NTNU – Trondheim
Norwegian University of
Science and Technology

# Subthreshold Real-Time Counter.

Jonathan Edvard Bjerkedok

Master of Science in Electronics
Submission date: June 2013
Supervisor: Snorre Aunet, IET
Co-supervisor: Øivind Ekelund, Energy Micro AS

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

# Abstract

Design and implementation in layout are performed for a Real-Time Counter (RTC) in subthreshold operation. The design uses 65nm CMOS technology from STMicroelectronics. The designed RTC approximate the functionality of the RTC in Energy Micros EFM32G series microcontroller [3].

The RTC is used to minimize the microcontrollers power consumption. This is done by putting the CPU in deep sleep, while the RTC keeps track of time and can awaken the CPU with interrupts. The RTC contains a 24-bit counter with two 24-bits compare registers witch contribute to generate interrupts. The RTC is the main component that consumes power while the CPU sleeps. Therefore a low power implementation of the RTC is expected to extend battery life. The RTC is designed for a temperature range of -40°C to 80°C. Simulations performed on layout shows that the RTC has a power consumption of $6.2nW$ on 500mV. This reduces the assumed power consumption down to $1.5 - 2.5\%$.

This thesis also proposes a new design methodology with uniform building blocks. The new methodology improves robustness regarding Process, Voltage and Temperature variation, and should simplify cell libraries and layout.

# Preface

This thesis is the final part of a *Master of Science* degree in *Circuit and System design* at the Department of Electronics and Telecommunication, at the Norwegian University of Science and Technology in Trondheim. The thesis was started in January and submitted in June 2013.

Working on this thesis has been both challenging and interesting. This work has provided me with valuable knowledge about among other things: Ultra low power design, design of integrated circuits, CAD-tools and the art of performing layout.

First and foremost, I want to express my gratitude to my supervisor, Professor Snorre Aunet for accepting me as a student and for all the valuable guidance and knowledge during this project. Our discussions has provided inspiration and been a source for new ideas. I also want to thank my co-supervisor Trond Ytterdal for guidance and technical support, and Øivind Ekelund at Energy Micro for showing interest and to help forming this project.

Finally, I want to thank my family and especially my wife Tone Lise for endless patience and support during my study and this final project.

Trondheim, June 2013

Jonathan Edvard Bjerkedok

# Contents

# List of Figures

# Chapter 1

# Introduction

During the last years there has been a increased focus on low power design as more and more electronic applications are battery powered. Systems are emerging where power consumption is more important than performance. Possible application are sensor nodes that are energy autonomous. It could be wireless sensor networks, ambient intelligence, wearable computing, biomedical and implantable devices or sub-sea systems. Battery charging or replacement could also be costly or impossible. Battery lifetime has become the primary design metric and battery lifetime of several years or decades are highly desirable [1].

Ultra low power was first explored in the 1970s by Dr. Eric Vittoz for applications such as wristwatch and calculator circuits [15]. Lowering the supply voltage has become one of the most effective techniques for reducing the power consumption of digital circuits. Low power design usually translates to low voltage design. Supply voltages below the transistor threshold voltage, called subthreshold operation, is ideal for applications where speed is of little or no importance. Subthreshold design can reduce energy per operation by an order of magnitude compared to conventional strong inversion operation.

## 1.1 Motivation

More and more providers of microcontrollers are targeting battery powered devices, and they desire to operate for decades on a battery. This is done by reducing active power consumption, reduced processing time by having the CPU sleep 99% of the time. While the CPU sleeps, other circuits monitor a sensor or a Real-Time Clock (RTC) are used to awaken the CPU when needed. The RTC is the main component that consumes power when the CPU is in deep sleep. Battery lifetime can be extended with several years by reducing the RTC power consumption.

## 1.2 Previous Work

The only know commercial product operating in subthreshold, is a timer circuit from Ambiq Micro, http://ambiqmicro.com. The have Real-Time Clocks operating with a power consumption of 15-55nA at 3V.

## 1.3 Problem Description

This thesis will explore the possibility of creating a Real-Time Clock using subtreshold design methodology. The Real-Time Clock consists of a 24-bit counter with reset and overflow detection. In addition, there are two 24-bit compare registers that triggers interrupts. The counter will run at 32768Hz. Care should be taken to approximate the functionality of the RTC in the EFM32G series microcontrollers.

Different aspects regarding the implementation should be studied. This includes potential gains with respect to power- and/or energy consumption and robustness regarding Process, Voltage and Temperature (PVT) variations.

## 1.4 Overview of the Thesis

The chapters and appendixes contain the following:

- Chapter 1 presents the motivation for designing a low power RTC.
- Chapter 2 gives an introduction to power consumption and subthreshold design.
- Chapter 3 gives a description of the Real-Time Counter. It then shows the design of the RTC. It walks through the different components of the RTC and the hierarchy of logic cells.
- Chapter 4 presents a new design methodology. It introduces the BA-structure, and shows how the methodology was tested for comparison with conventional implementation.
- Chapter 5 Shows how the transistor was chosen and how supply voltage was determined. It also shows how delay through the longest path was tested and how a clock gate was implemented and placed to minimize power consumption.
- Chapter 6 present the layout of the different cells making up the RTC.
- Chapter 7 presents the transistor count and the RTC area.
- Chapter 8 shows how the RTC design was tested. It presents the testbench and the details behind each simulation.
- Chapter 9 presents all the results from previous chapters.
- Chapter 10 discusses some of the main aspects of the thesis.
- Chapter 11 summarizes the main results and contributions, and some ideas for future work.
- Appendix A shows VHDL code used to verify the main structure. It also presents a larger picture of the plot from VHDL-simulation
- Appendix B presents schematic for all the different cells that make up the RTC.
- Appendix C presents layout for all the cells that make up the RTC. The last Figure presents the whole layout.

# Chapter 2

# Background

## 2.1 Power consumption

Power consumption and power density on chip is one of the main challenges within Very Large Scale Integrated Circuit (VLSI) design. Complementary Metal-Oxide Semiconductor (CMOS) technology emerged in the 1990s as a result of non-sustainable power density in bipolar and nMOS designs. CMOS is still the main VLSI technology used primarily because of its power characteristics. It is slower than earlier technologies, but CMOS, in contrast to bipolar and nMOS design, consumes most of the power when changing state and not in a steady state. CMOS technology traditionally operates with a strongly inverted channel, which means a supply voltage above the transistor threshold voltage, $V_t$ [10].

As more designs target wireless and battery-powered applications, the research has been focusing on low-power design. Subthreshold operates with a supply voltage below $V_t$. This is the most efficient low-power technique as this reduces the energy consumed in active operation and dissipated leakage power.

### 2.1.1 CMOS power consumption

In digital CMOS design, Power Consumption can be divided into three categories as seen in equation 2.1 [10]:

$$P_{total} = P_{dynamic} + P_{static} + P_{short\ circuit} \tag{2.1}$$

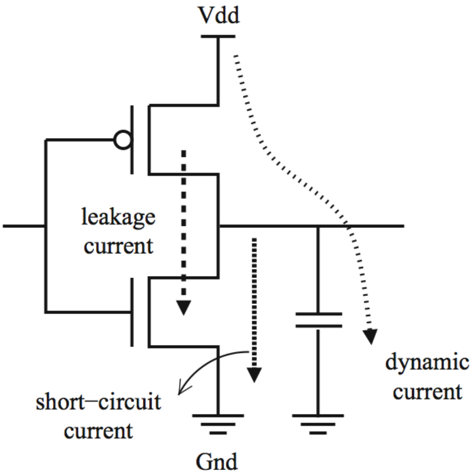


Figure 2.1: The dynamic, short circuit and leakage power components. [10]

Dynamic power has traditionally dominated the total power dissipation. As technology is scaling, the dynamic dissipation is reduced, and leakage increases. As a result, leakage and short circuit power needs to be accounted for when calculating total power dissipation. [11]

### 2.1.2 Dynamic Power

Dynamic and short circuit power consumption comes from the output changing state. The dynamic power is used to charge the load capacitance when the output transitions from 0 to 1. This charge is then drained when the output transitions back to 0 [13]. The dynamic power dissipation is given as:

$$P_{dynamic} = \alpha \cdot f \cdot C_L \cdot V_{DD}^2 \tag{2.2}$$

where $\alpha$ is the activity factor, $0 < \alpha < 1$, $C_L$ is the average load capacitance, and $f$ is the clock frequency.

As seen in equation 2.2, the dynamic power dissipation is proportional to the clock frequency and the supply voltage squared.

### 2.1.3 Short Circuit

Short circuit power dissipation comes from the current flowing from $V_{dd}$ to *Ground* while both the nMOS and pMos change states. Because of non-zero rise and fall times of the transistors, both transistors will be partly conducting current and hence make a direct current path from $V_{DD}$ to *Ground*. Short circuit power dissipation is usually not significant. [10]

### 2.1.4 Static power dissipation

Static power dissipation, or leakage power dissipation, is dissipated even in a stable state. There are several sources of leakage in CMOS technology where subthreshold leakage is the dominant [1].

Subthreshold leakage is a current going from the drain to the source even when the gate voltage is below the threshold voltage, $V_T$. This is a result of three effects [10]:

- First there is a weak inversion effect, which make carriers move by diffusion along the surface. This effect becomes significant when gate to source voltage get close to $V_T$. It is this leakage current we control and use in subthreshold design.
- The second effect is the Drain-Induced Barrier Lowering (DIBL). The DIBL is an effect equivalent of reducing the $V_T$ for higher drain voltages. When the drain voltage increase, the pn-junction between the drain and body increases and extends in under the gate. This affects the depletion region charge, which retains the charge balance by attracting carriers into the channel. This effect increase with higher drain voltages and shorter effective channel length. It is most prominent in weak inversion as the current is exponentially dependent on the surface potential.
- The third effect is the direct punch-through of electrons between drain and source. This is because the drain and source depletion regions electrically overlap deep in the channel.

## 2.2 Subthreshold Leakage Current Model

The following equation is used for modeling for the Subthreshold leakage current [14] [15].

$$I_D = I_{D0} e^{\frac{V_G}{nU_T}} ((e^{-\frac{V_S}{U_T}} - e^{-\frac{V_D}{U_T}}) \tag{2.3}$$

where $I_{D0}$ is the residual current in saturation for $V_G = V_S = 0$, given as:

$$I_{D0} \approx \beta e^{-\frac{V_T}{nU_T}} \tag{2.4}$$

where the transfer parameter of the transistor is:

$$\beta = \mu C_{OX} \frac{W}{L} \tag{2.5}$$

In these equations, $V_T$ is the threshold voltage, $n$ is the slope factor which is dependent of the oxide capacitance $C_{OX}$ and depletion capacitance $C_d$ as $n = (1 + \frac{C_d}{C_{OX}}$. $U_T$ is thermodynamic voltage $U_T = \frac{kT}{q}$. where $k$ is the Boltzmann constant and $q$ the elementary charge. For T = 300, or 27°C, $U_T$ is $25.8mV$ [15].

Equation 2.3 and 2.4 shows how the drain current is exponential dependent on threshold voltage and gate voltages. Since this is digital, $V_G$ is equal to supply voltage. This means that lowering the supply voltage will substantially reduce the drain current, and hence power consumption. Ideally, $V_{GS}$ in millivolts per decade of change in $I_D$ is $60mV/decade$ in room temperature [15].

### 2.2.1 Delay

The lower power consumption comes at a cost. Since drain current is reduced compared to strong inversion, it will take longer to charge the output of an transistor. In subthreshold, propagation delay for an inverter with output capacitance $C_g$ is given as [15]:

$$t_d = \frac{KC_gV_{DD}}{I_D} \tag{2.6}$$

where $K$ is a fitting parameter. Delay is therefor dependent of $I_D$ and hence growing exponentially with decreased supply voltages.

## 2.3 Robustness

### 2.3.1 Process, Voltage and Temperature

The exponential relationship between current and transistor parameters and conditions makes subthreshold very sensitive to Process, Voltage and Temperature variations, (PVT-variations) [1].

#### 2.3.1.1 Process

Transistors with equal structures in the layout do not have equal characteristics on the same chip. This is due to small variations in physical dimensions, and random dopant fluctuations [15] [1]. These are variations in the model parameters $V_{T0}$, $n$ and $\beta$.

#### 2.3.1.2 Voltage

For Ultra Low Power (ULP) devices, the voltage drops across the on-chip distribution is negligible since $I_{on}$ is orders of magnitude lower than in strong inversion. This means that voltage variations are due to small fluctuations in the external power supply. For battery powered devices, the supply voltage is relatively constant. Battery less systems suffer from more variations [1]. Supply regulators are therefor of great importance because of the exponential sensitivity in ULP systems.

