultra-low power digital IC. They demonstrate that when operating under the sub-threshold voltage, the power delay product (PDP, the amount of energy per transition) is lower than when operating in the normal strong inversion region, since the reduction in power consumption outweighs the increase in delay. It is also shown that sub-threshold circuits are much more prone to variation in the supply voltage. To conclude, pseudo-NMOS logic is suggested as an alternative to CMOS in the sub-threshold region, considering it has a speed improvement and less area usage. The same authors present two different sub-threshold logic families in (SOELEMAN; ROY; PAUL, 2001). They were designed to offer more robustness and tolerance to parameter variability than standard CMOS logic. The variable threshold voltage sub-threshold CMOS (VT-Sub-CMOS) logic is similar to the conventional CMOS logic with the addition of a stabilization circuit built to detect any abrupt variation of the current and minimize the effects of process and temperature variations. Sub-threshold dynamic threshold voltage MOS (sub-DTMOS) logic attempts to mitigate variations and achieve stability with direct body biasing by tying the gates to their substrates. In (**??**) the authors use the previous results to design an ultra-low power adaptive filter for hearing aid applications. The circuit was implement using pseudo-NMOS logic as suggested in (SOELEMAN; ROY, 2000).

The optimal voltage supply and threshold voltage points to achieve minimum energy operation in discussed in (WANG; CHANDRAKASAN; KOSONOCKY, 2002) and (CALHOUN; CHANDRAKASAN, 2004). The analysis result is used in (WANG; CHANDRAKASAN, 2005) to design a FFT processor that can operate down to $180\mathrm{mV}$. Results show that the processor is more energy-efficient than other generic circuits that offer similar solutions.

Device sizing is discussed in (SUTHERLAND; SPROULL, 1991), (KEANE et al., 2006) and (KWONG; CHANDRAKASAN, 2006). In (SUTHERLAND; SPROULL, 1991), the logical effort technique is presented. It consists in a method to estimate delay and optimal device sizing in CMOS circuits. His work is extended to sub-threshold circuits by Keane et al. in (KEANE et al., 2006). They demonstrated that transistors operating under the sub-threshold voltage present different characteristics from those operating in the strong inversion region, and takes these issues into account to derive the sub-threshold logical effort method. The main objective of both methods is to design fast circuits, constructing every cell to have speeds comparable to those of an inverter (theoretically the “fastest” gate). The values presented in the articles are not absolute — the designer may perform a series of simulations to determine the optimal PN ratio and device stack behavior for a certain technology. In (KWONG; CHANDRAKASAN, 2006), Kwong shows that even though the minimum energy operation point is reached when using minimum sized devices, they should be upsized since minimum sized devices are more susceptible to parameter variability. She suggests a sizing scheme using a butterfly plot to find the width ratio between single and stacked transistors. Kwong also extends her previous work to design a sub-threshold standard cell library in (KWONG, 2006) using a

65nm CMOS process technology. She discusses design decisions regarding selection of logic functions and drive strengths, as well as device sizing using the same methodology seen in (KWONG; CHANDRAKASAN, 2006). She also performs a detailed analysis of various register architectures operating under the sub-threshold voltage. Finally, the author synthesizes a fault-tolerant FIR filter with the previously constructed sub-threshold standard cell library and evaluates the simulation results.

In (STANGHERLIN, 2013), the author attempts to show the difficulties in digital IC design under the sub-threshold voltage regime, presents a number of advantages of near-threshold over sub-threshold voltage operation and proposes a standard cell library based on the voltage-frequency scaling (VFS) technique constructed to function in the near-threshold voltage regime that will also work well under the nominal voltage operation, presenting only minor loss of performance. In the first chapter, an analysis of fabrication process environmental (temperature and voltage) and aging variability and its effects on the performance of circuits operating on near-threshold and sub-threshold voltage regimes are discussed. A ring oscillator is used as a study case to evaluate these effects. The results show that CMOS circuits operating in lower voltage are much more sensible to process variation (and extremely sensible to temperature variations), and could operate, considering a worst-case scenario (high temperature variation, standard 10%voltage, and $3 - \sigma$ process variation), up to 99 times slower. Stangherlin also brings an overview of power consumption on CMOS devices and some of the strategies used to design low power CMOS devices. In the next chapter, the author defines near-threshold voltage operation and voltage-frequency scaling, and discusses the techniques used to build the proposed cell library. To achieve balanced rise/fall times and correct voltage swings and noise margins, the main concerns are transistor sizing and stacking limit. Finally, the cell library is used to synthesize a notch filter, an 8051-compatible core and a few benchmark sequential and combinational circuits, and the results are commented.

Near-threshold circuits are also discussed in (DRESLINSKI et al., 2010) and (MARKOVIC et al., 2010). The authors of (DRESLINSKI et al., 2010) present the advantages of near-threshold voltage operation over both sub- and super-threshold operation as well as the difficulties imposed by near-threshold digital design and techniques to mitigate or even overcome these issues. They claim that near-threshold operation can be used to design energy-efficient circuits while keeping delays at an acceptable level and reenable Moore's law in the semiconductor industry. In (MARKOVIC et al., 2010), Markovic et. al make similar observations about near-threshold voltage operation. They also present a delay model that fits both the weak and moderate inversion region. Results presented in the article show that a 20% energy increase from the minimum energy point (localized in the sub-threshold region) can give a 10 times performance increase. Then, the authors present a pass-transistor based logic style optimized to operate in the moderate inversion region. Parallelization, pipelining and time-multiplexing are also discussed as possible methods to mitigate performance loss in this region.

5 WORK PROPOSAL AND IMPLEMENTATION

Our goal in this work is to combine the knowledge about low-power techniques presented in Chapter 2 and design flow of standard cell libraries in Chapter 3 to construct a low-power standard cell library using low-voltage techniques. Considering the difficulties and benefits of both sub-threshold and near-threshold shown in Chapter 2 and the findings of previous articles showing several advantages of near-threshold logic over sub-threshold logic, we decided to design the library to operate in the near-threshold voltage operating region, which we will define in this technology as the 1.0V-1.5V voltage range. As mentioned before, the library will be constructed in the XFab $0.6\mu\text{m}$ technology process, as it is the process being used at CEITEC S.A. The detailed development flow consists of four steps (logic function selection, netlist or schematic level design, layout level design and electrical characterization) and its substeps. They are presented in the next sections. In the end, results obtained for synthesized circuits are commented.

5.1 Logic Function Selection

The chosen logic functions were based on the function list of the typical and low area CTC LIB project libraries. The functions present in these libraries are in compliance with the desired functions by the CEITEC S.A. designers, so we shall keep our function list as close as possible. The CTC LIB libraries contain 106 combinational cells, such as exclusive or and nor, and-or-invert (AOI) and or-and-invert (OAI), multiplexers, half- and full-adders, 4 tri-state cells (inverters and buffers) and 18 sequential cells, including register with scan and clock gating. The cells will be designed in a progressive way: first, basic cells to satisfy the functional completeness, i.e., an inverter, 2-input NAND and NOR gates, a tri-state inverter, a D-type flip-flop with set trigger, a D-type flip-flop with reset trigger and any D-type latch. Then, functions with larger stacks (3- and 4- input NAND and NOR gates), complex combinational gates (XOR and XNOR, AOI and OAI), inverters and buffers (both normal and tri-state) of different drive strengths, and functions with negated inputs. Finally, the last combinational cells (multiplexers, half- and full-adder) will be implemented, since they have multiple stages and several possible implementations which may or may not work properly at a lower supply voltage. Physical (antenna, filler, tie) cells also have to be developed. The partial function list can be found in Appendix A.

5.2 Netlist Level Design

In the netlist (or schematic) level, the designer can make use of powerful and accurate device models to simulate any desired parameter variation, including supply voltage, device width and temperature. For this purpose, a wide range of electrical simulators is available. The most

acclaimed electrical simulator in the digital design area is Synopsys HSPICE. Other excellent alternatives exist, such as Cadence Spectre, which we will be using in this work. In this section, we first address device sizing in combinational gates, showing the obtained results. Then, power consumption for certain combinational cells is estimated at several voltage values using a ring oscillator with the values obtained at the first step. Finally, the register architecture for the library is discussed.

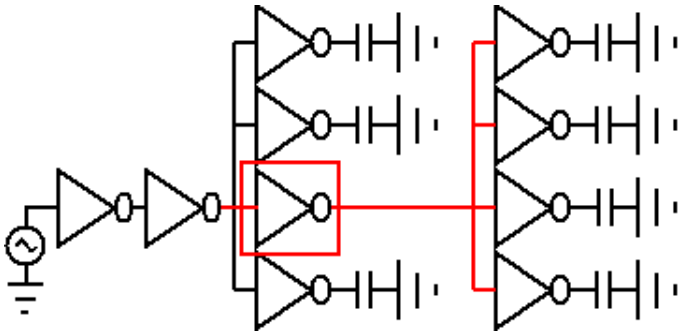
5.2.1 Combinational Cell Device Sizing Analysis

Both the logical effort and the sub-threshold logical effort methods aim to produce circuits with high performance. Since our goal in this work is to construct a low-power standard cell library, trade-offs between speed and power consumption must be considered. Thus, we shall adopt a different device sizing methodology, one that considers power consumption its primary target. An obvious solution would be to use minimum sized devices, since it would be possible to produce gates with the least possible capacitance, reducing power consumption. Unfortunately, given that CMOS circuits will often have stacked devices in only one of the networks (or stacks with fewer transistors in the other network) and NMOS and PMOS devices have different drive strengths, cells built with minimum sized devices will possibly have very unbalanced rise and fall delays, significantly degrading performance. Therefore, we propose a sizing methodology similar to the strategy seen in (STANGHERLIN, 2013): starting with minimum sized devices (PN ratio 1), the PMOS device width is increased until the ratio that equalizes rise and fall delays is found. Balanced rise and fall times The method is used to find the optimal PN ratio of an inverter and repeated for cells with similar characteristics to find how much PMOS stacked devices must be upscaled. This strategy guarantees that NMOS devices will be kept at the minimum possible width while the PMOS device is varied, unlike the logical and sub-threshold logical effort methods, since they assume variable NMOS device width. We shall introduce a subtle difference to the proposed sizing methodology: to design cells with increased drive strength, the author proposes upscaling every transistor in the cell by the desired factor. We propose the addition of a buffer (in the case of a function with inverted output, upscaling the output inverter is sufficient to produce higher drive strength) with its transistors upscaled by the factor.

To evaluate our chosen device sizing methodology (explained in more in the beginning of this chapter), a scheme similar to the one in Figure 5.1 was used.