#### 2.3.1.3 Temperature

Temperature also has a strong impact on the drain current. Both $V_T$ and channel mobility $\mu$ are temperature dependent, with the following relationship [15]:

$$\mu(T) = \mu(T_0)(\frac{T}{T_0})^{-M} \tag{2.7}$$

$$V_T(T) = V_T(T_0) - KT \tag{2.8}$$

where $K$ is the voltage temperature coefficient, $M$ is the mobility-temperature exponent and $T_0 = 300K$. $M$ and $K$ is technology dependent with typical values of $1,5$ and $2,4mV/K$ respectively [4].

The model match well across most temperatures , but slightly underestimate leakage at high temperatures. The lower mobility is the dominant in strong inversion which leads to slower circuits in high temperatures. In subthreshold, the lower $V_T$ will dominate where colder temperatures leads to slower circuits. This also leads to higher leakage in higher temperatures [15].

## 2.4 Power-Delay Product

For subthreshold design, Power-Delay Product is used as a figure of merit. It is calculated as power times delay which gives energy in Joule [15]. This is therefor a good measurement of cost per operation.

$$PDP = Power \cdot Delay \tag{2.9}$$

## 2.5 Tools

The main set of tools used in this thesis is part of the Cadence toolset. The design is done in Virtuoso where Virtuoso Accelerated Parallel simulator (APS) is used to simulate and perform Monte Carlo Analysis. Mentor Graphics Calibre is used to extract parasitic data from layout, in order to simulate the layout. This can be done with both xRC for 2D and xACT3D . 3D is more accurate and take into account how transistors and wires overlap. Simulation performed with 3D extraction is extremely computational intensive. It has not been possible to run Monte Carlo Analysis with 3D extraction on the available equipment. The VHDL simulations are performed with Aldec Active-HDL. Most of the circuit drawings presented in this thesis are made at *www.circuitlab.com*.

# Chapter 3

# RTC Design

## 3.1 Real-time Clock

A Real-Time Clock (RTC) is an important part of a microcontroller. This design approximates the functionality of the *EFM32G* series microcontroller from Energy Micro.

The Real Time Clock is used to keep track of time. The counter can be reset, and later read as a stop watch to see how much time that has passed. It also has two compare registers which works as an alarm clock. The compare register is set with a value and it will generate an interrupt when the counter reaches this value. One of the comparators can also reset the counter when the desired value is reached. This makes it possible to put the microcontroller in deep sleep mode, and have the RTC awake the CPU. The compare register can also be used with an external component to generate various waveforms [3].

### 3.1.1 Description

The RTC consists of a 24-bit counter and two compare registers. The counter is clocked either by a crystal or RC oscillator on $32.768kHz$. The clock driving the RTC contains a pre-scaler which gives the desired time-resolution [3].

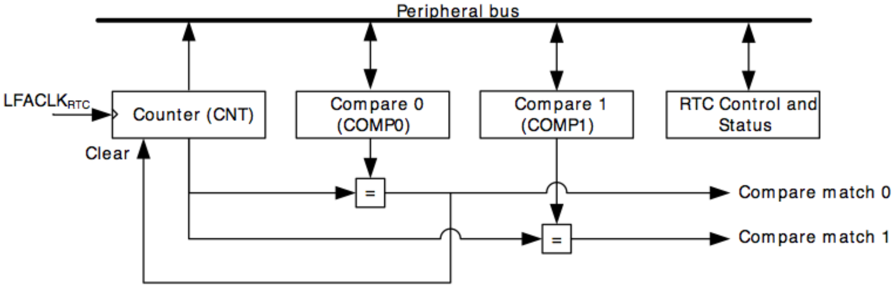


Figure 3.1: Energy Micros RTC Overview [3].

As seen in Figure 3.1, the clock contains a counter and two compare registers. The registers output a compare match signal when the counter is equal to the register. There is also a feedback from *Compare 0* which can reset the counter if a signal *COMP0TOP* in the control is set high.

## 3.2 Implementation

### 3.2.1 Flip-Flops

Both the counter and the compare uses flip-flop registers. To be able to reset the counter, it needs a flip-flop with asynchronous reset. The *Compare* register is set with their own *enable* signal, and thus do not need any reset. The C$^2$MOS flip-flop has been chosen as previous studies show that it is one of the preferred and since it does not contain any pass- or transmission-gates [2] [6] [15].

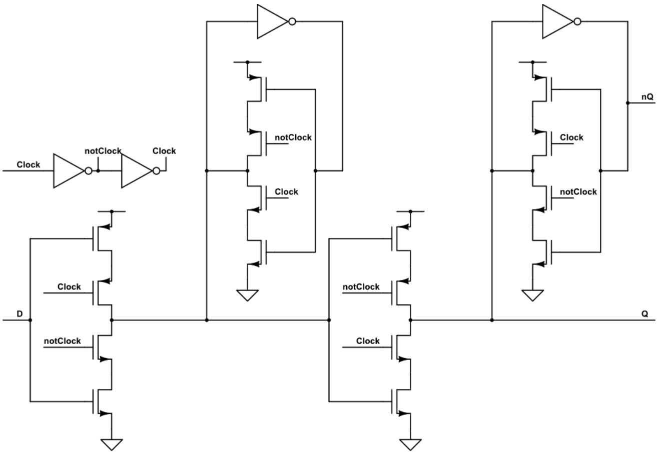


Figure 3.2: C$^2$MOS Flip-Flop.

As Figure 3.2 shows, the C$^2$MOS consists of 4 inverters and 4 clocked inverters. For the counter, this design was modified to make a flip-flop with asynchronous reset. The first inverter on the input of the clock signal is changed to a NOR gate. This makes sure that the internal signal *Clock* is *High* and *not Clock* is *Low* while the Flip-Flop is being reset. This state ensures that the data input is not read. The reset signal is then used to set the state in two internal nodes in the flip-flop. This modified C$^2$MOS is shown in Figure 3.3. This design was verified by simulation.

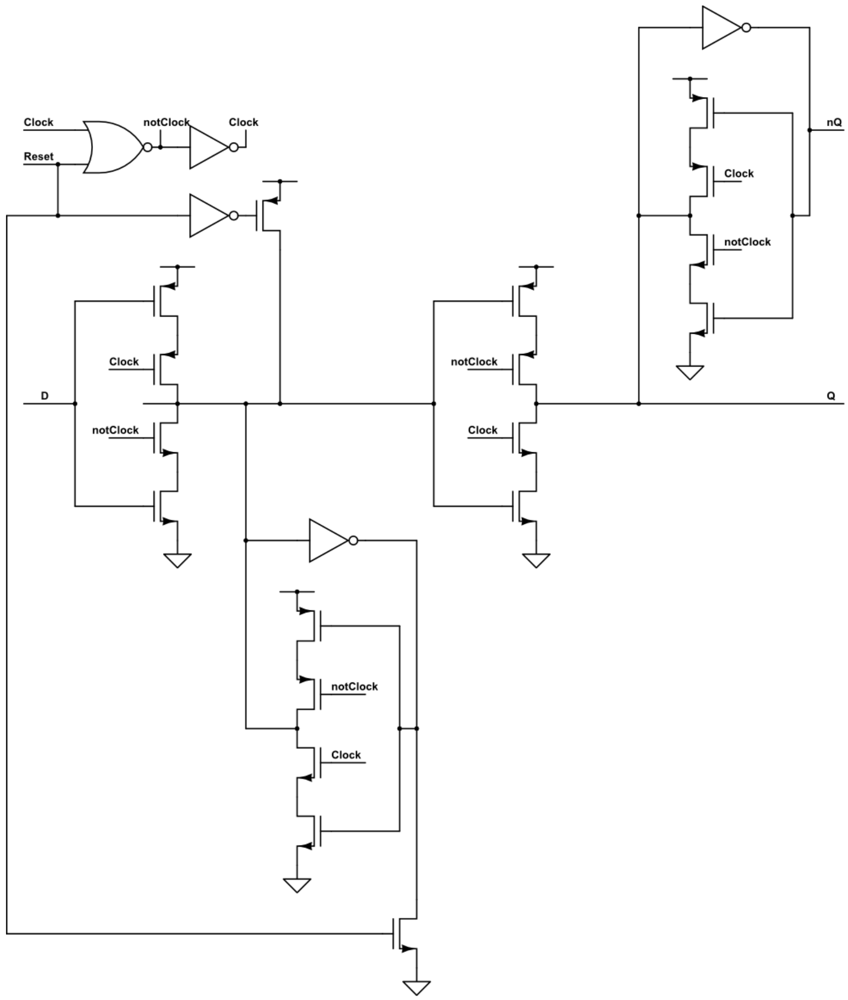


Figure 3.3: C$^2$MOS Flip-Flop with asynchronous reset.

### 3.2.2 The counter

The 24-bit counter consists of 25 flip-flops and 23 adders. Since the counter only increment, only half adders are needed to add each bit with a carry. This considerably decrease transistors count compared to full adders.

The *Least Significant Bit (LSB)* toggles on every positive clock-edge, and therefor the data input can be the inverted output and hence do not need an adder. One flip-flop is connected to the carry from *Most Significant Bit (MSB)* and used to hold the *overflow* signal *High* for one clock period when the counter goes back to 0. The half adder used is a conventional implementation using an XOR, NAND and INVERTER as seen in Figure 3.4.

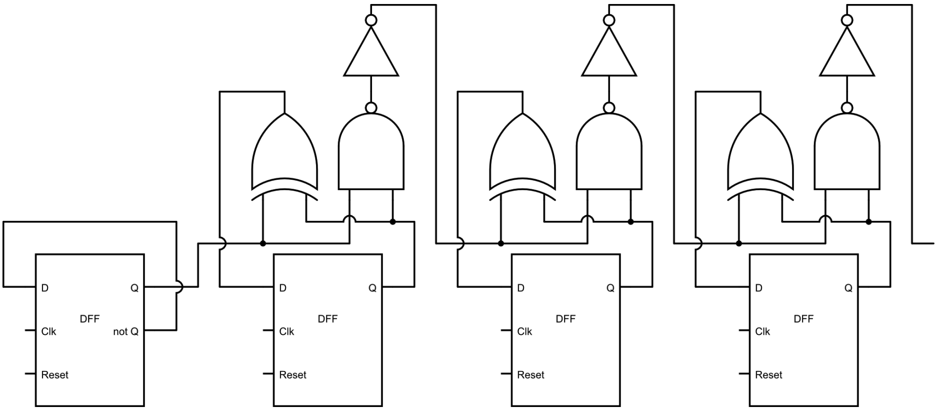


Figure 3.4: Four first counter bits.

### 3.2.3 Compare circuit

The compare circuit compares each bit of the counter to each bit of the compare register. In principle, *XNOR* gates, which has a *High* output if the inputs are equal, can be used to compare the counter to the register. Then *AND* gates can be used to compare all the *XNOR* gates to see that all the respective bits from the counter and the compare register are equal.

To save the use of inverters in *AND*-gates, every other bit consists of *XOR* / *XNOR* to compare the bits, and *NOR* / *NAND* to see if all the bits are equal. The structure is shown in Figure 3.6 and Table 3.1 is given for quick reference.

To enable clear on match, a flip-flop is connected to the *Compare 0* match signal. If *COMP0TOP* is set high, the output from the flip-flop will reset the counter one clock cycle later than the one producing the match. This enables the counter to reach the top value before becoming zero. The output from the circuit shows in Figure 3.5 is the input to the reset tree which will be discussed in Section 5.3.

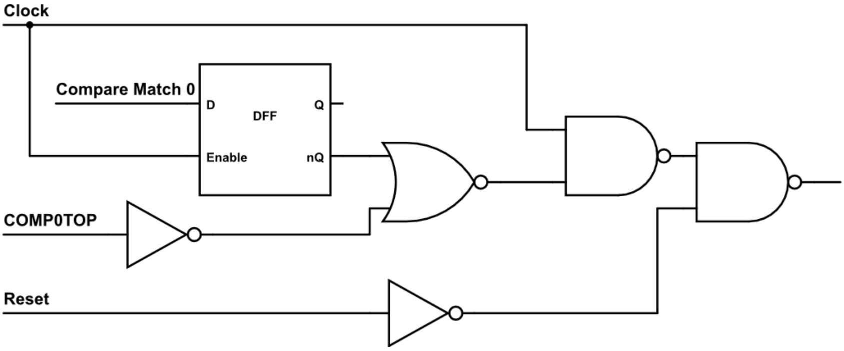


Figure 3.5: Clear on match circuit.

| A | B | | NAND | NOR | XOR | XNOR |
|---|---|---|---|---|---|---|
| 0 | 0 | | 1 | 1 | 0 | 1 |
| 0 | 1 | | 1 | 0 | 1 | 0 |
| 1 | 0 | | 1 | 0 | 1 | 0 |
| 1 | 1 | | 0 | 0 | 0 | 1 |

Table 3.1: Truth-tables for the used gates.