In the figure, rise and fall delays of the gate (in the example case, an inverter) inside the red box are observed. This inverter has a fan-out of 4, and an equivalent input slope, since the leftmost inverter is also driving four equally sized inverters. To simulate the effects of parameter variation over the rise and fall delays of the gate, we set the supply voltage at $1.0\mathrm{V}$ (the intended voltage value for this library) and vary the PMOS device width from the minimum size ($0.8\mu\mathrm{m}$) to a maximum size of $4.0\mu\mathrm{m}$ using $0.1\mu\mathrm{m}$ increments to find the PN ratio that best

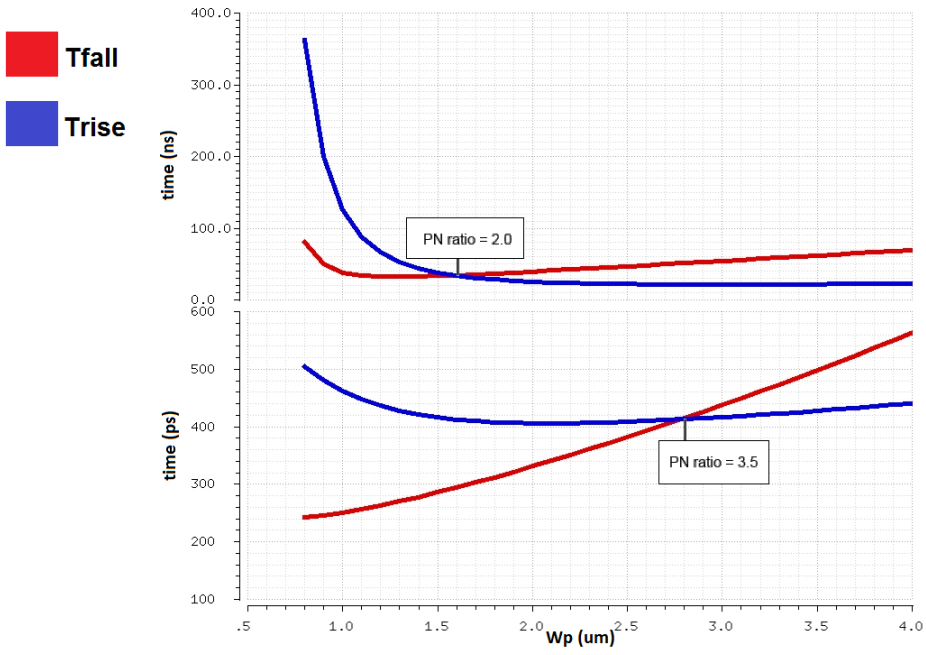
Figure 5.1 – Circuit used to measure rise and fall delays in an inverter.



approximates rise and fall delays. In this case we consider that a gate that requires more than $4.0\mu m$ to approximate the rise and fall delays won't be included in the library mainly because large capacitances increase dynamic power consumption and very high PN ratio may cause serious static current issues, specially when operating at lower voltages. Then, we repeat the same method using a voltage of $3.3V$ (the typical value for this technology) to find the desired PN ratio for this voltage. Then, the circuit is simulated using both PN ratios found (for $1.0V$ and for $3.3V$) and varying the voltage from $1.0V$ to $3.3V$ using increments of $0.1V$ per iteration in order to observe the behavior of the circuit when the voltage is scaled and how each sizing factor affects the rise and fall delays. These analyses are possible using the `swp` Spectre command, which allows an iteration over a set of specified values.

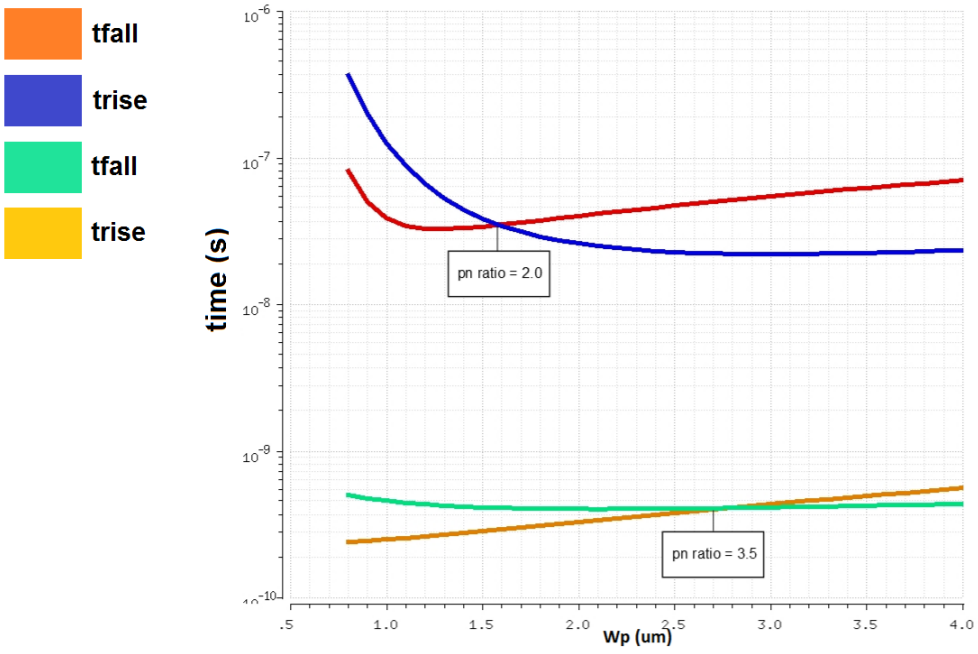
The first simulated gate was the inverter, as it is the simplest function with no device stacks, and it allows us to understand the direct relationship of a single PMOS device to a single NMOS device in this technology. Observe the graphs on Figure 5.2, where the PMOS device width is increased from $0.8\mu m$ to $4.0\mu m$ with the voltage set at $1.0V$ and $3.3V$, respectively:

Figure 5.2 – Rise and fall times versus W_p curves for a CMOS inverter. a) $V_{dd} = 1.0V$. b) $V_{dd} = 3.3V$.



In the graphs it is possible to indicate the PMOS device width value that best approximates the rise and fall delays for both 1.0V and 3.3V. As expected, the delays are much larger when the circuit is operating at 1.0V, ranging from approximately 30ns to 400ns. The delay values in 3.3Vrange from 0.2ns to around 0.6ns, almost three orders of magnitude smaller. The values are compared in Figure 5.3, where the curves are given in logarithmic scale. The blue and red curves are the rise and fall times of the inverter using a 1.0V supply voltage, and the green and orange use a 3.3V voltage. The graphs also confirms our initial supposition that minimum sized devices have unbalanced rise and fall delays, as can be seen when the PMOS transistor has a width of $0.8\mu\text{m}$ in the 1.0V voltage case. In this circunstance, the rise time is approximately 360ns and the fall time is 80ns, around 4.5 times greater. For the 3.3V voltage case, this difference is not so evident, but still significant, with a rise time of 0.5ns and a fall time of 0.25ns. Finally, the PMOS device width value that gives the best rise/fall times balance for the 1.0V voltage case is about $1.6\mu\text{m}$, giving a PN ratio of 2 at 35.0ns rise and fall delays and circa $2.75\mu\text{m}$ with a 3.4375 PN ratio at 0.410ns rise and fall delays, for the 3.3V voltage case. Using the exact values would make layout design more complex, so we decided that devices would be sized as multiples of 0.1, meaning a value of 2.8 would be used in the 3.3V case. In Figure 5.4, the PMOS device width is set at $1.6\mu\text{m}$ (the optimal value previously found for 1.0V) and the voltage is increased from 1.0V to 3.3V. Even though the rise and fall delay difference slowly increases when the voltage is scaled up, the final rise and fall delay values for a 3.3V voltage are 0.416ns and 0.286ns, a difference of about 1.45 times, which means the gate will function reasonably well at higher voltages with only small performance loss.

Figure 5.3 – Rise and fall times versus W_p curves for a CMOS inverter in logarithmic scale. a) $V_{dd} = 1.0\text{V}$. b) $V_{dd} = 3.3\text{V}$.



The next studied gates were the 2-input NAND and NOR gates. For these gates, a slightly

modified version of Figure 5.1 was used. Figure 5.4 shows the circuit for the NAND gate. The leftmost inverter is present to provide a more realistic signal to the input load, unlike one produced directly by the square wave voltage generator. To make sure the NMOS device stack behavior was being observed, one of the inputs is set at the logic value '1' and the other input has its signal alternated between 0 and 1. The circuit for the NOR gate analysis is virtually the same, the single difference being that since we wish to observe the PMOS device stack behavior, one of the inputs is set at '0' instead of '1' and the other has its signal varied.

Figure 5.4 – Circuit used to measure rise and fall delays in a 2-input NAND gate.

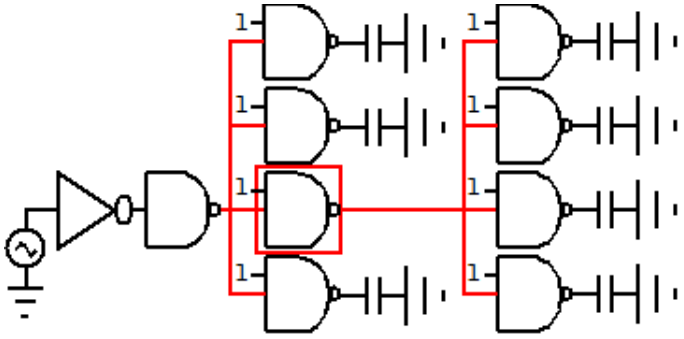


Table 5.1 shows the results obtained for the 2-input NAND and NOR gates.

-

Table 5.1 – Optimal PMOS width and PN ratio for the 2-input NAND and NOR gates.

Gate	*Voltage*	*PMOS width*	*PN ratio*	*Rise Time*	*Fall Time*
NAND-2	1.0V	1.2	1.5	76.561ns	69.886ns
	3.3V	1.8	2.25	0.512ns	0.512ns
NOR-2	1.0V	2.1	2.625	46.354ns	47.084ns
	3.3V	3.4	4.25	0.741ns	0.735ns

Again, the gates working at 1.0V need much smaller PN ratio values to balance rise and fall delays than the gates in the 3.3V case, meaning they are also more energy-efficient, since smaller devices have less overall capacitance and thus consume less dynamic power. In both cases, taking the inverter as the initial parameter, the NAND gate requires a smaller PMOS device width and the NOR gate requires a larger PMOS device width to approximate rise and fall delays. This result is expected – the NMOS device stack present in the NAND gate is weaker in comparison to the single NMOS transistor in the inverter, and the PMOS device stack in the NOR gate is also weaker than the single inverter PMOS transistor. The simulations were repeated for 3- and 4-input NAND and NOR gates, and the results demonstrate that our suppositions are correct. In other words, the 3 and 4 device stacks are further weakened relative to the 2 device stack. Such measurements are shown in Table 5.3.

Finding a PMOS-to-NMOS size factor and a NMOS or a PMOS device stack factor that can be generalized to any cell is a complicated task. Multi-stage networks, complex logic

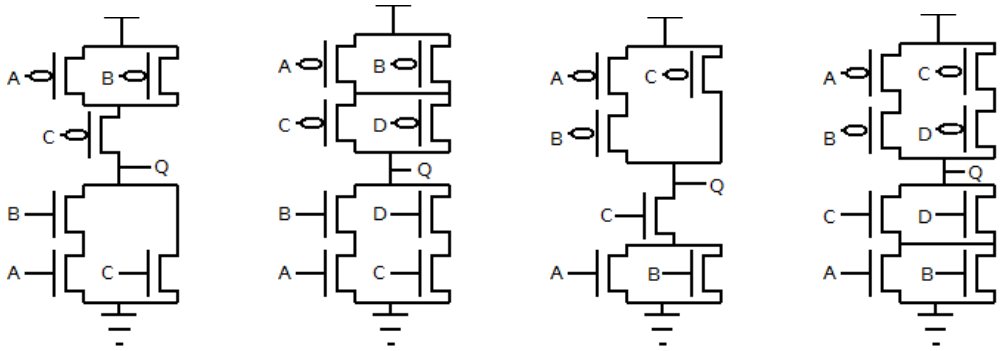
-

Table 5.2 – Optimal PMOS width and PN ratio for the 3 and 4-input NAND and NOR gates.

Gate	*Voltage*	*PMOS width*	*PN ratio*	*Rise Time*	*Fall Time*
NAND-3	1.0V	1.1	1.375	110.466ns	111.080ns
	3.3V	1.4	1.75	0.614ns	0.629ns
NOR-3	1.0V	2.6	3.25	62.795ns	66.431ns
	3.3V	4.3	5.375	1.172ns	1.181ns
NAND-4	1.0V	1.0	1.25	166.928ns	152.764ns
	3.3V	1.1	1.375	0.739ns	0.726ns
NOR-4	1.0V	2.9	3.625	72.695ns	73.369ns
	3.3V	4.9	6.125	1.545ns	1.531ns

and funcions with negated inputs or outputs are some of the obstacles preventing the usage of a unique, universal sizing factor. Therefore, we decided to include a different approach into our sizing methodology. First, we assumed that cells with the same stack size characteristics would yield similar rise and fall time curves. Hence, such cells would present an equivalent or approximate sizing factor to balance rise and fall delays. To verify our assumptions, we started by simulating a group of cells that contained a 2-device NMOS stack and a 2-device PMOS stack. In Figure 5.5 it is possible to observe the schematics of cells AN21, AN22, ON21 and ON22.