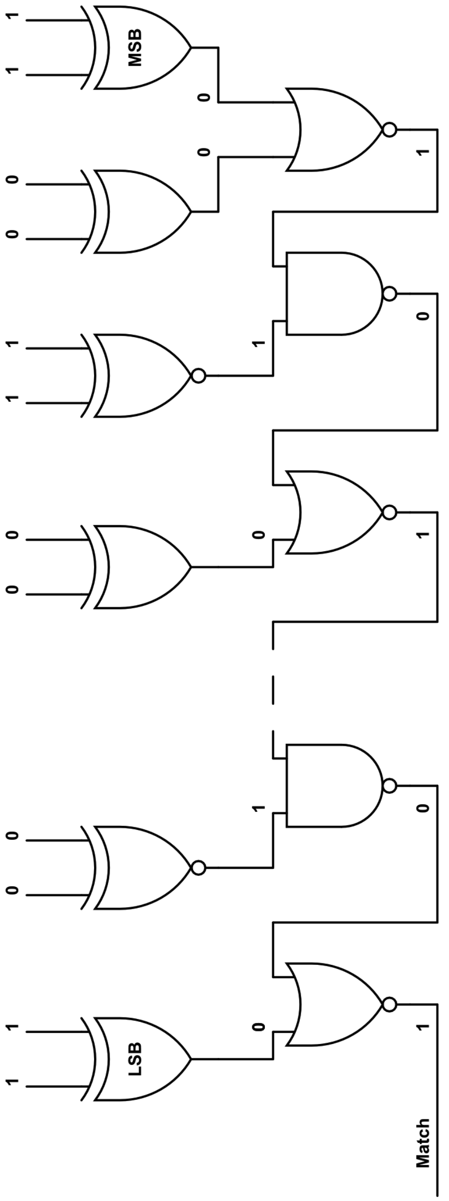


Figure 3.6: Comparator circuit.

## 3.3 Hierarchy

The design is done with a bottom-up approach using functional blocks. The following subsections will describe each block. A figure from each kind is shown, and all blocks are presented in Appendix B.

### 3.3.1 compBit-type

The comparator is split up bitwise and consists of three types of blocks: *compBitXorNor*, *compBitXnorNand* and *compBitXor*. Each block consists of a C$^2$MOS flip-flop and gates to compare the comparator to the counter and the next more significant comparator-bit as shown in Figure 3.7. CompBitXor without Nor is used for the MSB.

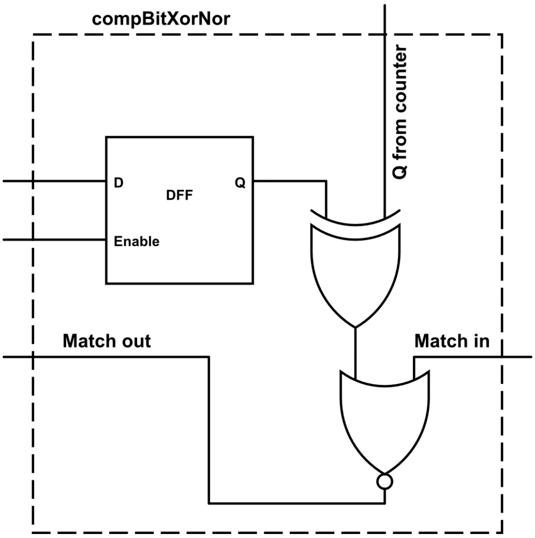


Figure 3.7: Comparator-bit with Xor and NOR.

### 3.3.2 bitWith-type

The three comBit types are then put into four kinds of blocks which will make up the 24-bits in the RTC: *bitWithXorNor* *bitWithAdderXorNor*, *bitWithAdderXnorNnand* and *bitWithAdderXor*, . In general, a C$^2$MOS with asynchronous reset for the counter is put together with two compBits, one for each comparator. The exception is for LSB and MSB. LSB do not contain a half adder and bitWithAdderXor without Nor is for MSB. The two inverters after the flip-flop acts as a buffer to maintain a low fan-out from the flip-flop.

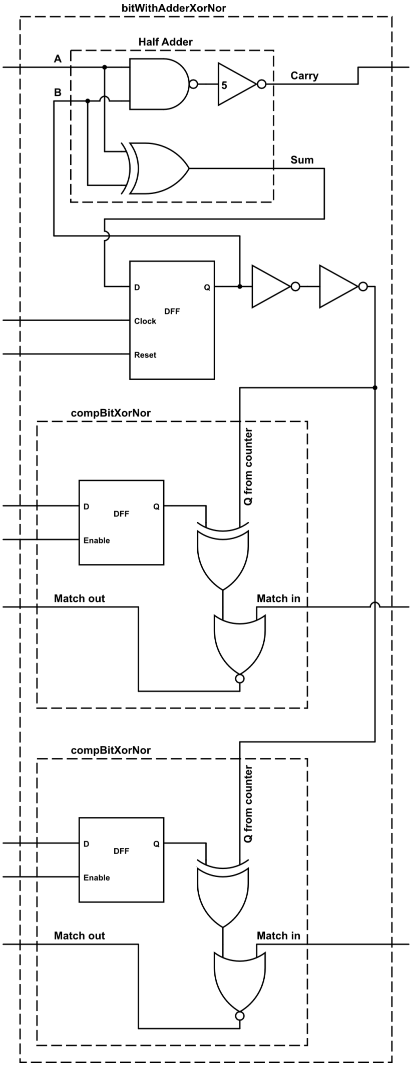


Figure 3.8: Bit with Adder and two comparator blocks with Xor and Nor.

### 3.3.3 Verification

To verify the counter and the comparator circuit, the structure for the first four bits was written in VHDL and simulated. VHDL code can be seen in Appendix A. The comparator register is loaded with the number 9 and *LastCarry* is the carry from 4th-bit. In the design, carry from bit 24 goes to a flip-flop which will make the overflow signal go *High* when the counter go to zero, and not at the last value.

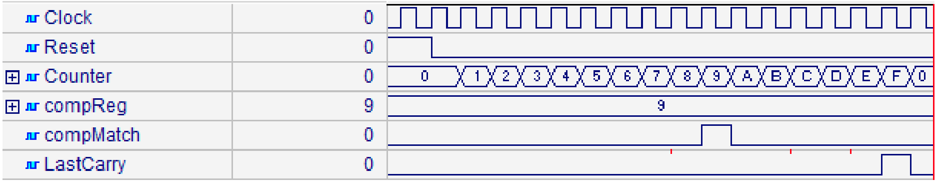


Figure 3.9: VHDL simulation of the four first bits.

# Chapter 4

# A new design methodology

Robustness is the main challenge in subthreshold design, and there has been several design strategies proposed to cope with sensitivity for PVT-variations. Minority-3 gates has been suggested as a general building block which yield more robustness than traditional static CMOS [8]. The idea of a general building block inspired the methodology to be presented. This methodology was conceived in a collaboration between the author and Professor Snorre Aunet of the Norwegian University of Science and Technology. The proposed name is therefor *BA-structure* or *BA-gates*.

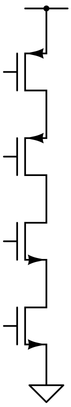


Figure 4.1: A slice

## 4.1 Introduction of BA-Structure

The BA-gates are combinational logic implemented using uniform blocks. This uniform block consists of a fixed stack of transistors, called a slice, as seen in Figure 4.1. The hight, number of transistors, of the slice is decided by desired fan-in. A single slice can implement an inverter and clocked inverter. Putting more slices together can in principle make any combinatorial logic function. This includes, but is not limited to traditional Boolean logic gates: NAND, AND, NOR, OR, XOR, XNOR, INV, BUF, as well as any threshold logic function. In addition, one may implement memory like for example latches and flip-flops. By combining logic and memory for finite state machines, this could enable in principle any general digital system.

Figure 6.2 and 4.3 shows how this methodology is used to implement NAND and NOR by putting two slices together, compared to conventional implementation.

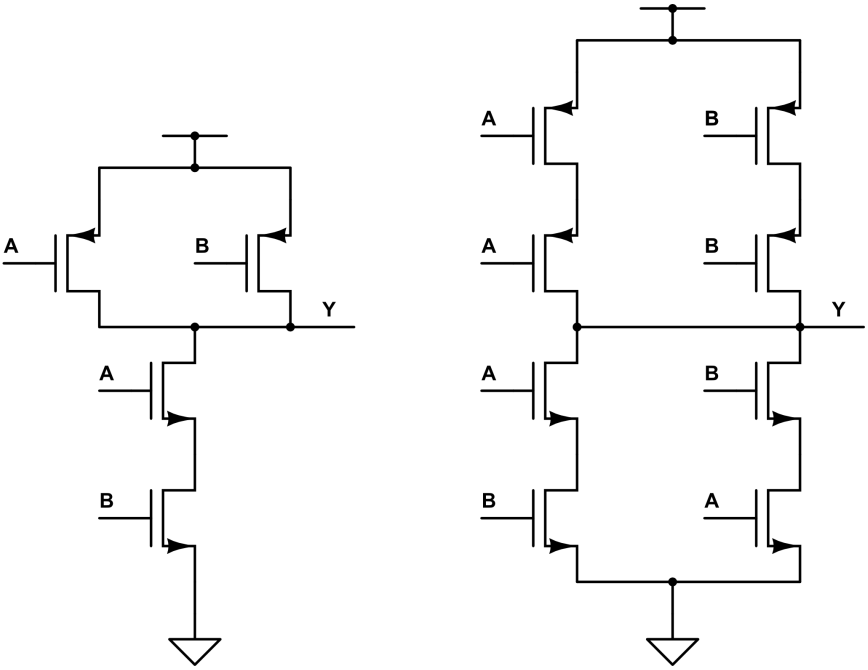


Figure 4.2: Conventional NAND compared to NAND with BA-gate

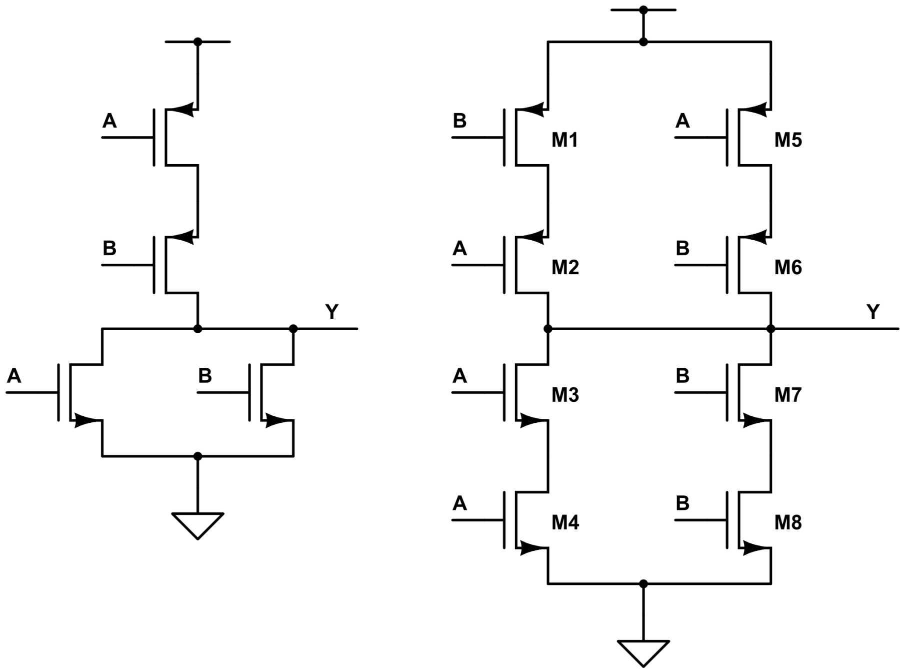


Figure 4.3: Conventional NOR compared to NOR with BA-gate

## 4.2 Benchmark

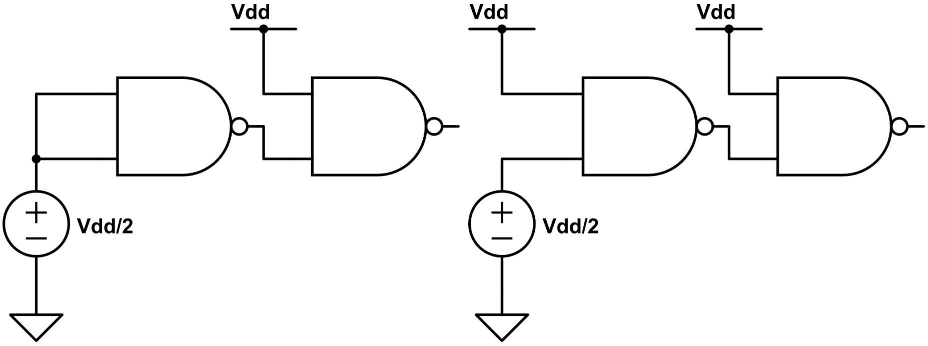
To benchmark and compare the robustness of the two topologies, both 2-input NAND-gate shown in Figure 6.2 was implemented and tested. First different ways of balancing the transistors are tested. The best case for both topologies are then used in Monte Carlo simulation to give the statistical distribution of: Delay, Power, Power-Delay Product and Leakage over process and mismatch variation.