Figure 5.5 – Schematics of cells with 2-device NMOS and PMOS stacks. a) AN21. b) AN22. c) ON21. d) ON22.



(a) (b) (c) (d)

All of them fit the 2-device NMOS and PMOS stack criteria when they are specified with a certain input value combination. In cell AN21, for instance, if the A and B inputs receive an '1' logic value and the C inputs receives a '0', the PMOS network will be turned off and the NMOS network will conduct through the transistors in series connected to the A-B inputs. If the A and C input receive a '0' and the B input receives an '1' value, the PMOS network will conduct current through the stack formed by the devices connected to the A-C inputs. Gates were simulated using a circuit similar to the one in Figure 5.1, with the situations previously

described taken into consideration. The results for the $1.0\mathrm{V}$ supply voltage are present in Table 5.4.]

Table 5.3 – Optimal PMOS width and PN ratio cells with 2-device NMOS and PMOS stacks.

Gate	*Voltage*	*PMOS width*	*PN ratio*	*Rise Time*	*Fall Time*
AN21	1.0V	1.5	1.875	92.189ns	86.674ns
AN22	1.0V	1.5	1.875	93.919ns	86.830ns
ON21	1.0V	1.5	1.875	86.544ns	89.752ns
ON22	1.0V	1.5	1.875	87.848ns	92.533ns

As expected, the values are close. To verify if the specific 2-device PMOS and NMOS device stack case was not an isolated situation, we simulated another group of cells, this time satisfying the 3-device PMOS stack and 2-device NMOS stack criteria. The rise and fall times measurements for cells ON31, AN211, AN221 and AN222 can be observed in Table 5.4. Again, the same PN ratio is found for all of the simulated cells. The results show that we can use this approach to find the optimal ratio for cells with the same characteristics simulating only one or two cells within the group. The remaining gates will be divided into several categories according to the NMOS and PMOS stack size and the presence of inverters in the inputs or outputs. Complex, multi-stage functions such as adders and multiplexers will still be optimized individually. This concludes our study on combinational cell sizing.

Table 5.4 – Optimal PMOS width and PN ratio cells with 2-device NMOS and 3-device PMOS stack.

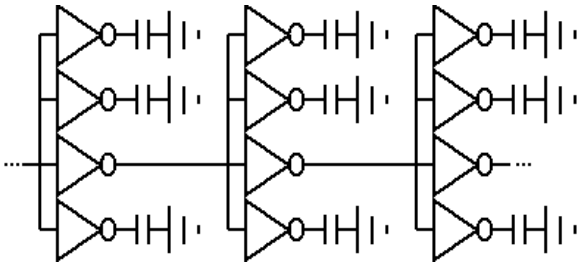
Gate	*Voltage*	*PMOS width*	*PN ratio*	*Rise Time*	*Fall Time*
ON31	1.0V	1.7	2.125	110.066ns	106.923ns
AN211	1.0V	1.7	2.125	106.755ns	99.878ns
AN221	1.0V	1.7	2.125	108.921ns	102.328ns
AN222	1.0V	1.7	2.125	118.461ns	112.004ns

5.2.2 Power and Delay Estimation Using a Ring Oscillator

Now, in order to measure how power consumption and performance behaves when the voltage is scaled from $1.0\mathrm{V}$ to $3.3\mathrm{V}$, three 19-stage ring oscillators were used, with each stage composed of 4 cells. The first oscillator consisted of inverters, the second of NAND3 cells and the third oscillator used NOR3 cells. The architecture can be observed in Figure 5.6.

The ring oscillator is an interesting architecture to estimate delays since we can calculate them individually for each cell if the total period and the number of stages is known. Considering that each stage is composed of the exact same cells, a formula for the gate delay can be

Figure 5.6 – 3 stages of a ring oscillator with inverters.



deduced:

$$d_g = \frac{T}{2.N} \tag{5.1}$$

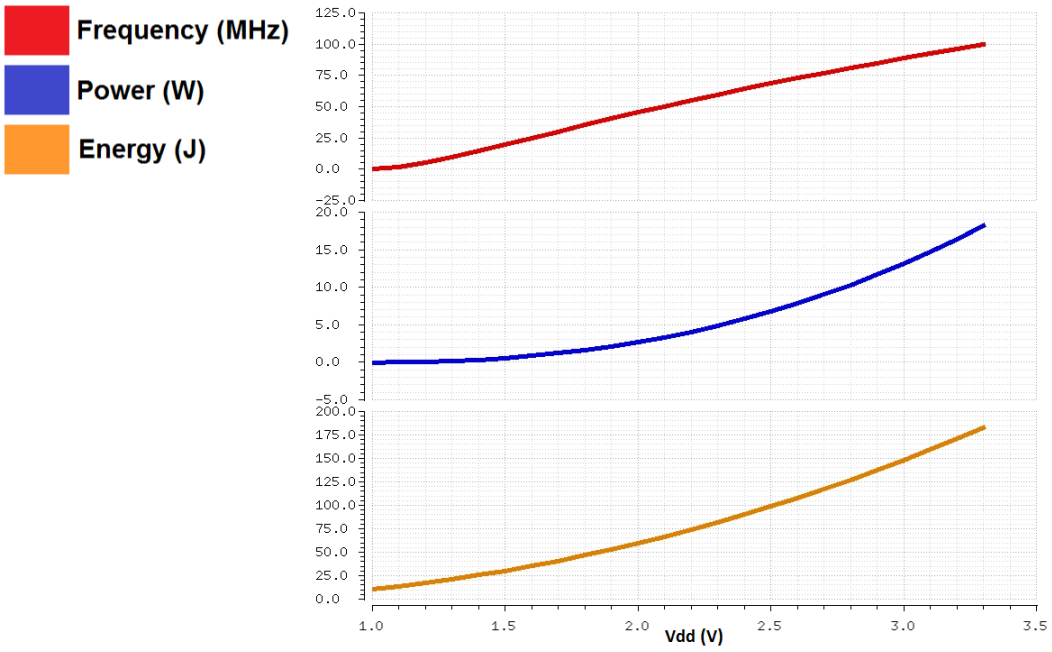
Where d_g is the gate delay, T is the total oscillation period and N is the number of stages. Now we can use either the total oscillation period or the individual gate period to estimate average power consumption over the specified time interval, by integrating the current over the period, dividing it by the period and multiplying it by the voltage supply. This procedure is summarized in the following equation:

$$P_{avg} = \frac{V_{dd}}{T_e - T_s} \int_{T_s}^{T_e} I(t)dt \tag{5.2}$$

Where T_s is the start of the interval, T_e is the end of the interval, V_{dd} is the supply voltage and $I(t)$ is the current over time. Measuring average power over the total or individual gate period guarantees that each gate will realize exactly two transitions (one $0 \rightarrow 1$ and one $1 \rightarrow 0$) and that secondary elements such as idle currents are being considered. With both equations, we can observe the effects of voltage scaling on gate delay and power consumption. Gates in the oscillator were sized according to the methodology explained last section. Again, voltage was scaled from 1.0V to 3.3V using increments of 0.1V. In each iteration, gate delay, average power consumption and average energy (power multiplied by delay) per cycle was measured. This allowed us to produce graphs depicting variations on such parameters. Figure 5.7 shows the frequency, average power and average energy per period in the ring oscillator.

To have a better understanding of the curves, some of the obtained results for the inverter, the NAND3 and the NOR3 cells are presented in table 5.5. In all cases, the circuit working at the lower voltage spends less energy in a period, meaning that even if the frequency increases when the voltage goes up, the increment in power consumption are more significant. In the inverter case, the frequency at 3.3V is about approximately 165 times larger than the frequency at 1.0V, and the power consumption is about 2700 larger. It is evident from the curves that the most dramatic power and delay changes happen in the 1.0V-1.5V range. In the inverter case, increasing the voltage from 1.0V to 1.5V makes the frequency around 33 times higher, while the power consumption becomes about 90 times higher. The results demonstrate that if performance is only of secondary concern, working at the lowest possible voltage gives us

Figure 5.7 – Frequency, average power and average energy in function of supply voltage.



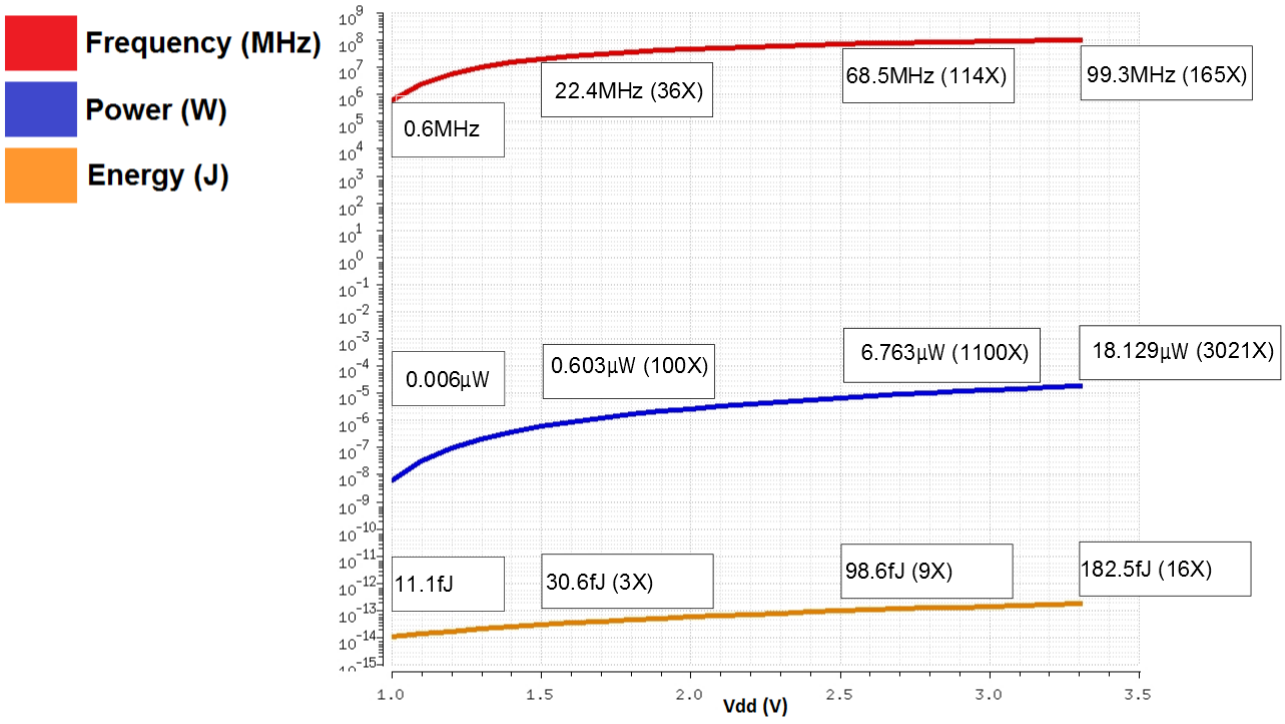
the most energy-efficient operation. Besides, working at the nominal supply voltage gives us the highest frequency, but at the cost of a very high power consumption. Thus, if we need an intermediate approach, working at around 1.5V would give a huge increase in performance in comparison to the 1.0V voltage while consuming much less power than the 3.3V supply voltage case. In Figure 5.8 presents the values in logarithmic scale, where the results can be confirmed — the power consumption grows in a larger rate than the frequency as the supply voltage is increased.

Table 5.5 – Frequency, power and energy values for specific voltage values.

Gate	*Parameter*	Supply Voltage			
		1.0V	1.5V	2.5V	3.3V
INV	Frequency	0.590MHz	22.450MHz	68.488MHz	99.324MHz
	Power	0.006µW	0.603µW	6.763µW	18.129µW
	Energy	11.114fJ	30.585fJ	98.751fJ	182.529fJ
NAND3	Frequency	0.112MHz	8.434MHz	37.848MHz	60.166MHz
	Power	0.001µW	0.197µW	2.952µW	8.682µW
	Energy	8.801fJ	23.437fJ	78.005fJ	144.31fJ
NOR3	Frequency	0.290MHz	7.273MHz	25.198MHz	37.274MHz
	Power	0.004µW	0.372µW	4.522µW	12.652µW
	Energy	16.185fJ	51.221fJ	179.482fJ	339.427fJ

Also in Figure 5.8, it is possible to observe regions with distinct performance-energy compromises. For instance, if the designer wishes to develop an application that needs to consume the least possible power but has no concerns on performance, a voltage of around 1.0V is rec-

Figure 5.8 – Frequency, power and energy values versus supply voltage in logarithmic scale.



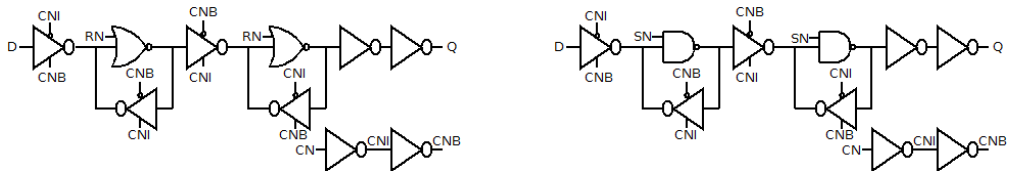
ommended. If the application needs to combine low-power and performance, the designer can consider a voltage of about 1.5V, since the frequency increase is approximately 20X but the energy per operation increase is only about 3X. If the application has to run on the highest possible frequency and has no established limit on power consumption, then it can be operated in the nominal voltage range. As the focus of this work is the near-threshold voltage operation, sub-threshold voltages were not considered in the analyses, but they constitute another important region, suited for ultra-low power applications such as hearing aids, wireless sensor networks and pacemakers, i.e., applications that need to function for a very long time with no need to switch batteries.

5.2.3 Flip-flop Architecture

Throughout the years, countless flip-flops architectures with different objectives were developed. In (ALIOTO; CONSOLI; PALUMBO, 2010), Alioto et. al discuss more than ten different architectures and its impact on the energy and performance of the circuit. Some of the architectures mentioned are very unorthodox, and there is a large probability that some of their logic is not recognized by the electrical characterizer. For this reason, we have chosen a conservative register implementation, specifically the same architecture used by the CTC LIB typical and low area libraries (Figure 5.9). The first circuit in the figure shows a flip-flop with falling clock edge trigger and a low reset input, meaning the setreset should be in 0 logic state to set the output at '0'. The second circuit is the flip-flop with low set trigger which is analogous

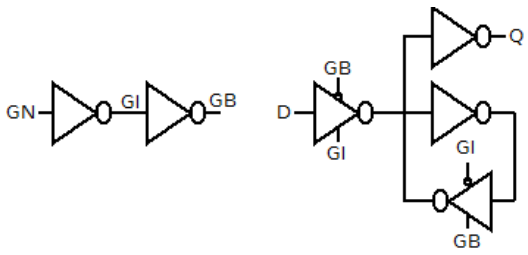
to the first one, the difference being on the presence of a 2-input NAND gate instead of a NOR gate to implement either the set or reset input. The clock signal is represented by the 'CLK' input, which passes through two inverters to be sped up. the 'D' and 'Q' pins are the registers' data input and output, respectively. Simulation results show that we can obtain symmetrical rise and fall times using a W_p of $1.5\mu m$ and a minimum sized W_n in the master and slave stage networks. The input and output buffers use the same PN ratio found in section 5.2.1 (2). To keep capacitance as low as possible, registers with higher drive strengths (i.e., X2 or X4) will only have its output inverters upsized.

Figure 5.9 – Two D-type flip-flop implementations. a) Flip-flop with low reset ('RN') input. b) low set ('SN') input.



The master-slave architecture is also useful to quickly implements latches. We can use one of the stages of the registers and employ slight modifications to obtain different latch architectures. In Figure 5.10, a simple D-type latch with no set or reset is implemented. Again, the 'D' and 'Q' signals are the data input and output, and the clock signal is the 'G' pin. Besides the presented cells, other types of flip-flops were planned, such as flip-flops with rising clock edge trigger, clock gating and scan chain (a technique used to provide observability and controlability of certain signals of sequential cells in a circuit).

Figure 5.10 – D-type latch implementation.



5.3 Layout Level Design

Before we start to implement the selected logic functions, it is interesting to study the most important layout characteristics of the XFab $0.6\mu m$ technology and the design decisions taken in the other CTC LIB project libraries (typical and low area).

The technology has three metal layers (MET1,MET2 and MET3) used to perform routing. Therefore, the MET1 layer will be used in both internal routing (transistor to transistor) and

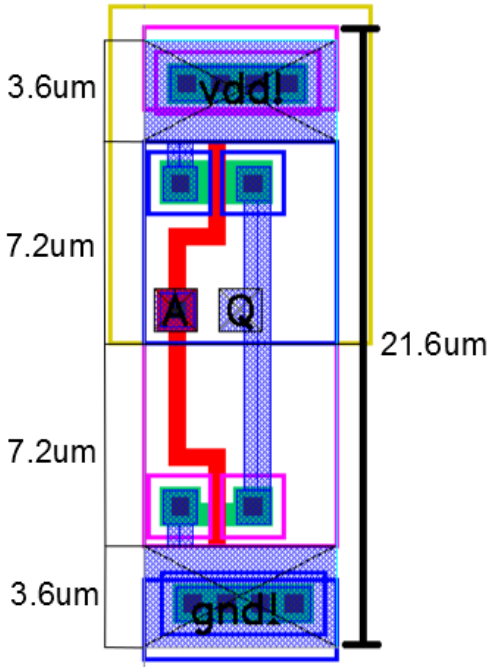
in global routing (cell to cell), while the MET2 and MET3 layers will only be used in global routing. In the default routing grid for this technology, the MET1 and MET3 layers runs horizontally and the MET2 layer runs vertically, thus the cell height is defined as a multiple of the MET1 pitch, and the cell width is defined as a multiple of the MET2 pitch. The pitch is usually defined as $W_{min} + d_{min}$, where W_{min} is the minimum width of the metal track and d_{min} is the minimum center-to-center distance between two tracks. Using this formula would give a value of 1.7 to both MET1 and MET2 pitch, but the foundry designers decided that the MET1 pitch would be 2.4 and the MET2 pitch would be 2.3, probably considering that MET1 and MET2 pins must have a larger width ($1.5\mu\mathrm{m}$) to be able to fit a contact inside, so we shall use these values. Active area or diffusion has a minimum width of $0.8\mu\mathrm{m}$, therefore when we discuss a minimum sized device in this technology we actually refer to a device with a $0.8\mu\mathrm{m}$ width active area. N-type implants, P-type implants and N-type wells are available to differentiate between NMOS and PMOS transistors.

Now, we analyze the design parameters of both the typical and the low area standard cell library of the CTC LIB project. The designers could choose between setting the width and varying the height or setting the height and varying the width, and the latter was chosen. Consequently, supply rails are placed at the top and at the bottom of the cell. The typical library has a fixed height of $26.4\mu\mathrm{m}$ (11 times the MET1 pitch) and supply rails of $3.8\mu\mathrm{m}$, meaning there is a space of $18.8\mu\mathrm{m} \cdot w$ (where w is the cell width) to place pins, transistors and internal routing wires. For the low are library, the height was fixed in $16.8\mu\mathrm{m}$ (7 times the MET1 pitch) and supply rails of $3.6\mu\mathrm{m}$, leaving a $9.6\mu\mathrm{m} \cdot w$ space to finish the gate. Since our goal is to design a low voltage library, there is no need to follow the most radical approach used by the low area library. We then decided for an intermediate approach, setting the height at $21.6\mu\mathrm{m}$ (9 times the MET1 pitch). Regarding pin placement, pins should be placed upon the points where a MET1 track intercepts a MET2 track to ease the routing automation process.

Cells in the typical library are designed in a conservative CMOS style. Bulk contact lines are placed in both supply rails, and remaining space can be used to route internal signals if there is no space left in the middle. In the low area library, extreme area saving techniques had to be used, such as using diffusion and polysilicon layers to realize internal routing, employing different logic implementations that use less area and devices and placing fewer or even no bulk contacts to be able to use supply rail area efficiently. Since the cell height of our low voltage library is an intermediate value, there is still plenty of space and the use of the techniques described below will probably not be required. Taking this information into consideration, the layout template (a simple CMOS inverter) in Figure 5.11 was developed.

The template exhibits several characteristics. Two substrate contact lines run horizontally on the bottom and top of the ground and supply rails, respectively. The red line represents the polysilicon layer, the blue, dotted lines are MET1 wires responsible for internal routing and the diffusion layer is the green area. The extra space in the ground and supply rails can be used to do routing with the polysilicon layer if needed. The layout is divided into two equal parts to

Figure 5.11 – Layout template for the standard cell library.



place the pull-down and pull-up networks. The upper area presents a P-type implant, meaning it contains the PMOS network. The NMOS network is placed in the lower area, where a N-type implant layer is present. The yellow rectangle around the PMOS network is a N-type well. Crossed black lines upon a metal layer represent input and output pins. The left pin is the input pin and goes directly to the gate of the transistor through the polysilicon layer. The bottom and top pins are the ground and V_{dd} pins. The right pin is the output pin.

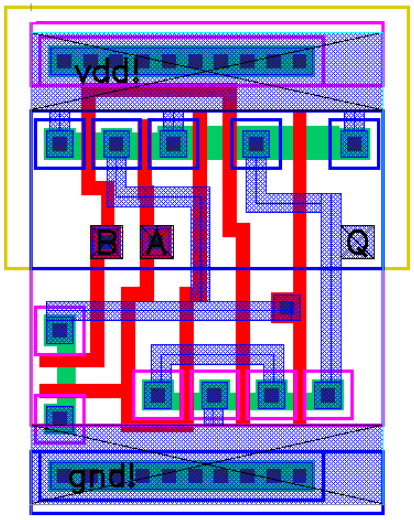
The rest of the cells should be built using this template as reference, with the height fixed and the width changed in increments of the MET pitch ($2.3\mu m$). An example of layout constructed using the template as reference can be seen in Figure 5.12. The layout depicted in the figure is the EN2 cell, which is a XNOR gate. It has a width of $16.1\mu m$ (7 times the MET2 pitch) and an area of $270.48\mu m^2$.

5.4 Characterization of the Library

The library was characterized with the Cadence ELC (Encounter Library Characterizer) tool, which uses a series of SPICE simulations to give power and delay estimation. To obtain realistic measures, input slopes, output loads, total simulation time and step size have to chosen according to the general characteristics of the library under test. Therefore, we decided to use the same input slopes and output loads as the low area XFab $0.6\mu m$ standard cell library, as its MOS devices are sized with similar values when compared to our library.

The library was characterized with 6 different processes, each one with typical $25°C$ temperature, no variation in the PMOS or NMOS devices threshold voltage and voltage ranging

Figure 5.12 – Layout of the EN2 cell.