The simulation was done with Standard Threshold General Purpose *(stvgp)* transistors in $65nm$ technology. $V_{dd} = 250mV$, gate-length = $90nm$ and temperature = 20°C. Threshold-voltage for this transistor type was measured in the simulator to be: pMOS = 318mV and nMOS = 361mV.

### 4.2.1 Transistor balancing

Two different kind of transitions on input makes the NAND-gate change state on the output: One input *High* and the other changing state and if both inputs change state. See Table 3.1.

The nMOS and pMOS can be balanced by stopping the transition half way through at $V_{DD}/2$, and tune the n/p-MOS transistor width to make the output $V_{DD}/2$. This gives two possible ways of balancing the NAND-gates. As Figure 4.4 shows, both cases was tested with a NAND-gate as load.



(a) Both input at $V_{DD}2$

(b) One input at $V_{DD}/2$

Figure 4.4: Balancing NAND-gates

The two different transistor sizes was recorded and tested in respect to both transitions, toggling one and two inputs. Output delay for both rising- and falling-edge and power was recorded. PDP for each transition was calculated using the longest delay. The results can be seen in Table 9.1 and 9.2.

The proper method for balancing the NAND-gates is found by evaluate the PDP for both balancing methods. The best *worst case* PDP decides the method for balancing and it is also the most interesting transition for the Monte Carlo simulations.

### 4.2.2 Robustness

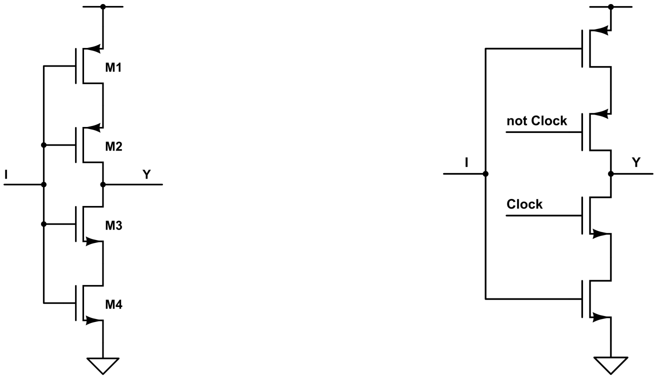
The Monte Carlo simulations are performed with three NAND-gates connected as a ring oscillator. The conventional implementation is balanced with one input at $V_{DD}/2$ and the other at $V_{DD}$. The BA-gate topology is balanced with both inputs at $V_{DD}/2$. Both topologies have *worst case* PDP when toggling both inputs, so this is the transition tested. Leakage is measured for all input vectors: One input *High*, two inputs *High* and both inputs *Low*. The worst case is recorded in the appropriate results Table.

To test robustness, Monte Carlo simulation with 200 runs was used. The results can be seen in Section 9.2 and 9.3.

# Chapter 5

# RTC Design part II

The new design methodology with *AB-gates* are used in the rest of the design. The gates used in the design are INVERTER, CLOCKED INVERTER, NAND, NOR, XOR and XNOR. All except the inverters are 2-input gates, therefor the slice has a fixed stack hight of 2 nMOS and 2 pMOS. NAND and NOR are shown in Figure 6.2 and 4.3. The INVERTER has all the transistor gates connected to the input as shown in Figure 5.1a. The clocked INVERTER is a conventional implementation. It is important that the transistors connected to the clock signal are placed closest to the output, see Figure 5.1b. This is to prevent charge sharing which makes the output voltage swing to decrese [12].

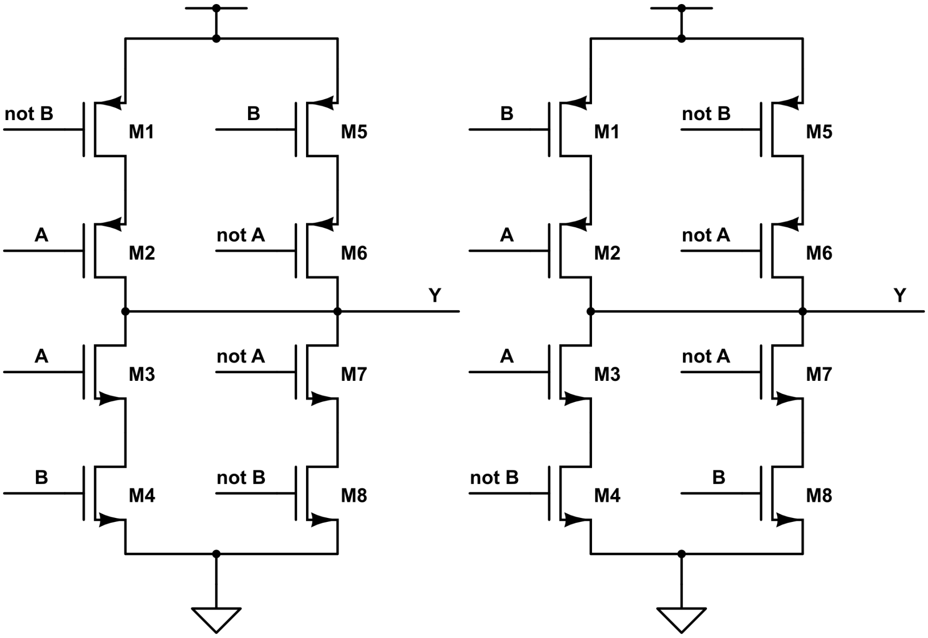


(a) INVERTER

(b) Clocked INVERTER

Figure 5.1: Implementation of INVERTER gates.

The XOR and XNOR used are a conventional implementation which naturally conforms to the design methodology with 2 pMOS and 2 nMOS [7]. See Figure 5.2. The input to XOR and XNOR gates are connected to the $C^2$MOS flip-flop and the half adder. This means that the input signal already exists as inverted in these circuits, and therefor the gates do not need to have internal inverters. A consequence of this is that the bitWith-type presented in Section 3.3.2 needs two more inverters to buffer the flip-flop output $\overline{Q}$.



(a) XOR (b) XNOR

Figure 5.2: Implementation of exclusive gates.

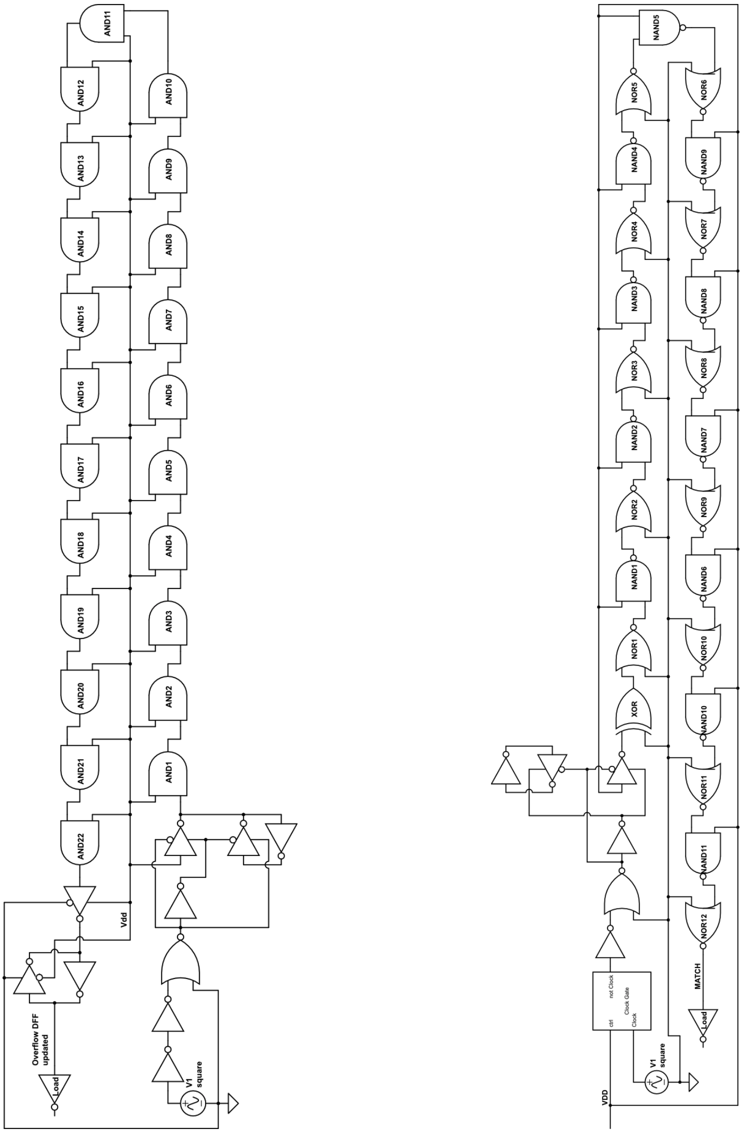
## 5.1 Choosing transistor

The choice of transistor was primarily done by looking at the speed requirements and transistor count in Table 7.1 The circuit is designed to work at $32.768kHZ$ which is fairly slow. As 90% of the circuit is in a stable state, the High Threshold Low Power (hvtlp) transistor was chosen. This transistor has the highest $V_T$ in $65nm$ technology and hence reduce leakage to a minimum.

This transistor has been measured to have a threshold voltage for nMOS $V_T 718mV$ and pMOS $V_T = 635mV$. The threshold voltages changed marginally from schematic to layout.

## 5.2 Longest Path

To choose supply voltage, the longest path in the design was used to simulated delays for various voltages and transistor sizes. The RTC runs on 32.768kHz which give $30,518\mu S$ between every positive clock edges. This means that the flip-flop for the *overflow* and *clear on match* needs a stable updated state before it is clocked. Both of these situations which is the two longest paths in the design was simulated. The testbences are presented in Figure 5.3. The transistor sizes was found for $400mV$, $450mV$, $475mV$ and $500mV$. These voltages was used to test delay at different supply voltages. Results are presented in Section 9.4.1. The simulation is performed with temperature -40$^{circa}$C. Since $V_{DD} = 500mV$ gave a result within 30% of the limit, it was performed Monte Carlo Analysis for this supply voltages.



(a) Overflow Testbench

(b) Match Testbench

Figure 5.3: Testbench Longest path.

## 5.3 Optimal Fanout

Now we have decided on a supply voltage of 500mV with *hvtlp* transistor. Gate length is 90nm, widths: pMOS = 270nm and nMOS = 605 nm. This will be used for the rest of the design.

Both enable signals to the comparator registers, the reset and the clock to the counter flip-flops needs a distribution tree. A large fanout gives a slower propagation through the gates, wile smaller fanout gives a higher tree which gives more gates to go through. For 24 leaf nodes in the tree, the hight of the tree as a function of fanout is given as:

$$\lceil Hight \rceil = log_x(24)$$

To figure out optimal fanout, simulations was performed. Testbench is shown in Figure 5.4. Delay from *In* to *Out* was measured for fanout of 2 through 7. Fanout for the enable, reset and clock tree design is set to 5, while the rest of the design has fanout as small as possible. All the results are presented in Section 9.5.

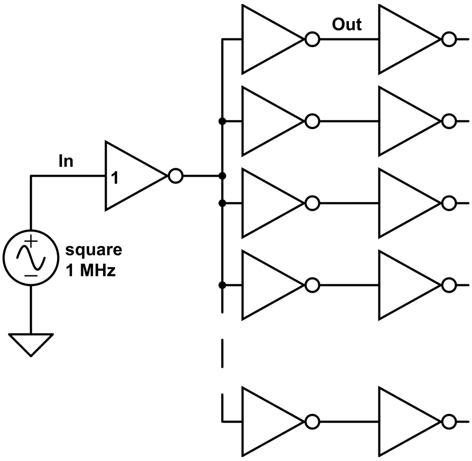


Figure 5.4: Fanout test

## 5.4 Clock gate

### 5.4.1 Placement

A clock gate is used in order to minimize energy used on clocking the flip-flops in the counter. The cost of clocking the counter up to an overflow can be calculated by adding the number of times flip-flops are clocked in front and after the clock gate.

The number of times the flip-flops in front of the clock gate is clocked, are the number of flip-flop times number of increments to overflow:

$$2^{24} \cdot x$$

where x is the number of flip-flops in front of the clock gate. The number of times flip-flops are clocked by the clock gate is the number of flip-flops times the number of increments to overflow those flip-flops:

$$(24 - x) \cdot 2^{24-x}$$

If we put these together, we can describe total cost of clocking as:

$$2^{24}xk + (24 - x)2^{24-x}k \tag{5.1}$$

The cost is in terms of *clocked flip-flops* to reach overflow on 24 bits. The numbers for the six first placements are given in Table 5.1, where 0 is the circuit without a clock gate. The rest are plotted in Figure 5.5. The unit on y-axis is ignored since the goal is to find the minimum point. This shows that optimal placement of the clock gate is after the fourth flip-flop which gives a reduction of 78% less clocked flip-flops.

| Placement | Cost |
|---|---:|
| 0 | 402653184 |
| 1 | 209715200 |
| 2 | 125829120 |
| 3 | 94371840 |
| 4 | 88080384 |
| 5 | 93847552 |

Table 5.1: Cost of clocking with different placement of clock gate

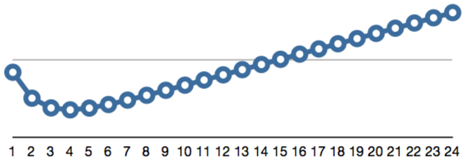


Figure 5.5: Cost of clocking with different placement of clock gate

### 5.4.2 Implementation

The clock gate is implemented with a C$^2$MOS latch and a NAND gate as shown in Figure 5.6. Signal $Q$ is updated while *Clock* is *Low* and holds when the clock is *High*. If *ctrl* and thereby $Q$ are *High* on a positive clock edge, the output will go *Low*. The *ctrl* signal is connected to carry from half adder at the fourth flip-flop.

The clock gate is designed to have as short logic depth as possible. Since signal $Q$ is set up before positive clock edge, the total delay through the clock gate is the NAND gate. Having the output from the clock gate inverted, makes the total delay to the flip-flops as small as possible. Since fanout was decided to be 5 in the clock-tree, it is sufficient with one level of inverters to distribute the clock from the clock gate to the 20 flip-flops.

The whole clock-tree is shown in Figure 5.7. The clock skew between the flip-flops should be minimal as all clock paths goes through two gates before reaching the flip-flops. Ideally, the clock should reach overflow flip-flop and MSB first and LSB last to make sure that the carry is not cleared before the flip-flop has locked in the value. To slow down clearing of the carry signal, two inverters are placed on the carry-path when passing the clock gate. The clock going to the flip-flop for overflow is not shared to other flip-flops, and should be the fastest. The clock going to bit 0-3 is shared with the COMP0TOP flip-flop which reads the match signal. This should make the first bits marginally slower than the upper bits.

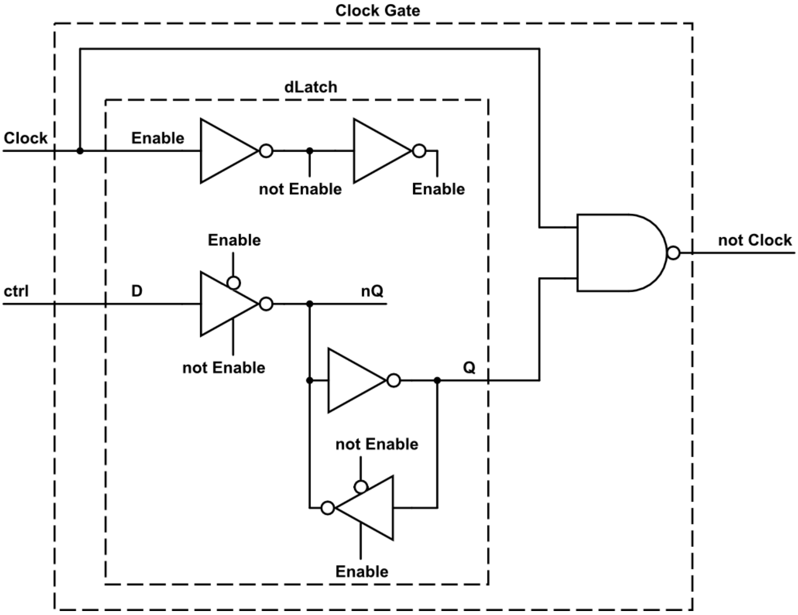


Figure 5.6: Cost of clocking with different placement of clock gate

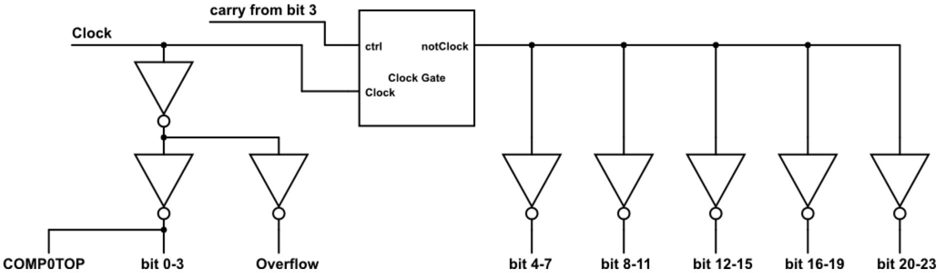


Figure 5.7: Clocking circuit

## 5.5 Reset and Enable tree

The reset signal for the counter and the two enable signal to comparator registers also need a tree for distributing the signals. The reset tree shares the signal to bit 21-23 with the flip-flop for overflow. A fanout of first four and then three should give a fast and even signal distribution, as shown in Figure 5.8.

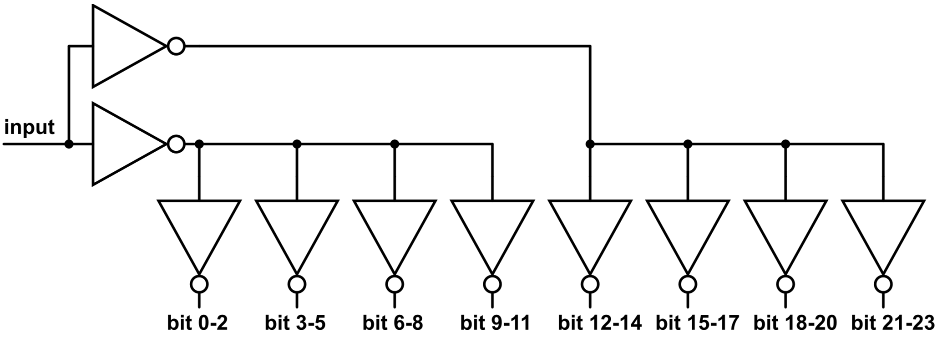


Figure 5.8: Enable and Reset tree

# Chapter 6

# Layout

The layout is performed in the same hierarchy as the schematics design and all the blocks can be seen in Appendix C. All the cells share the basic setup and metrics, which gives a very symmetrical and regular design. As AB-structure is used, all the nMOS transistors have equal dimensions across all cells. The same is true for all pMOS transistors. Gate length is set to $L_n = L_p = 1.5 \cdot L_{min} = 90nm$ to minimize process variations [5]. The design is done with regular poly pattern in a single direction with a fixed poly (PO) pitch [9]. The layout is also done in accordance with the design rules for the 65nm technology by STMicroelectronics.

## 6.1 General structure

Each metal layer are used in only one direction: Metal 1 (M1) only vertical, Metal 2 (M2) only horizontal and so on. Exception is done for $V_{DD}$ and GND as M1. There is also some small distances between the stacked transistors that is all done in M1. The VIA from M1 to PO is placed in the middle of the PO to ensure symmetry. Horizontal paths in M2 are at defined heights over the gate. It is aligned with the middle point on the PO. The distance between each path is the minimum M1 to M2 VIA distance. This gives 7 designated paths that do not overlap the transistors. See Figure 6.1. The *nwell* is stretched $2\mu m$ in all directions from the gate on the pMOS transistors to minimize the well proximity effect [9]. The distance between the stacked transistors are $580nm$ which gives room for three M2 layers in-between. The square marking the mandatory nwell around the pMOS is used to align transistors sideways to get a uniform PO pitch.

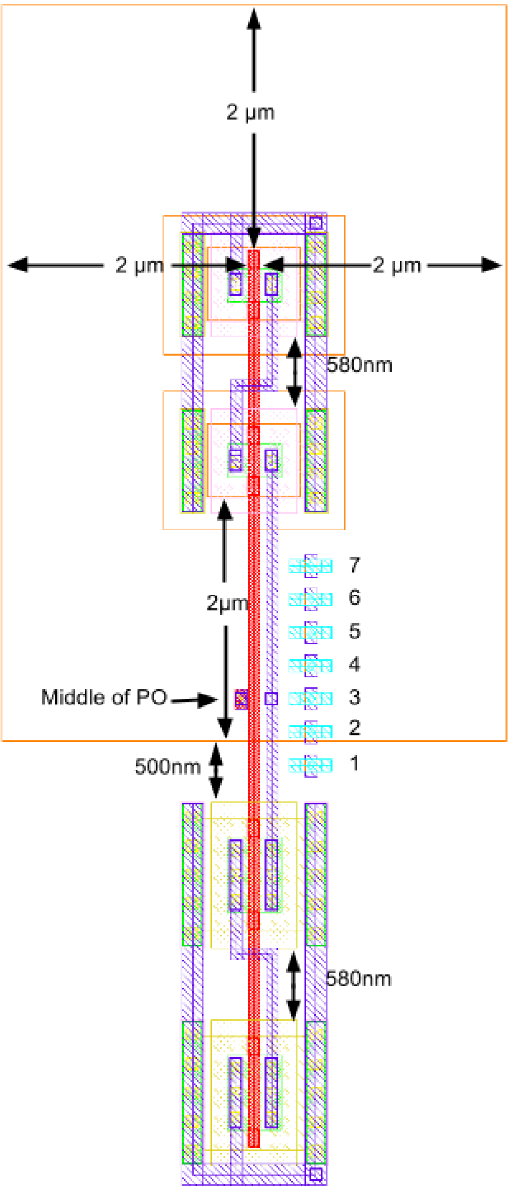
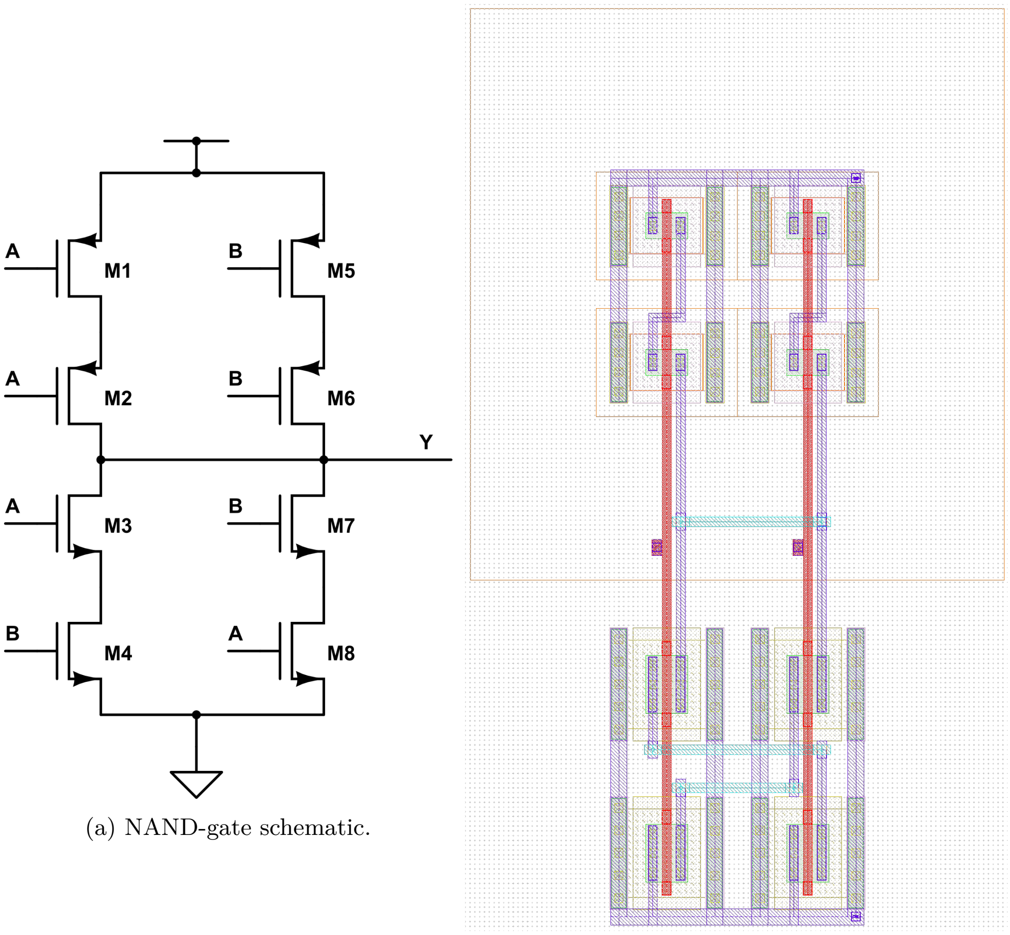


Figure 6.1: Physical layout structure.