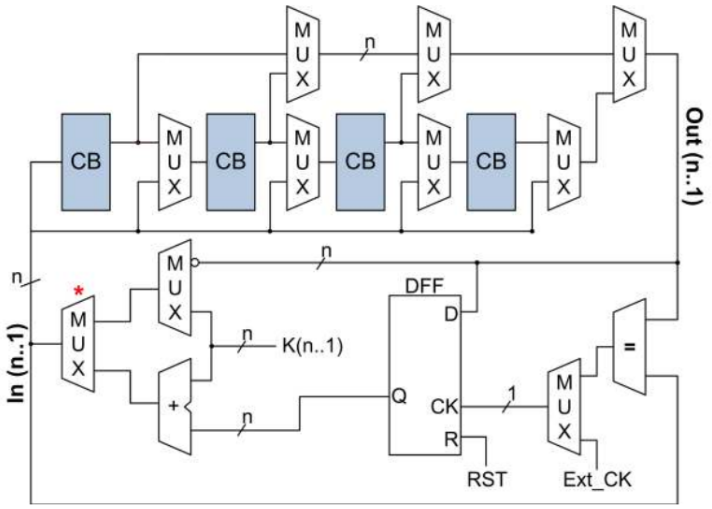
from 1.0V to 1.5V, with 0.1V increments. Unfortunately, it was not possible to characterize the library with a 1.0V voltage, since the sequential cells failed at some of the realized tests. As we are not allowed to observe exactly what kind of result the characterization tool expects in a specific case, and the output log file is not informative enough, the accurate cause of the errors could not be inferred. An alternative solution would be to use another electrical charater-izer, such as Cadence Liberate, which is slowly replacing the already obsolete ELC. The extra functionalities of the Cadence Liberate could help identify and fix any present issues in the sequential cells.

5.5 Obtained Results

The library was tested by synthesizing the architecture proposed in (RIBAS et al., 2010), known as LibTest. This architecture guarantees that the entirety of the standard cell library will be tested, as opposed to conventional benchmark circuits. It is based on an entity named Combinational Block, which is composed of two stages. The first stage contains a variable number of cells under test. Their inputs are connect to the inputs of the block and their outputs are connect to the second stage. The second stage is constructed considering the cells instantiated in the first stage. It is designed to receive the output of the first stage and reproduce the input of the Combinational Block. In other words, the block outputs should be made equal to the block inputs. Several of these blocks are cascaded until every cell under test is instantiated, and the output of each block is reutilized as stimuli to the next block. This ensures that cells can be fully tested by applying all possible logic combinations at the input of the block chain. The chain is controlled by the specified operation mode. In the synchronous mode, a counter and a register guarantees that the block chain will be stimulated by a different logic signal at every clock trigger. The asynchronous mode is essentially the same, but the register will be

triggered by a comparison between the chain inputs and outputs, instead of by an external clock signal. When the inputs and outputs are equal, the counter will calculate a new value, and the cycle continues. If any of the cells are defective, then the inputs will not propagate to the end of the chain and the circuit will stop working. An interesting feature is that it is not necessary to test the entire cell set. The circuit provides multiplexers between the blocks which allows the designer to isolate a single block in the chain, in case there is interest in testing a reduced number of cells. The LibTest architecture can be seen in Figure 5.13.

Figure 5.13 – Overview of the Libtest architecture.



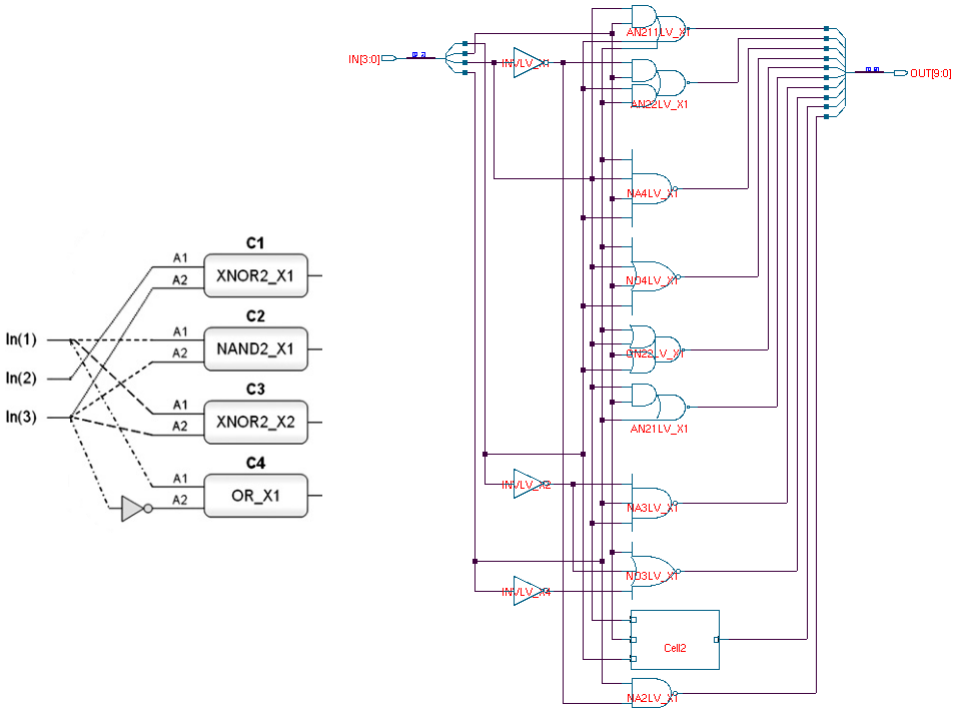
Source: (RIBAS et al., 2010).

The architecture was slightly modified due to issues in the asynchronous mode operation. Three comparators were added to assure that no glitches would compromise the asynchronous mode operation. Four different signals were compared: the input of the combinational block, the output of the combinational block, and two internal signals in the chain. The comparison result is only used as a clock for the register when all four signals are equal, which provides an additional level of security against glitches that could propagate.

The circuit was implemented using two distinct tools: one, to generate the control part of the circuit (which contains the adder, the register and the multiplexers) according to the cell with the maximum number of inputs in the library, and the the other, to generate the combinational blocks. The firs tool was designed by the members of the CTC LIB project and the second tool was developed by the same authors of (RIBAS et al., 2010). The circuit was synthesized using the Cadence RTL Compiler tool, which performs the logic synthesis of an initial RTL description. The resulting netlist had 1947 cells and 3 combinational blocks were generated.

A very large number of inverters were also instanced. Some of them were used in the clock tree, but since we have not implemented any buffer yet, most of the inverters were used in pairs to form buffers and speed up internal signals. In Figure 5.14 it is possible to observe the first stages of one of the combinational blocks, in which some of the cells of the library is present, for example, AN211 or NO3.

Figure 5.14 – First stage of a combinational block.



The logic synthesis was done using the five .lib files obtained in the previous section. The power consumption was estimated for the asynchronous operation considering the critical (longest) path of the circuit. Table 5.6 shows power consumption values for each voltage case. As expected, the consumption increases as the voltage is scaled up, but it is necessary to consider that the power estimation given by the tool in this point is not reliable.

Table 5.6 – Power estimation of the LibTest logic synthesis for different voltage values.

Voltage	*Leakage Power*	*Dynamic Power*
1.1V	1.444nW	2.208mW
1.2V	1.794nW	2.637mW
1.3V	2.174nW	3.118mW
1.4V	2.583nW	3.639mW
1.5V	3.022nW	4.203mW

6 CONCLUSION AND FUTURE WORK

Using the theoretical background present in the first chapters and the knowledge acquired reading the related work, it was possible to design the proposed low-voltage standard cell library and pass through almost every design flow step. The work also showed the differences between sub-threshold and near-threshold voltage operation, what kind of precautions the designer has to take to succesfully employ such techniques and our reasons to have chosen near-threshold voltage design. The schematics and layouts of the partial cell set were sucessfully designed and the library was characterized for a number of different supply voltage values.

Further work consists of finishing the standard cell library. Up to the date of publication of this work, 21 cells were already designed, and more than 100 cells (including cells with different drive strengths) are under design. Besides digital cells, physical cells also have to be designed for the library to be able to perform physical synthesis of a netlist. Circuit simulation is another important point that was not covered in this work, as tools require the verilog files that describe each function present in the library. To generate such files, we have to extend our electrical characterization and include additional parameters. Even though the ring oscillator simulation showed promising results, the library has yet to be compared to another libraries in a real circuit, since power estimation given by logic synthesis tools is not very accurate and it was not possible to realize physical synthesis. Such comparisons could be valuable to, for instance, evaluate our sizing methodology.

REFERENCES

ALIOTO, M.; CONSOLI, E.; PALUMBO, G. Flip-flop energy/performance versus clock slope and impact on the clock network design. **IEEE Transactions on Circuits and Systems I**, v. 57, p. 1273 – 1286, 2010.

CALHOUN, B. H.; CHANDRAKASAN, A. P. Characterizing and modeling minimum energy operation for subthreshold circuits. **Proceedings of the 2004 International Symposium on Low Power Electronics and Design**, p. 90 – 95, 2004.

CHANDRAKASAN, A. P.; BRODERSEN, R. W. Minimizing power consumption in digital cmos circuits. **Proceedings of the IEEE**, v. 83, p. 498 – 523, 1995.

CHANDRAKASAN, A. P.; SHENG, S.; BRODERSEN, R. W. Low-power cmos digital design. **IEEE Journal of Solid State Circuits**, v. 27, p. 473 – 484, 1992.

DRESLINSKI, R. G. et al. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. **Proceedings of the IEEE**, v. 98, p. 253 – 266, 2010.

KEANE, J. et al. Subthreshold logical effort: A systematic framework for optimal subthreshold device sizing. **Design Automation Conference, Proceedings of the 43rd IEEE/ACM**, p. 425 – 428, 2006.

KWONG, J. **A Sub-Threshold Cell Library and Methodology**. Dissertation (Master) — Massachusetts Institute of Technology, Cambridge, Massachussetts, 2006.

KWONG, J.; CHANDRAKASAN, A. P. Variation-driven device sizing for minimum energy sub-threshold circuits. **Proceedings of the 2006 International Symposium on Low Power Electronics and Design**, p. 8 – 13, 2006.

MARKOVIC, D. et al. Ultralow-power design in near-threshold region. **Proceedings of the IEEE**, v. 98, p. 237 – 252, 2010.

PEDRAM, M. Power minimization in ic design: Principles and applications. **ACM Transactions on Design Automation of Electronic Systems**, v. 1, p. 3 – 56, 1996.

RABAEY, J. M.; CHANDRAKASAN, A. P.; NIKOLIC, B. **Digital Integrated Circuits - A Design Perspective**. 2nd. ed. [S.l.]: Prentice Hall, 2002.

RIBAS, R. P. et al. Contributions to the evaluation of ensembles of combinational logic gates. **Microelectronics Journal**, v. 42, p. 371 – 381, 2010.

SOELEMAN, H.; ROY, K. Ultra-low power digital subthreshold logic circuits. **Proceedings of the 1999 International Symposium on Low Power Electronics and Design**, p. 94 – 96, 1999.

SOELEMAN, H.; ROY, K. Digital cmos logic operation in the sub-threshold region. **GLSVLSI '00 Proceedings of the 10th Great Lakes symposium on VLSI**, p. 107 – 112, 2000.

SOELEMAN, H.; ROY, K.; PAUL, B. C. Robust subthreshold logic for ultra-low power operation. **IEEE Transactions on Very Large Scale Integration**, v. 9, p. 90 – 99, 2001.

STANGHERLIN, K. H. **Energy and Speed Exploration in Digital CMOS Circuits in the Near-threshold Regime for Very-Wide Voltage-Frequency Scaling**. Dissertation (Master) — Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, Brazil, 2013.

SULISTYO, J. B. **On the Characterization of Library Cells**. Dissertation (Master) — Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2000.

SUTHERLAND, I.; SPROULL, R. Logical effort: Designing for speed on the back of an envelope. **University of Californa, Santa Cruz Conference in VLSI**, p. 1 – 16, 1991.

WANG, A.; CHANDRAKASAN, A. P. A 180-mv subthreshold fft processor using a minimum energy design methodology. **IEEE Journal of Solid State Circuits**, v. 40, p. 310 – 319, 2005.

WANG, A.; CHANDRAKASAN, A. P.; KOSONOCKY, S. V. Optimal supply and threshold scaling for subthreshold cmos circuits. **Proceedings of the IEEE Computer Society Annual Symposium on VLSI**, p. 7 – 11, 2002.