## 6.2 NAND- and NOR-gates

Some of the transistors have been moved compared to the schematic. This is done to keep the poly pattern to only go in a single direction. The NAND is shown in figure 6.3a. By swapping transistor M4 and M8, the poly for signal $A$ and $B$ can go from top to bottom as shown in Figure 6.3b. The same is done for the NOR-gate.



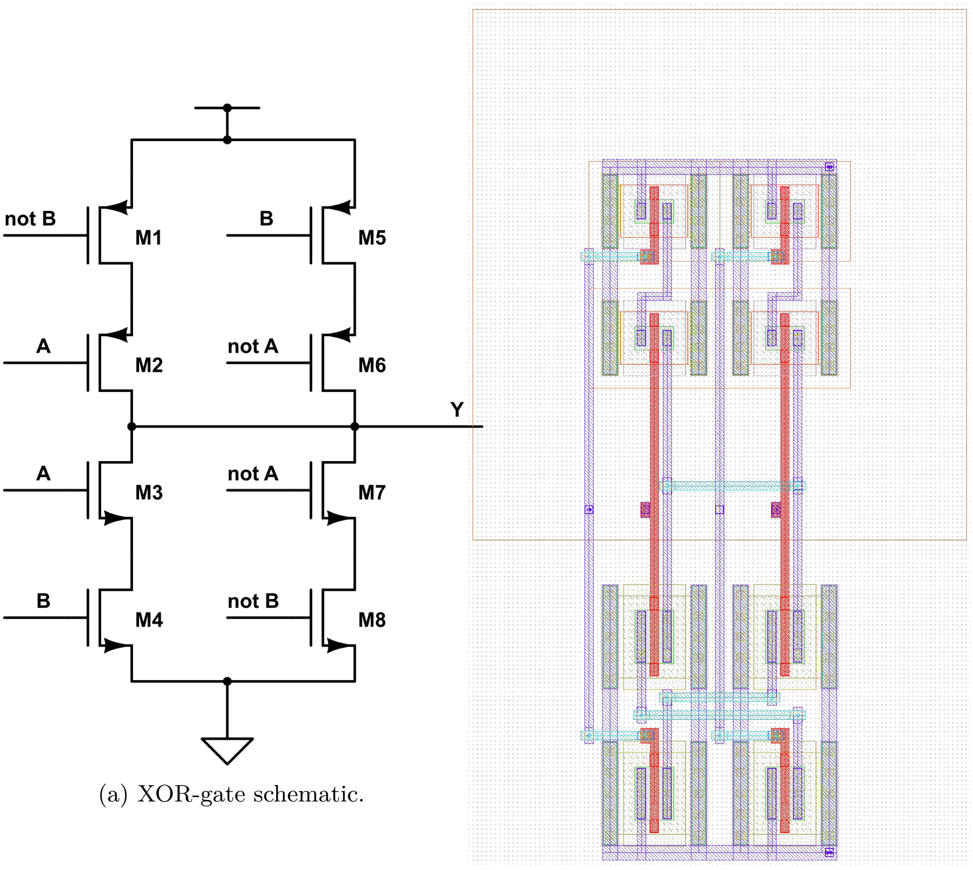
(a) NAND-gate schematic.

(b) NAND-gate layout.

Figure 6.2: NAND-gate schematic and layout.

## 6.3 XOR- and XNOR-gates

Both XOR and XNOR have four different signals on the transistor gates, and therefor it is not possible to make the same poly go from top to bottom. The transistors that have signal *A* and *not A* is placed so the drain is connected to the output. This makes the poly extend as much as posseble in the vertical direction. Transistor M4 and M8 is also swopped to make signal *B* and *not B* go in a single vertical axis to maintain as much symmetry as possible.



(a) XOR-gate schematic.

(b) XOR-gate layout.

Figure 6.3: XOR-gate schematic and layout.

# Chapter 7

# Transistor count and area

Table 7.1 gives the transistor count for the whole design. The transistors are grouped into *dynamic* and *static*. All the transistors in front of the clock gate is calculated as dynamic, except the comparator circuit which not usually get a match at every clock. The transistor count behind the clock gate is divided on $2^4$ which is how often they will be clocked. For the flip-flops in the counter the calculation is as follows:

$$\text{Dynamic:} \quad 44(4 + \tfrac{24-4}{2^4}) = 231$$

$$\text{Static:} \quad 1056 - 231 = 825$$

The physical layout of the RTC is $452.44 \mu m$ wide and $39.87 \mu m$ high, which gives a total area of $18038.8 \mu m^2$ or about $0.018 mm^2$.

| **Counter** | Used | Transistor count | Dynamic | Static | Total |
|---|---|---|---|---|---|
| XOR | 23 | 8 | 34 | 150 | 184 |
| NAND | 23 | 8 | 34 | 150 | 184 |
| INVERTER | 27 | 4 | 18 | 90 | 108 |
| Flip-flops w/reset | 24 | 44 | 231 | 825 | 1056 |
| Clock Gate | 1 | 28 | 9 | 19 | 28 |
| Clock tree | 8 | 4 | 32 | 0 | 32 |
| Reset tree | 10 | 4 | 0 | 40 | 40 |
| ClearOnMatch | 1 | 64 | 0 | 64 | 64 |
| **Comparator** | | | | | |
| Flip-Flops | 48 | 32 | 0 | 1536 | 1536 |
| bitWith-buffer | 96 | 4 | 84 | 300 | 384 |
| XOR/XNOR | 48 | 8 | 0 | 384 | 384 |
| NAND/NOR | 46 | 8 | 0 | 368 | 368 |
| Enable tree | 20 | 4 | 0 | 80 | 80 |
| **Dummy** | | | | | |
| bitWithXorNor | 1 | 4 | | | 4 |
| bitWithAdderXor | 8 | 4 | | | 32 |
| bitWithAdderXorNor | 44 | 4 | | | 176 |
| top cell RTC | 15 | 4 | | | 60 |
| Total dummys | | | | | 272 |
| **Total** | | | **442** | **4006** | **4720** |
| | | | **9.94%** | **90.06%** | |

Table 7.1: Total transistor count.

# Chapter 8

# RTC Simulations

Simulation on the final design has been done to verify behavior and evaluate performance, power and robustness. The behavior of design was tested with the testbench as seen in Figure 8.1. Both comparators, reset, overflow, and COMP0TOP was tested successfully. Performance in regards to delay and power consumption is shown in Section 8.1 and 8.2.

## 8.1 Delay

To test delay of the longest paths, the simulation had the following setup:

- Simulation performed on schematics and layout with 2D parasitic extraction.
- Monte Carlo with 100 runs.
- Temperature -40°C.
- Counter was not reset, but the flip-flops was set to a stable initial value of 0xFFFE.
- Both comparator register is set to 0x0000.
- COMP0TOP was set to *Low*.
- Clock period: $30\mu S$.

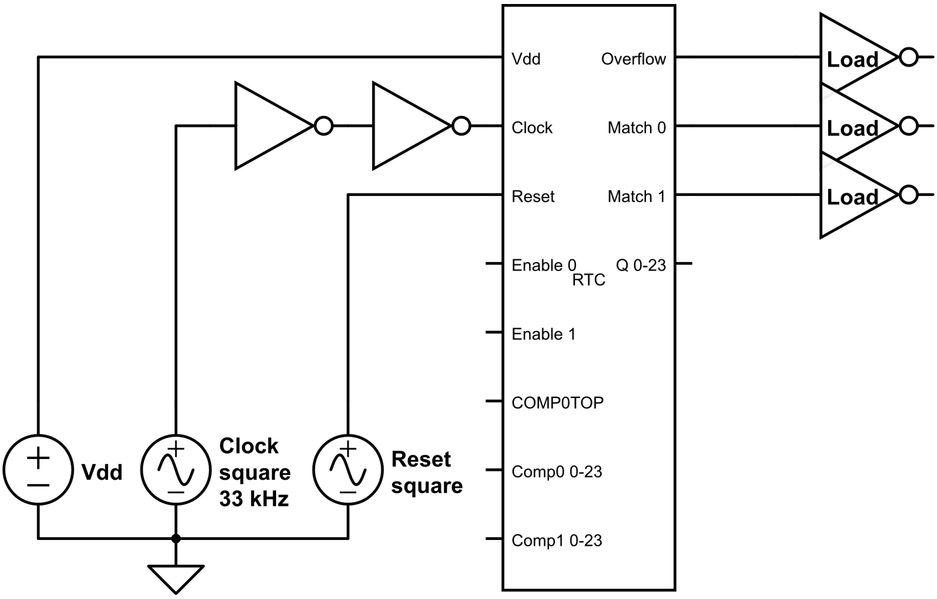


Figure 8.1: RTC testbench.

- Overflow signal was measured at carry from last bit, at overflow flip-flop input.
- First clock input generates the carry overflow signal.
- Second clock input makes the clock overflow to zero, and generate match signal.

Results can be found in Section 9.6.1 and 9.6.2.

## 8.2 Power consumption

To test power consumption, the simulation had the following setup:

- Simulation performed on layout with 2D parasitic extraction.
- Monte Carlo with 10 runs for each temperature.
- Temperature -40°C, 20°C and 80°C.
- Counter was reset, and no initial values.
- Comparator 0 was set to 0x0008.
- Comparator 1 was set to 0x0800.
- COMP0TOP was set to *Low*.
- Clock period: $30\mu S$.
- Power was calculated as average consumption of running 2mS.

Because of longer simulation time, it was not feasible to run 100 Monte Carlo simulations for each temperature, but this should give a good indication. Results can be found in Section 9.6.3.

## 8.3 Delay and Power consumption for various supply voltages

To figure out how much increase of supply voltage would impact delay and power consumption, simulations was performed for voltages from $V_{DD} = 500mV$ to $V_{DD} = 600mV$ with $10mV$ steps.

The simulations was performed as single runs and not Monte Carlo, but otherwise as described in Section 8.1 and 8.2. The results are presented in Section 9.6.4.

# Chapter 9

# Results

## 9.1 Gate topology results

### 9.1.1 Balancing of conventional NAND-gate

| | Width | Width | Delay | Delay | |
|---|---|---|---|---|---|
| | $pMOS$ | $nMOS$ | rising | falling | PDP |
| **Conventional** | | | | | |
| Both inputs at $Vdd/2$ | | | | | |
| Toggle both | $300nm$ | $2.08um$ | $1.73nS$ | $1.54nS$ | $86.34aJ$ |
| Toggle one | $300nm$ | $2.08um$ | $2.25nS$ | $798.9pS$ | $80.7aJ$ |
| | | | | | |
| One input at $Vdd/2$ | | | | | |
| Toggle both | $290nm$ | $300nm$ | $700pS$ | $1.82nS$ | $38.8aJ$ |
| Toggle one | $290nm$ | $300nm$ | $969.4pS$ | $1.14nS$ | $20.2aJ$ |

Table 9.1: Balancing of conventional NAND-gate

### 9.1.2 Balancing of NAND BA-gate

| | Width<br>$pMOS$ | Width<br>$nMOS$ | Delay<br>rising | Delay<br>falling | PDP |
|---|---|---|---|---|---|
| **BA-gate**<br>Both inputs at $Vdd/2$ | | | | | |
| Toggle both | $339nm$ | $300nm$ | $2.35nS$ | $2.06nS$ | $55.6aJ$ |
| Toggle one | $339nm$ | $300nm$ | $2.78nS$ | $1.04nS$ | $47.1aJ$ |
| One input at $Vdd/2$ | | | | | |
| Toggle both | $2.87um$ | $300nm$ | $2.73nS$ | $6.24nS$ | $292aJ$ |
| Toggle one | $2.87um$ | $300nm$ | $3.02nS$ | $3.74nS$ | $143aJ$ |

Table 9.2: Balancing of NAND BA-gate

## 9.2 Monte Carlo simulation on NAND-gates

### 9.2.1 4T NAND-gate

| **4T** | -40°C | 20°C | 80°C |
|---|---|---|---|
| Toggle Both | | | |
| Delay Rising: | 2.61$nS$ | 783$pS$ | 322$pS$ |
| std: | 512$pS$ | 141$pS$ | 65.1$pS$ |
| Delay Falling: | 5.67$nS$ | 1.94$nS$ | 990$pS$ |
| std: | 1.60$nS$ | 411$pS$ | 171$pS$ |
| Power: | 6.7$nW$ | 20.7$nW$ | 44.58$nW$ |
| std: | 965$pW$ | 2.02$nW$ | 3.58$nW$ |
| PDP: | 36.86$aJ$ | 39.5$aJ$ | 43.68$aJ$ |
| std: | 7.67$aJ$ | 5.99$aJ$ | 5.27$aJ$ |
| Leakage: | 192$pW$ | 631$pW$ | 3.44$nW$ |
| std: | 7.46$pW$ | 113$pW$ | 652$pW$ |

Table 9.3: Monte Carlo results - 4T NAND

### 9.2.2 AB-structure NAND-gate

| **AB-gate** | -40°C | 20°C | 80°C |
|---|---|---|---|
| Toggle Both | | | |
| Delay Rising: | 7.14$nS$ | 2.40$nS$ | 1.18$nS$ |
| std: | 973$pS$ | 252$pS$ | 103$pS$ |
| Delay Falling: | 6.23$nS$ | 2.10$nS$ | 1.03$nS$ |
| std: | 1.01$nS$ | 273$pS$ | 117$pS$ |
| Power: | 7.51$nW$ | 23.58$nW$ | 50.4$nW$ |
| std: | 582$pW$ | 1.59$nW$ | 2.50$nW$ |
| PDP: | 54.14$aJ$ | 56.39$aJ$ | 59.8$aJ$ |
| std: | 7.88$aJ$ | 5.29$aJ$ | 4.52$aJ$ |
| Leakage: | 207$pW$ | 402$pW$ | 1.64$nW$ |
| std: | 3.01$pW$ | 28.9$pW$ | 183$pW$ |

Table 9.4: Monte Carlo results - 8T NAND

## 9.3 Monte Carlo Results Plotted for 20°C

### 9.3.1 Delay

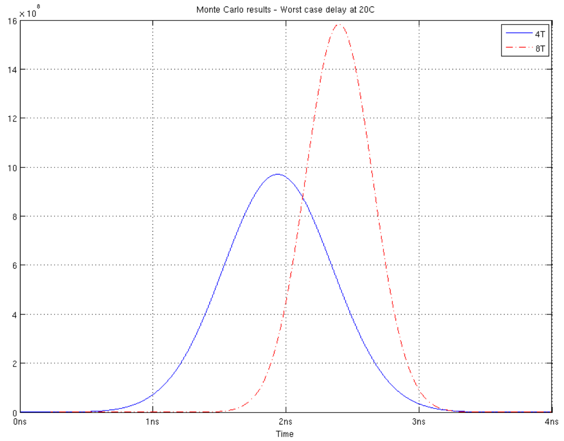


Figure 9.1: NAND-gates worst case Delay-plot at 20°C

### 9.3.2 Power

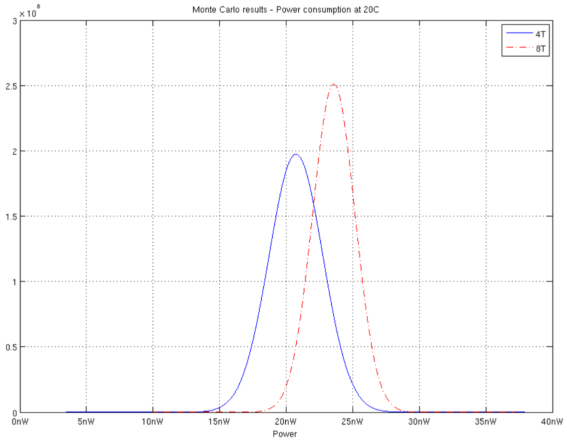


Figure 9.2: Power-plot NAND-gates at 20°C

### 9.3.3 Power-Delay Product (PDP)

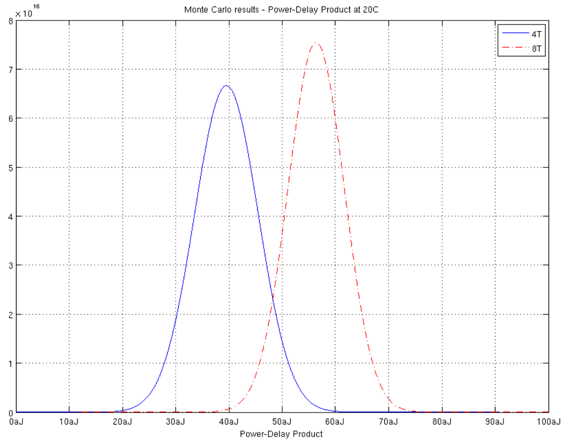


Figure 9.3: Power-plot NAND-gates at 20°C

### 9.3.4 Leakage

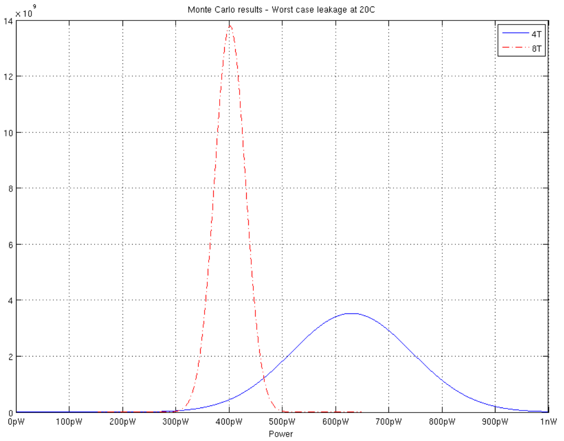


Figure 9.4: Leakage-plot NAND-gates at 20°C

## 9.4 Longest Path

### 9.4.1 Supply voltages and transistor sizes

Transistorsizes for each $V_{DD}$ is found by setting all inputs to $V_{DD}/2$ and adjust to output is $V_{DD}/2$. Transistor *hvtlp*. Simulations done at temperature -40°C.

| Transistor width | $400mV$ | $450mV$ | $475mV$ | $500mV$ |
|---|---|---|---|---|
| pMOS | $270nm$ | $270nm$ | $270nm$ | $270nm$ |
| nMOS | $830nm$ | $735nm$ | $725nm$ | $605nm$ |
| | | | | |
| Delay | | | | |
| Carry Propagation | $216.1\mu S$ | $37.27\mu S$ | $16.24\mu S$ | $6.75\mu S$ |
| Match | $100.7\mu S$ | $17.11\mu S$ | $7.38\mu S$ | $3.06\mu S$ |

Table 9.5: Delay and transistor sizes at different supply voltages

### 9.4.2 Monte Carlo simulation of Longest path

Monte Carlo results from 100 runs. Simulations was performed on Longest paths schematics. Transistor *hvtlp*. Simulations done at temperature -40°C. $V_{DD} = 500mV$.

| **Scematics** | Max | Mean | Sigma |
|---|---|---|---|
| Delay Carry Propagation. | $9.14\mu S$ | $7.63\mu S$ | $410nS$ |
| Delay Match | $3.90\mu S$ | $3.33\mu S$ | $211nS$ |

Table 9.6: Monte Carlo results - Delay Longest paths

## 9.5 Fanout Results

The delay for different fanout simulated at 20°C. It was the falling-edge delay that was worst. This delay is added for each level as the tree-height increases. When the values do not change for more nodes, the values are not filled into the table. The same numbers are plotted in Figure 9.5.

| Fanout: | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| Nodes: | | | | | | |
| 2 | $66.7nS$ | | | | | |
| 3 | $133.4nS$ | $85.5nS$ | | | | |
| 4 | – | $173nS$ | $106nS$ | | | |
| 5 | $201nS$ | – | $212nS$ | $125.3nS$ | | |
| 6 | – | – | – | $250.6nS$ | $144.4nS$ | |
| 7 | – | – | – | – | $288.8nS$ | $163.3nS$ |
| 8 | – | – | – | – | – | $326.6nS$ |
| 9 | $266.8nS$ | – | – | – | – | – |
| 10 | – | $259.5nS$ | – | – | – | – |
| 17 | $333.5nS$ | – | $318nS$ | – | – | – |
| 24 | $333.5nS$ | $259.5nS$ | $318nS$ | $250.6nS$ | $288.8nS$ | $326.6nS$ |

Table 9.7: Fanout delay

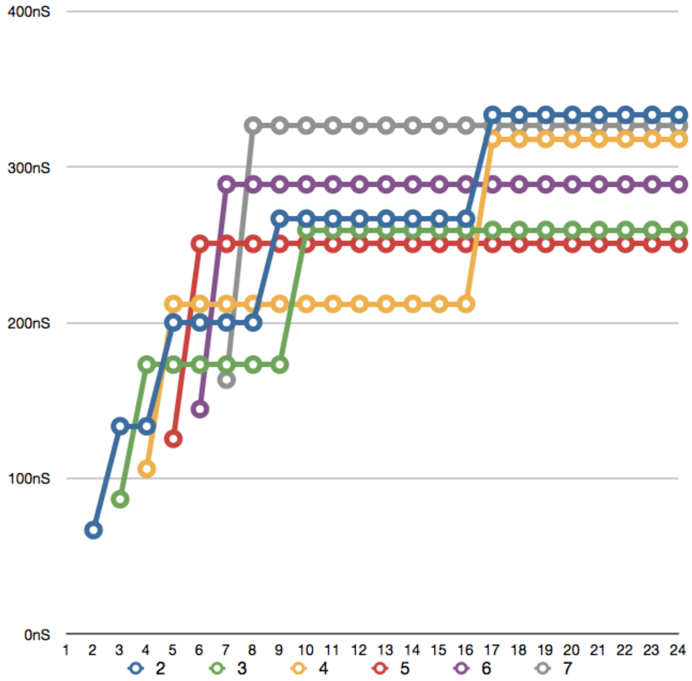


Figure 9.5: Delay for different fanout

## 9.6 RTC Simulations

### 9.6.1 Monte Carlo results Delay on RTC-schematics

Monte Carlo simulations performed on schematics.
Setup: 100 runs, $V_{DD} = 500mV$ at -40°C.

| **Scematics** | Max | Mean | Sigma |
|---|---|---|---|
| Delay Carry Propagation. | $13.1\mu S$ | $10.82\mu S$ | $678.3nS$ |
| Delay Match | $6.16\mu S$ | $5.25\mu S$ | $328.8nS$ |

Table 9.8: Monte Carlo results - Delay on RTC-schematics

### 9.6.2 Monte Carlo results Delay on RTC-Layout

Monte Carlo simulations performed on layout with 2D extraction.
Setup: 100 runs, $V_{DD} = 500mV$ at -40°C.

| **Scematics** | Max | Mean | Sigma |
|---|---|---|---|
| Delay Carry Prop. | $29.59\mu S$ | $26.13\mu S$ | $1.42\mu S$ |
| Delay Match | $14.86\mu S$ | $12.63\mu S$ | $739.9\mu S$ |

Table 9.9: Monte Carlo results - Delay on RTC-layout

### 9.6.3 Monte Carlo results Power on RTC-Layout

Monte Carlo simulations performed on layout with 2D extraction.
Setup: 10 runs, $V_{DD} = 500mV$ and simulating over 2mS

| | -40°C | 20°C | 80°C |
|---|---|---|---|
| Max: | $3.54nW$ | $3.74nW$ | $4.80nW$ |
| Mean: | $3.62nW$ | $3.81nW$ | $4.88nW$ |
| std: | $49.8pW$ | $44.6pW$ | $65.6pW$ |

Table 9.10: Monte Carlo results - Power on RTC-Layout

### 9.6.4 Delay and Power consumption for various voltages

For each category, first a simulation result from a single run on schematic with $V_{DD} = 500mV$ are presented. Then tables with simulations for various voltages from runs on layout with 3D extraction of parasitics are presented. For the layout, the supply voltages varies from $V_{DD} = 500mV$ to $V_{DD} = 600mv$ with $10mV$ interval.