APPENDIX A STANDARD CELL LIBRARY PARTIAL CELL LIST

In the logic function field, the output is represented by 'Q', the inputs by 'A', 'B',..., 'E', '!' is the negation (NOT) operator, '.' is the conjunction (AND) operator and '+' is the disjunction (OR) operator.

Table A.1 – Partial Cell List.

Cell Name	*Drive Strength*	*Logic Function or Description*
INLV_X1	X1	Q = !A
INLV_X2	X2	Q = !A
INLV_X4	X4	Q = !A
INLV_X8	X8	Q = !A
NA2LV_X1	X1	Q = !(A . B)
NA3LV_X1	X1	Q = !(A . B . C)
NA4LV_X1	X1	Q = !(A . B . C . D)
NO2LV_X1	X1	Q = !(A + B)
NO3LV_X1	X1	Q = !(A + B + C)
NO4LV_X1	X1	Q = !(A + B + C + D)
AN21LV_X1	X1	Q = !((A . B) + C)
AN22LV_X1	X1	Q = !((A . B) + (C . D))
AN211LV_X1	X1	Q = !((A . B) + (C . D) + E)
ON21LV_X1	X1	Q = !((A + B) . C)
ON22LV_X1	X1	Q = !((A + B) . (C + D))
ON31LV_X1	X1	Q = !((A + B + C) . D)
ITLLV_X1	X1	Tri-state inverter with low enable trigger
ITHLV_X1	X1	Tri-state inverter with high enable trigger
BTLLV_X1	X1	Tri-state buffer with low enable trigger
BTHLV_X1	X1	Tri-state buffer with high enable trigger
DFFRQLV_X1	X1	D-type flip-flop with low reset trigger
DFFSQLV_X1	X1	D-type flip-flop with low set trigger
DLHQLV_X1	X1	D-type latch.

APPENDIX B TG1 ARTICLE

Construction of a Low-Voltage Standard Cell Library for Ultra-Low Power Applications

Luís Henrique Reinicke[1], **Renato Perez Ribas**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

Abstract. *This article presents discusses the construction of a standard cell library aimed at ultra-low power applications, using sub-threshold voltage scaling. It presents an overview on power consumption in CMOS circuits, techniques to achieve power reduction, the advantages and challenges of operating circuits under the sub-threshold voltage regime and the guidelines to design a standard cell library functional under such constraints.*

1. Introduction

The standard cell-based approach is one of the most used methodologies in ASIC design. A standard cell is a network of transistors and interconnections arranged to yield a boolean logic function of its inputs. Such functions can be primitive(like AND,OR,NOT) or complex(like XOR,multiplexers or registers). Standard cells are used to automatize the construction of digital circuits through technology mapping(expressing an independent logic description in terms of library gates). Cells are built in a standardized manner(for example, fixed height and variable width) to ease the automation task of place and route tools. A standard cell library is an ensemble of standard cells. Cell libraries are often projected to provide solutions for a wide range of digital designs, so we could have, besides the typical(designed to operate at normal conditions) cell set, cells designed to operate at lower voltages to reduce power consumption, to use less area or to produce less noise, all within the same cell library.

Initial efforts on VLSI research were focused on the construction of increasingly faster circuits, and power consumption was considered only a secondary factor. This trend began to change when battery-based applications which demanded high performance(like cell phones and laptops) became popular, beginning in late-1980s and early-1990s. Using standard, off-the-shelf components to build such devices would yield a high power consumption, and making it necessary to carry a large quantity of battery, which would make the use of those devices impracticable. The interest on low power circuits continued to increase as applications that required ultra-low power but had no or little concern on performance emerged, such as sensor networks, environmental monitoring and portable. Techniques for saving power, such as those mentioned on [Chandrakasan et al. 1992] [Chandrakasan and Brodersen 1995] [Pedram 1996], vary from layout to circuit and architectural level. Some of them are mentioned on chapter III.

Among those techniques, voltage scaling is clearly the most efficient to obtain immediate results due to its quadratic relationship with dynamic power. Reducing supply voltage comes at a cost, though. In [Chandrakasan et al. 1992, Chandrakasan and Brodersen 1995] it is shown that we pay a speed penalty when using a reduced supply voltage. These penalties become even harsher when the supply voltage

approached or even surpasses the threshold voltage. Several other issues arise when devices are operating under the sub-threshold or near-threshold regimes, as circuits become dramatically sensitive to process, temperature and voltage variations [Wang et al. 2002]. Issues introducted by using low voltages are discussed in more detail in chapters III and IV.

Our goal in writing this article is to, firstly, demonstrate the difficulties and advantages of constructing circuits that work well operating in very low voltages, and how to overcome or mitigate issues such as process,temperature and voltage variations, degraded noise margins and reduced logic swings. Finally, we present the guidelines to build a carefully designed standard cell library (using the XFAB 0.6μ technology process), constructed to operate under the sub-threshold voltage regime, aimed at ultra low power applications. The library development must take into consideration the previously discussed side-effects of using very low voltages, as well as paying special attention to transistor sizing and stacking, factors that become crucial at building robust, variation-aware circuits when working under the sub-threshold voltage regime.

The article is organized as follows: first, a brief description of power dissipation components in CMOS devices. Then, we present some of the strategies used to reduce power consumption. In chapter III, we discuss about sub-threshold and near-threshold voltage digital design. Next, the article proposal(objectives, expected challenges and difficulties, structure of the cell library, design flow) is presented. We conclude by talking about previous works, presenting the activity schedule for the remainder of the work and finally summarizing the expected results as well as discussing the chosen test platform.

2. Power Consumption in CMOS Devices

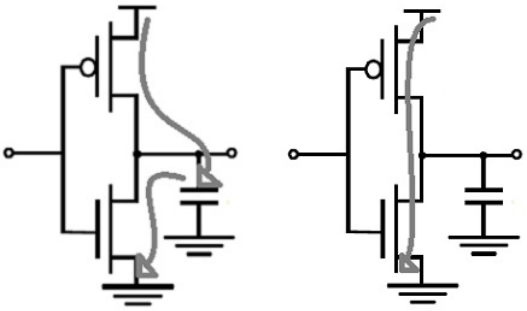
There are two sources of power dissipation in digital CMOS circuits: dynamic power and static power. Each source has a number of components that contribute more or less to the total average power consumption depending on various factors. These components will be briefly discussed in the following subsections. It is important to notice that since we are using an older technology(XFAB 0.6μ process), the dominant(by far) component will be dynamic power, when the circuit is operating at nominal supply voltage.

2.1. Dynamic Power

Dynamic power can be divided into two main components: switching activity and dissipation due to the short-circuit current. Switching power dissipation happens during logic changes, when the load capacitance is charged(from the supply voltage rail through the PMOS circuit to the load capacitance) or discharged(from the load capacitance through the NMOS circuit to ground). It is given by the following equation:

$$P_{switching} = p_{sw}.C_L.V_{dd}^2.f_{clk} \tag{1}$$

Where p_{sw} is the probability that a power-consuming transition occurs in a clock period, C_L is the load capacitance, V_{dd}^2 is the supply voltage and f_{clk} is the clock frequency. The product $P_{sw}.C_L$ is also known as C_{eff}(effective capacitance), the average capacitance being switched per clock cycle. The other component of dynamic power consumption happens in the brief moment where both CMOS and NMOS networks are active, creating



(a) (b)

Figure 1. Currents that contribute to dynamic power dissipation. a) Capacitance being charged through PMOS network or discharged through NMOS network. b) Short-circuit current flowing through the direct path between supply and ground. Images modified from [Rabaey et al. 2002]

a direct path between supply and ground. The current that flows through this path is called I_{sc}(short-circuit current). It is is given by:

$$P_{sc} = I_{sc}.V_{dd} \tag{2}$$

In Fig.1 both short-circuit and capacitance charging/discharging currents are shown.

2.2. Static Power

Static power consumption occurs due to leakage current when the circuit input is not switching. Leakage current has a significant number of components, mostly determined by fabrication process. The main components are sub-threshold leakage current(occurs between drain and source when the device is operating in the sub-threshold region), gate tunneling current(occurs from gate to substrate, caused by aggressive downscaling of the oxide thickness) and reverse bias junction current(from drain/source to substrate, caused by drain/source-to-well junctions being reverse biased), though there are other leakage mechanisms such as channel punchthrough, hot-carrier injection and gate induced drain leakage. Static power is given by:

$$P_{static} = I_{leakage}.V_{dd} \tag{3}$$

These different current components can be seen in Fig. 2.

3. Reducing Power Consumption

The previous section analyzed the main components of power dissipation in CMOS devices. According to equations 1, 2 and 3, we should be capable to reduce power consumption by modifying certain parameters, such as circuit activity, capacitance, voltage and short-circuit current. The techniques listed below attempt to achieve power reduction by changing one or more of these parameters:

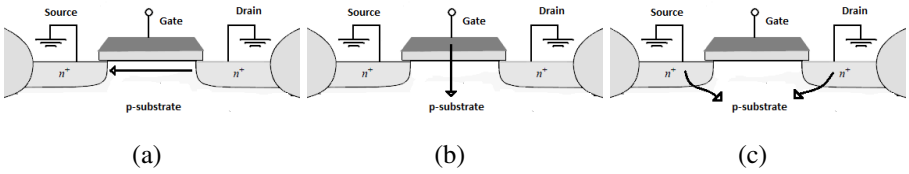


Figure 2. Leakage current components. a) Sub-threshold leakage current. b) Gate tunneling current c) Reverse bias junction current. Image modified from [Rabaey et al. 2002]

- *Logic style.* Using dynamic over static logic could reduce parasitic capacitance (less transistors) and the number of spurious transitions (also know as glitches), which are transitions that happen before the node settles down, mainly caused due to imbalanced paths in the network. However, circuits built using dynamic logic would have increased switching activity due to pre-charging and would consume extra clock power. Dynamic logic is also free of short-circuit currents. Pass-transistor logic has the advantage of using much less transistors in some implementations like XOR, adder and multiplexer gates, consequently reducing parasitic capacitance.
- *Implementation of Logic Functions.* There are various ways to implement a given logic equation. For example, to implement a N-bit adder, we could simply use a ripple-carry adder, which is simple to build but could have long delays due to slow carry propagation (last 1-bit adder has to wait for the carry to pass through every intermediate 1-bit adder) and present a significant number of glitches (due to delays on carry propagation) which could affect the functionality of the circuit and cause several power consuming transitions. On the other hand, using a carry-lookahead architecture would be faster, almost glitch-free but much more complex.
- *Capacitance.* It is possible to reduce parasitic capacitance using simple strategies, including using fewer and shorter wires, smaller devices and minimizing logic.
- *Operand Isolation.* In large combinational circuits, it is possible that unnecessary logic operations whose results are not going to be used happen. We shall call these transitions redundant operations. The operand isolation technique seeks to block the signal from propagating into parts of the circuit that would otherwise perform a redundant operation, preventing unnecessary power consumption. Operand isolation is usually achieved using multiplexers or latches on certain cell inputs.
- *Clock Gating.* In circuits where there is little clock activity, it is interesting to turn registers off whenever they are not loading new data values. Since clock toggles in registers happen at every clock cycle, a significant part of dynamic power consumed comes from clock tree buffers. Clock gating adds extra circuitry to gate a group of registers controlled by the same enable signal. Clock gating is not, however, an interesting technique to employ in high clock activity circuits, since registers would seldom be gated.
- *Voltage Scaling.* Reducing the supply voltage is the most effective way of minimizing power consumption, since it affects both dynamic and static power dissipation and has a quadratic relationship to dynamic power, but this reduction comes with a speed penalty. Also, a number of other issues arise when voltage is

scaled near or under the threshold voltage, as transistors become more sensitive to process, environmental and aging variability.

Since voltage scaling is the main focus of the article, a more detailed analysis on the advantages and issues of sub-threshold and near-threshold operation will be presented in the next chapters.

4. Sub-threshold and Near-threshold Voltage Digital Design

To understand the behavior of the MOS transistor, consider the I_d(drain current) *versus* V_{gs}(gate-source voltage) curve of Fig. 3 (on logarithmic scale). We can define three operating regions for the device: weak-inversion, moderate-inversion and strong-inversion region. Strong-inversion is the typical operating region, where the supply voltage exceeds the threshold voltage V_th and the drain currents increase in an approximately linear fashion. In the weak-inversion region, the drain current decays exponentially. The quadratic of moderate-inversion region is a transition region between the two. Thus, circuits operating under the threshold voltage(or sub-threshold circuits) will present weak-inversion drain currents. We assume that near-threshold circuits will operate slightly above the thresold voltage. Therefore, they normally present moderate-inversion drain currents.

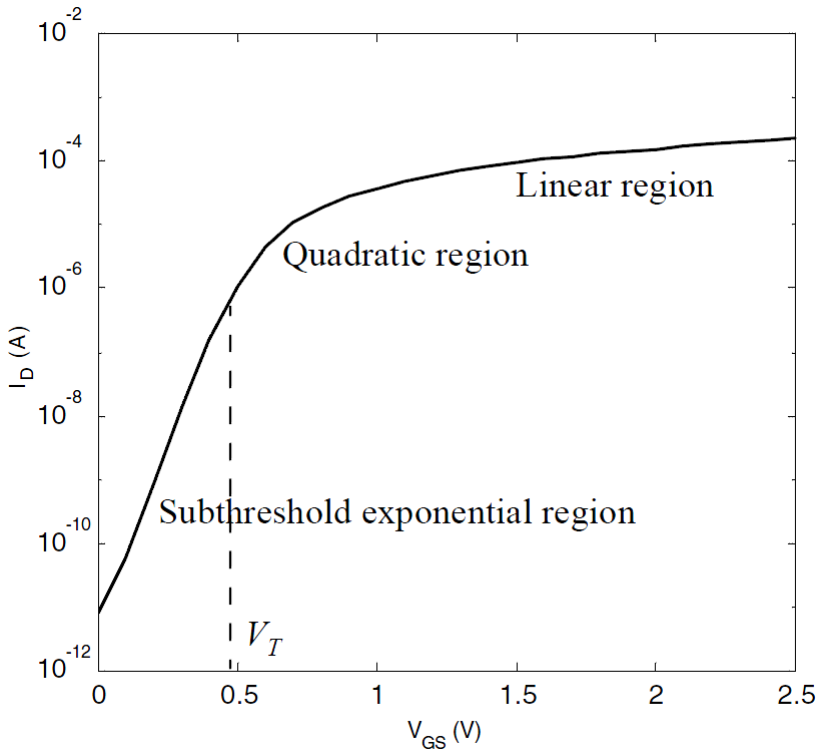


Figure 3. I_d ***versus*** V_{gs} **curve(in logarithmic scale) of the MOS transistor. Retired from [Rabaey et al. 2002].**

4.1. Process, Environmental and Aging Variability

Designing CMOS circuits to function in deeply scaled voltages is no simple task, as several factors have to be considered. For instance, devices operating under these conditions are more susceptible to process, environmental and aging variability. In

[Stangherlin 2001] the author demonstrates that for a circuit designed using the IBM 65nm process technology and operating at 250mV (considering a strong-inversion voltage of 1.2V), the worst-case scenario(3-σ variation, 10% voltage variation and -40° C temperature) would induce up to 99 times increased cycle time. A more detailed analysis of this article is presented in chapter VI.

4.2. Frequency Limits

According to Fig. 3, drain currents decay exponentially when device voltage is scaled below the subthreshold voltage, so we have to keep in mind that delays rapidly increase when the supply voltage approached the threshold voltage. Hence, any deep voltage reduction must be accompanied by a frequency reduction, or the circuit will not function properly. This will limit sub-threshold and near-threshold voltage usage to applications in the low-to-moderate performance range. For example, in [Wang et al. 2002], the authors demonstrate that circuits designed using a $0.18\mu m$ technlogy process operating in the sub-threshold voltage regime could operate in the $100KHz - 10MHz$ range, whilst a typical commercial microprocessor using this technology usually operates in the $800MHz - 1GHz$ range.

4.3. Device Sizing and Stacking

Transistor sizing and stacking also have to be taken into account. Sutherland in [Sutherland and Sproull 1991] suggested the logical effort method to calculate the optimal PMOS/NMOS width ratio (expressed as W_p/W_n) in order to obtain roughly equal pull-up and pull-down currents and minimize delay in CMOS circuits. The logical effort of a gate is defined as the number of times it is worse at delivering output current than would be an inverter with identical input capacitance. It describes a gate drive capability relative to that of a reference inverter. The author also proposes that the effective width of a n-stack of transistors is $1/n$, so stacked devices should be upscaled in order to conduct the same amount of current as a single transistor.

In [Keane et al. 2006], Keane demonstrates that these suppositions don't hold when devices operate in deeply scaled voltages due to the different nature of currents in sub-threshold region compared to those in the strong-inversion region. He then presents his findings on how to calculate a sub-threshold logical effort as well as the sizing factor for stacked devices. Results found show that even though the W_p/W_n ratio can be lower than the usual value of 2:1, the stacked transistors should be upscaled more agressively. Since the calculated scaling factor depends on both supply voltage and technology, it is possible that the stacked devices width become wide enough to not fit in the cell height, so we might have to limit the number of maximum stacked transistors in a cell.

4.4. Logic Style and Implementation

Another important factor is logic style. In [Wang and Chandrakasan 2005], the author shows that, depending on inputs, the I_{on}/I_{off} ratio can be degraded for a tiny XOR gate (using only one transmission gate) since the tiny XOR gate has unbalanced pull-up/pull-down networks and thus, the output has a reduced voltage swing. The I_{on}/I_{off} ratio can be used to measure the functionality of a logic gate. The I_{on} current is the drive current of the device and the I_{off} is the idle current. Since the tiny XOR gate has unbalanced pull-up/pull-down networks, there is a case where the idle currents compete with the drive

currents, and the voltage swing is reduced. To mitigate this issue, a double-transmission gate XOR (which has balanced pull-up/pull-down networks) is used.

In [Soeleman et al. 2001], Soeleman proposes two alternatives to conventional CMOS circuits in the sub-threshold region, namely Variable Threshold voltage sub-threshold logic(VT-Sub-CMOS) and sub-threshold Dynamic Threshold voltage logic(Sub-DTMOS). The VT-Sub-CMOS logic is similar to the conventional CMOS logic with the addition of a stabilization circuit built to detect any abrupt variation of the current and minimize the effects of process and temperature variations. Sub-DTMOS logic attempts to mitigate variations and achieve stability with direct body biasing by tieing the gates to their substrates. Results demonstrated that both families show superior robustness and tolerance to process and environmental variability issues, as well as gains in performance, with only a small increase in power consumption.

5. Sub-threshold Standard Cell Library

In the previous chapter, we discussed the main issues and challenges of sub-threshold design. We now analyze the design flow of a typical standard cell library and extend it to the sub-threshold design space, considering the constraints defined beforehand. The cell library is intended for usage along with commercial CAD tools to accomplish automated digital design and will be constructed using XFAB $0.6\mu m$ process technology. It is aimed at applications that require ultra-low power consumption, such as hearing aids, pace makers, distributed sensor networks, environment monitoring devices and wrist watches. We assume that digital circuits designed using this cell library will operate at voltages lower than the threshold voltage of the technology(approximately $0.9V$, according to the technology datasheet). We start the chapter by analyzing the development phases of a standard cell library.

5.1. Standard Cell Library Design Flow

The development of a standard cell library can be summarized in the following steps:

- *Choice of logic functions*. Definition of which logic functions (and how they are implemented) will be present in the library, as well as each function provided drive strengths.
- *Library-level project decisions*. Definition of constraints and global parameters that will be applied through every cell in the library, including cell width, height, metal layers used in intra and inter-cell connections, pin placement, supply rail direction and dimensions, PMOS/NMOS ratio(W_p/W_n) and stacked transistor upsizing.
- *Schematic design*. Construction and circuit-level simulation of electric schematic of a cell.
- *Layout design*. Construction, validation(layout versus schematic, design rules checking, extraction) and circuit-level simulation of cell layout using appropriate CAD tools (like Cadence Virtuoso or Synopsys Galaxy Custom Designer).
- *Generation of library files*. After each cell is validated, two different types of files must be generated: one containing information about the geometry of the cells (for example, .LEF files, to be used by Place and Route tools) and the other containing information about various cell parameters (the industry standard is the

Synopsys .lib extension) such as delay, capacitance, power consumption, area and logic function.

Logic function selection is a specially important step. A library with a large variety of cells could reduce workload from logic synthesis tools as well as minimize area and power due to the resultant circuit having less interconnect wires. For instance, suppose we synthesize a circuit from a functional description(written in, for example, Verilog) of a full adder using a cell library that does not contain a full adder cell. The synthesis tool would have to assemble a full adder from less complex cells instead of just instantiating a full adder cell. The resulting circuit would have more cells, and thus, more interconnect wires, increasing area, capacitance and consequently power consumption.

Library-level project decisions also critical because they affect the circuit as a whole. For example, if we define an arbitrary cell height to construct the cell layouts but later conclude that this height would not be enough to fit every logic function and decide to increase it by ten percent, the entire circuit would be upscaled by ten percent, so these decisions must be really well thought. In schematic design, we simulate the cell and verify if performance and functionality meets the specification. In this step, it is possible to check if the PMOS/NMOS ratio and stacked transistor upsizing factor were chosen correctly. Layout design is a straightforward process if we planned each cell beforehand. After cells are validated and extracted, we use commercial tools (like Cadence ELC and Abstract Generator) to port the library to the timing and physical layout files.

In the next subsection, we discuss the project decisions for the proposed cell library.

5.2. Sub-threshold Cell Library Logic Function Choices and Project Decisions

The selection of logic functions in a sub-threshold cell library should be more strict than the one realized in a conventional cell library. Cells that implement very complex logic functions or large transistor stacking may require agressive device upscaling to work properly and this could demand a severe increase in cell parameters (for example, cell height and width), resulting in impracticable area occupation. Initially, we employ a conservative approach in function choice. The chosen cells as well as the functions they implement can be seen on Table 1. It is probable that the final version of the library implements a greater number of functions and also distinct drive strengths for each cell.

To facilitate the work of Place and Route tools, the remaining project decisions should be taken considering certain properties. Since our chosen technology(XFAB 0.6μm) has only two metal layers, we should use only the METAL1 layer in intracell connections, and leave the METAL2 layer to the routing tool. Assuming we use horizontal supply rails, all cells must have the same height and be able to accomodate both supply rails(power/ground) and a reasonable number of routing metal tracks. Cell height must be a multiple of horizontal grid spacing and cell width must be a multiple of vertical grid spacing. These conditions guarantee the continuity of the routing grid between two adjacent vertical or horizontal cells. A layout template containing these properties could be defined to simplify the work of the designer. The remaining parameters(W_p/W_n ratio and stack sizing factor) shall be determined in the next month.

Table 1. Suggested cell set.

Cell Name	Function	Description
INV	Q = !A	Simple CMOS inverter.
BUF	Q = A	CMOS buffer. Two inverters in series.
NAND2	Q = !(A.B)	CMOS negated AND gate with two inputs.
AND2	Q = A.B	CMOS AND gate with two inputs.
NOR2	Q = !(A+B)	CMOS negated OR gate with two inputs.
OR2	Q = A+B	CMOS OR gate with two inputs.
XNOR2	Q = !(!(A+B)+(A.B))	Negated XOR gate with two inputs.
XOR2	Q = !(!(A.B).(A+B))	XOR gate with two inputs.
ADDF	CoutN = ((A+B).Cin + A.B) Cout = !CoutN SN = ((A+B+Cin).CoutN + A.B.Cin) S = !SN	Full Adder gate. A, B and Cin are its inputs, S and Cout its outputs.
ADDH	Cout = A.B S = A B	Half Adder gate. A and B are its inputs, S and Cout its outputs.
DFFR	When reset = 1, Q = 0. When positive clock edge, Q = D, otherwise Q = previously stored D	D Flip-Flop register with reset and positive clock edge trigger.
DFFS	When set = 1, Q = 1. When positive clock edge, Q = A, otherwise Q = previously stored A	D Flip-Flop register with set and positive clock edge trigger.
INVTR	When enable = 1, Q = !A, otherwise Q = Z(high impedance)	Tri-state Inverter.
AOI2	Q = !(A.B)+(C+D))	AND-OR-Invert gate with two inputs.
OAI2	Q = !(A+B).(C+D))	OR-AND-Invert gate with two inputs.

6. Activity Schedule

Here, we present an estimated activity schedule for our project, shown in Table 2.

In order to acquire the theoretical basis to proceed with this work, the first two months were dedicated to reading publications about the specified subjects. Many articles, most written by distinguished authors in the low power area, were read and analyzed. The project decisions phase will probably start at the end of june and end at early august. The remainder of august and september will be used to go through the schematic-level and layout-level design phases. October will be used to conclude the work, and the final article should be ready by mid-November. Details on each phase were discussed in the previous chapter.

7. Previous Works

In [Sulistyo 2000], the author presents the basic design flow to construct a standard cell library. He starts by discussing certain properties that a cell library must have to work properly, such as functional completeness(the library must have certain functions, namely AND/NAND, OR/NOR, inverter, any kind of tristate cell, D latch and flip-flop with asynchronous set and reset, since most tools are unable to build complex sequential logic from AND/OR gates) and the regularity of shapes and sizes of a cell(to facilitate auto-

Table 2. Activity schedule for the project.

Phase		Mar	April	May	June	July	Aug	Sept	Oct	Nov
Reading	Low-power design space	■								
	Sub-threshold and near-threshold voltage design space		■							
	Characterization of standard cell libraries, sub-threshold and near-threshold libraries			■						
	Initial article			■	■					
Project Decisions	Study and analysis of the XFAB 0.6µ technology				■	■				
	Project decisions					■				
Schematic-level	Design of cell netlists						■			
	Circuit-level simulation						■			
	Analysis of obtained results						■	■		
Layout level	Design of cell layouts							■		
	Layout verification phase(DRC,LVS,extraction)							■		
	Circuit-level simulation of extracted layouts							■		
	Analysis of obtained results							■	■	
Conclusion	Port cells to timing abstract .lib and geometry .LEF libraries								■	
	Library simulation using the LibTest architecture								■	
	Final article									■

matic routing). Then, he proceeds to talk about the development flow of a standard cell library, which consists of three steps: netlist/schematic-level design and simulation, layout design and porting to synthesis and Place and Route libraries. In the next chapter, techniques to build regular cell layouts are presented. For example, metal tracks of the same layer must have the same width in all cells of the library, supply rails should run in the same directions, pins should be placed over the grid, among other strategies. In chapter IV, the author discusses characterization of delay and capacitance of logic cells. For simplicity, he utilizes the linear delay model(which consists of simple equations, is a good approximation of the real delay model and is used by most commercial synthesis tools) and defines instrinsic delay, transition delay and slope delay in combinational cells. For sequential cells, he argues that most parameters (for example, hold time, recovery/removal times,clock width) aren't as important as setup time and don't need to be characterized, and proceeds to define setup time more precisely. Finally, he presents a list of methods to calculate intrinsic, transition and slope delays, setup time in sequential cells, input/output capacitance in both combinational and sequential cells and techniques to calculate the previous parameters in tristate cells. Every definition is accompanied by a SPICE example. In chapter V, power characterization is discussed. The author introduces the basics of power dissipation(static and dynamic power) in CMOS devices and the Synopsys model of power dissipation to estimate static and dynamic power consumption. He finishes the chapter by talking about his own method to estimate power consumption in combinational, sequential and tristate cells(again, SPICE examples are given in each

definition). Finally, in the last chapter, he provides an example RTL description(using VHDL) of a twin 4-bit counter that has been synthesized using a standard cell library designed according to the general guidelines proposed in the article.

In [Stangherlin 2001], the author attempts to show the difficulties in digital IC design under the near-threshold voltage regime, presents a number of advantages of near-threshold over sub-threshold voltage operation and proposes a standard cell library based on the VFS(voltage-frequency scaling) technique constructed to function under the near-threshold voltage regime that will also work well under the nominal voltage operation, presenting only minor loss of performance. In the first chapter, an analysis of process(lithography and dopant), environmental(temperature and voltage) and aging variability and its effects on the performance of circuits operating on near-threshold and sub-threshold voltage regimes are discussed. A ring oscillator is used as a study case to evaluate these effects. The results show that CMOS circuits operating in lower voltage are much more sensible to variations(and extremely sensible to temperature variations), and could operate, considering a worst-case scenario(high temperature variation, standard 10% voltage, and 3-σ process variation), up to 99 times slower. Chapter III brings an overview of power consumption on CMOS devices and some of the strategies(clock and power gating, multi-VT devices, operand isolation, voltage-frequency scaling) used to design ultra-low power CMOS devices. In the next chapter, the author defines near-threshold voltage operation and VFS(voltage-frequency scaling), and discusses the techniques used to build the proposed cell library. To achieve balanced rise/fall times and correct voltage swings and noise margins, the main concerns are transistor sizing and stacking limit. Finally, the cell library is used to synthesize a notch filter, a 8051-compatible core and a few benchmark sequential and combinational circuits, and the results are commented. The author also presents his own conclusion of the article.

8. Expected Results

At the end of the estimated schedule (mid-November), the objective is to have a full functional standard cell library specifically designed to synthesize circuits operating under the threshold voltage. The library shall contain abstract symbols, layouts and netlists for every proposed cell, .LEF file containing information about geometry and physical layout of cells and timing abstract files (in the Liberty .lib format) for synthesis and simulation.

The library will be simulated with the architecture proposed in [Ribas et al. 2010], known as LibTest. This architecture guarantees that the entirety of the standard cell library will be tested, as opposed to conventional benchmark circuits. It is based on an entity named Combinational Block, which is composed of two stages. The first stage contains gdfgdfgdfgdfg cells under test. Their inputs are connect to the inputs of the block and their outputs are connect to the second stage. The second stage is constructed considering the cells instantiated in the first stage. It is designed to receive the output of the first stage and reproduce the input of the Combinational Block. In other words, the block outputs should be made equal to the block inputs. Several of these blocks are cascaded until every cell under test is instantiated, and the output of each block is reutilized as stimuli to the next block. This ensures that cells can be fully tested by applying all possible logic combinations at the input of the block chain. The chain is controlled by the specified operation mode. In the synchronous mode, a counter and a register guarantees that the block chain will be stimulated by a different logic signal at every clock trigger.

The asynchronous mode is essentially the same, but the register will be triggered by a comparison between the chain inputs and outputs, instead of by an external clock signal. When the inputs and outputs are equal, the counter will calculate a new value, and the cycle continues. If any of the cells are defective, then the inputs will not propagate to the end of the chain and the circuit will stop working. An interesting feature is that it is not necessary to test the entire cell set. The circuit provides multiplexers between the blocks that allows the designer to isolate a single block in the chain, in case there is interesting in testing a reduced number of cells. The LibTest architecture can be seen in fig. 4.

9. Conclusion

This work presented the challenges and advantages of operating MOS devices in deeply scaled supply voltages. Such devices, specially when operating under the threshold voltage, can achieve minimum energy operation and are useful in ultra-low power applications that operate in low-to-medium performance ranges.

An overview on general digital circuit power consumption, as well as general guidelines to construct a standard cell library are also presented. Analysis of previous works demonstrate that, taken appropriate precautions, it is entirely plausible to construct a standard cell library aimed at sub-threshold circuits. Studies on designing standard cell libraries are then extended to subthreshold digital circuits, and the project proposal to construct a subthreshold cell library is presented.

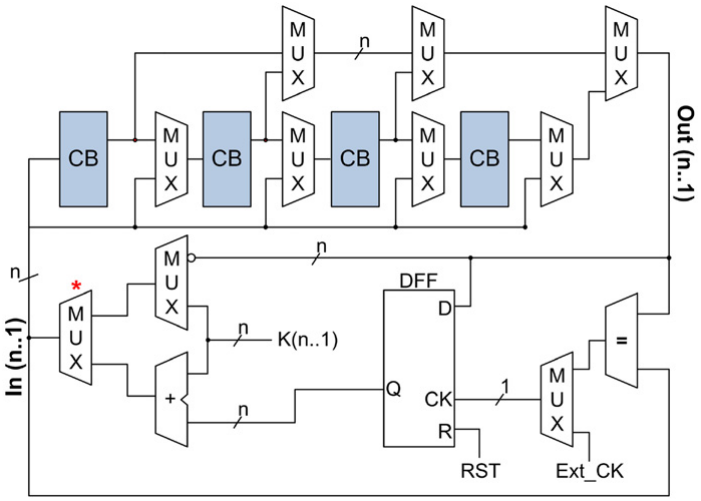


Figure 4. Overview of the Libtest architecture. Image retired from [Ribas et al. 2010]

References

Calhoun, B. H. and Chandrakasan, A. P. (2004). Characterizing and modeling minimum energy operation for subthreshold circuits. *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, pages 90 – 95.

Chandrakasan, A. P. and Brodersen, R. W. (1995). Minimizing power consumption in digital cmos circuits. *Proceedings of the IEEE*, 83:498 – 523.

Chandrakasan, A. P., Sheng, S., and Brodersen, R. W. (1992). Low-power cmos digital design. *IEEE Journal of Solid State Circuits*, 27:473 – 484.

Keane, J., Hanyong, E., Tae-Hyoung, K., , and Kim, C. (2006). Subthreshold logical effort: A systematic framework for optimal subthreshold device sizing. *Design Automation Conference, Proceedings of the 43rd IEEE/ACM*, pages 425 – 428.

Kwong, J. and Chandrakasan, A. P. (2006). Variation-driven device sizing for minimum energy sub-threshold circuits. *Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, pages 8 – 13.

Pedram, M. (1996). Power minimization in ic design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems*, 1:3 – 56.

Rabaey, J. M., Chandrakasan, A. P., and Nikolic, B. (2002). *Digital Integrated Circuits - A Design Perspective*. Prentice Hall, 2nd edition.

Ribas, R. P., Bavaresco, S., Schuch, N., Callegaro, V., Lubaszewski, M., and Reis, A. I. (2010). Contributions to the evaluation of ensembles of combinational logic gates. *Microelectronics Journal*, 42:371 – 381.

Sedra, A. S. and Smith, K. C. (2009). *Microelectronic Circuits*. Oxford University Press, 6th edition.

Soeleman, H. and Roy, K. (1999). Ultra-low power digital subthreshold logic circuits. *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, pages 94 – 96.

Soeleman, H., Roy, K., and Paul, B. C. (2001). Robust subthreshold logic for ultra-low power operation. *IEEE Transactions on Very Large Scale Integration*, 9:90 – 99.

Stangherlin, K. H. (2001). Energy and speed exploration in digital cmos circuits in the near-threshold regime for very-wide voltage-frequency scaling. Master's thesis, Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, Brazil.

Sulistyo, J. B. (2000). On the characterization of library cells. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Sutherland, I. and Sproull, R. (1991). Logical effort: Designing for speed on the back of an envelope. *University of Californa, Santa Cruz Conference in VLSI*, pages 1 – 16.

Wang, A. and Chandrakasan, A. P. (2005). A 180-mv subthreshold fft processor using a minimum energy design methodology. *IEEE Journal of Solid State Circuits*, 40:310 – 319.

Wang, A., Chandrakasan, A. P., and Kosonocky, S. V. (2002). Optimal supply and threshold scaling for subthreshold cmos circuits. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pages 7 – 11.