#### 9.6.4.1 Delay Carry Propogation

| **Schematic** | -40°C | 20°C | 80°C |
|---|---|---|---|
| $V_{DD} = 500mV$ | $9.63\mu S$ | $1.84\mu S$ | $623nS$ |

Table 9.11: Delay Carry Prop. on schematics

| **Layout**<br>$V_{DD}$ | -40°C | 20°C | 80°C |
|---|---|---|---|
| $500mV$ | $29.78\mu S$ | $5.54\mu S$ | $1.87\mu S$ |
| $510mV$ | $21.92\mu S$ | $4.46\mu S$ | $1.59\mu S$ |
| $520mV$ | $16.27\mu S$ | $3.62\mu S$ | $1.37\mu S$ |
| $530mV$ | $12.16\mu S$ | $2.95\mu S$ | $1.18\mu S$ |
| $540mV$ | $9.18\mu S$ | $2.43\mu S$ | $1.02\mu S$ |
| $550mV$ | $6.99\mu S$ | $2.02\mu S$ | $881nS$ |
| $560mV$ | $5.38\mu S$ | $1.69\mu S$ | $767nS$ |
| $570mV$ | $4.19\mu S$ | $1.42\mu S$ | $672nS$ |
| $580mV$ | $3.30\mu S$ | $1.20\mu S$ | $591nS$ |
| $590mV$ | $2.62\mu S$ | $1.02\mu S$ | $522nS$ |
| $600mV$ | $2.12\mu S$ | $870nS$ | $463nS$ |

Table 9.12: Delay Carry Prop. versus $V_{DD}$ - On layout

#### 9.6.4.2 Delay Match

| **Schematic** | -40°C | 20°C | 80°C |
|---|---|---|---|
| $V_{DD} = 500mV$ | $4.57\mu S$ | $858nS$ | $289nS$ |

Table 9.13: Delay Match on schematics

| **Layout**<br>$V_{DD}$ | -40°C | 20°C | 80°C |
|---|---|---|---|
| $500mV$ | $13.95\mu S$ | $2.56\mu S$ | $850nS$ |
| $510mV$ | $10.27\mu S$ | $2.06\mu S$ | $723nS$ |
| $520mV$ | $7.58\mu S$ | $1.67\mu S$ | $618nS$ |
| $530mV$ | $5.68\mu S$ | $1.36\mu S$ | $531nS$ |
| $540mV$ | $4.26\mu S$ | $1.12\mu S$ | $458nS$ |
| $550mV$ | $3.26\mu S$ | $923nS$ | $398nS$ |
| $560mV$ | $2.51\mu S$ | $768nS$ | $346nS$ |
| $570mV$ | $1.95\mu S$ | $643nS$ | $305nS$ |
| $580mV$ | $1.53\mu S$ | $542nS$ | $267nS$ |
| $590mV$ | $1.21\mu S$ | $460nS$ | $236nS$ |
| $600mV$ | $979nS$ | $393nS$ | $209nS$ |

Table 9.14: Delay Match versus $V_{DD}$ - On layout

#### 9.6.4.3 Power Consumption

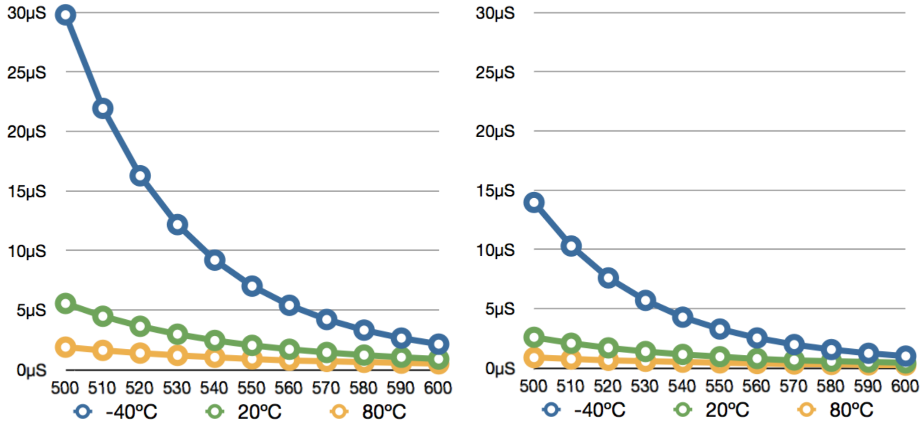
| **Schematic** | -40°C | 20°C | 80°C |
|---|---|---|---|
| $V_{DD} = 500mV$ | $1.84nW$ | $2.00nW$ | $3.13nW$ |

Table 9.15: Power consumption on schematics

| **Layout**<br>$V_{DD}$ | -40°C | 20°C | 80°C |
|---|---|---|---|
| $500mV$ | $4.87nW$ | $5.16nW$ | $6.14nW$ |
| $510mV$ | $5.08nW$ | $5.23nW$ | $6.38nW$ |
| $520mV$ | $5.25nW$ | $5.45nW$ | $6.62nW$ |
| $530mV$ | $5.46nW$ | $5.66nW$ | $6.87nW$ |
| $540mV$ | $5.68nW$ | $5.88nW$ | $7.13nW$ |
| $550mV$ | $5.90nW$ | $6.11nW$ | $7.38nW$ |
| $560mV$ | $6.13nW$ | $6.35nW$ | $7.65nW$ |
| $570mV$ | $6.36nW$ | $6.58nW$ | $7.92nW$ |
| $580mV$ | $6.59nW$ | $6.83nW$ | $8.19nW$ |
| $590mV$ | $6.83nW$ | $7.08nW$ | $8.47nW$ |
| $600mV$ | $7.20nW$ | $7.36nW$ | $8.76nW$ |

Table 9.16: Power consumption versus $V_{DD}$ - On layout

#### 9.6.4.4 Delay versus $V_{DD}$ Plot



(a) Delay Carry Propagation

(b) Delay Match

Figure 9.6: Delay versus V$_{DD}$ - on layout

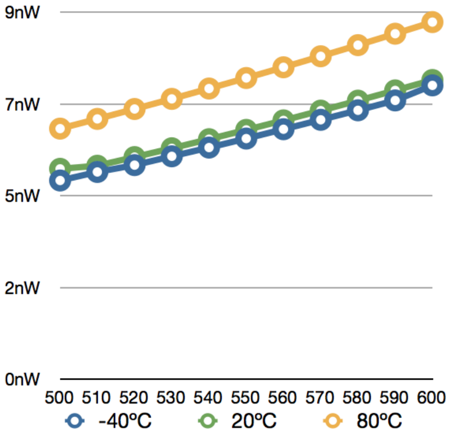


Figure 9.7: Power consumption versus $V_{DD}$

# Chapter 10

# Discussion

## 10.1 Results

The main focus of this thesis has been to design a low-power Real-Time Clock in subthreshold.

As seen from Table 9.6.1 and 9.6.2, the delay increase from the schematic to layout with a factor of about 2.2. These measurements are done with Monte Carlo on layout with 2D extraction of parasitics. If we look at the numbers from Table 9.11 and 9.12, the delay increase with a factor 3 from a single run on schematic compared with layout with 3D extraction of parasitic. The Monte Carlo on 2D extraction seem optimistic when comparing to a single run with 3D extraction. The simulation with 3D extraction shows that it operates on the lowest possible supply voltages for reaching deadlines in -40°C. This will not hold for all process and mismatch variations. To ensure proper behavior at -40°C, the $V_{DD}$ could be increased with $10mV$. This reduces the carry propagation delay at -40°C with 26% down to $21.92\mu S$ while power increase with 4% to $5.08nW$.

The results show that this is a very power efficient design. The exact power consumption for the RTC in EFM32G series microcontroller is not known, but it has been indicated that it is around $400nW$. This design could reduces the assumed power consumption down to $1.5 - 2.5\%$.

## 10.2 BA-structure

The BA-structure is probably this thesis most substantial contribution into the ultra low voltage / low-power design discipline. As the results show, the robustness to mismatch and process-variations are substantially increased. While the speed is slower and PDP marginally increased, the relative deviation ($\sigma$) is almost half for a NAND in BA-structure compared to a conventional NAND. The increased robustness makes it possible to operate less pessimistic while maintain yield from production, and thereby save power. The standard deviation for $V_T$ in subthreshold is usually given as [15] :

$$\sigma(V_T) = \frac{K}{\sqrt{WL}} \tag{10.1}$$

where $K$ is a constant. In order to double the robustness, the area usually grows with a factor 4:

$$\frac{\sigma(V_T)}{2} = \frac{K}{\sqrt{4WL}} \tag{10.2}$$

In BA-structure, the transistor count doubles for NAND, NOR and INVERTER, but are the same for XOR and XNOR. This means that the area could increase by a factor 1 - 2 depending on the circuit.

Possible benefits of BA-structure include, but is not limited to:

- Since the stack height is constant throughout a design, equal transistor dimensions can be used, to a much larger extent than usually, for the entire integrated circuit. This simplifies cell libraries for integrated circuit design, and should reduce development costs.
- Enhanced robustness regarding Process, Voltage and Temperature variations (PVT-variations).
- Since all dimensions are equal, simpler regulator mechanisms can be used to adjust well-biasing, adjusting the design to compensate for PVT-variation.
- Well-biasing can be used to adjust the transistor threshold-voltages, as seen from driving nodes, to enhance performance and functionality.
- Since all slices are equal, programmable wiring can be used to implement FPGAs, for example in conventional CMOS as well as other technologies, like for example FinFET and other multigate transistor technologies.
- The regularity provided is expected to improve manufacturability and yield for integrated circuits, both traditional 2D and 3D.

# Chapter 11

# Concluding Remarks

Subthreshold operation has proven to be an effective method to implement low power design. In this thesis, a subthreshold Real-Time Counter has been designed, implemented in layout and tested. This is all done in $65nm$ technology provided by STMicroelectronics and performed in Cadence Virtuoso.

The RTC is designed for $500mV$. To ensure proper behavior over all mismatch and process variation at -40°C, the supply voltage could be increased to about 510mV. This supply voltage has a power consumption of $6.4nW$. Therefor, this design could reduce the assumed power consumption down to $1.5 - 2.5\%$ and is well suited for low power microcontrollers.

A new design methodology is proposed. This gives increased robustness regarding Process, Voltage and Temperature (PVT) variations. It also simplifies cell library and make layout more regular and uniform.

## 11.1 Future work

The RTC should be taped out, to be physically tested on chip. If the design is to be adopted on a microcontroller, a new layout with other parameters has to be made in the used technology.

The new design methodology should be further developed, and cell library needs to be developed. Then known circuits should be implemented and compared to conventional implementations. Work is in progress for publication of results.

# Appendix A

# VHDL-verification of 4-bits

## A.1 Blocks

### A.1.1 Flip-Flop

Listing A.1: D-Flip-Flop

```
library IEEE;
use ieee.std_logic_1164.all;
use work.all;

entity DVIPPE is
    port(
    D         :in      std_logic;
    Reset     :in      std_logic;
    Clk       :in      std_logic;
    Q         :out     std_logic
    );
end entity DVIPPE;

Architecture Behavior of DVIPPE is
begin
    process(Clk, Reset)
    begin
        if (reset = '1') then
            Q <= '0';

            elsif rising_edge(Clk) then
                Q <= D;
            end if;
        end process;
end Architecture;
```

### A.1.2 First Bit

Listing A.2: Bit without an adder

```
library IEEE;
use ieee.std_logic_1164.all;
use work.all;

entity FirstBIT is
    port(
    Carry    :out    std_logic;
    Qout     :out    std_logic;
    Clk      :in     std_logic;
    Reset    :in     std_logic
    );
end entity FirstBIT;

Architecture Behavior of FirstBIT is
    component DVIPPE is
        port(
            D        :in     std_logic;
             Reset   :in     std_logic;
             Clk     :in     std_logic;
            Q        :out    std_logic
             );
    end component DVIPPE;

    signal D     :std_logic;
    signal Q     :std_logic;

begin
    VIPPE:DVIPPE
    port map(D, Reset, Clk, Q);

    D        <= not Q;
    Carry    <= Q;
    Qout     <= Q;
end architecture;
```

### A.1.3 Bit with adder

Listing A.3: Bit with adder

```
library IEEE;
use ieee.std_logic_1164.all;
use work.all;

entity BIT is
    port(
    Input   :in     std_logic;
    Carry   :out    std_logic;
    Qout    :out    std_logic;
    Clk     :in     std_logic;
    Reset   :in     std_logic
    );
end entity BIT;

Architecture Behavior of BIT is
    component DVIPPE is
        port(
            D       :in     std_logic;
            Reset   :in     std_logic;
            Clk     :in     std_logic;
            Q       :out    std_logic
            );
    end component DVIPPE;

    signal D    :std_logic;
    signal Q    :std_logic;

begin
    VIPPE:DVIPPE
    port map(D, Reset, Clk, Q);

    D       <= Input XOR Q;
    Carry   <= Input AND Q;
    Qout    <= Q;
end architecture;
```

## A.2 4-bit counter with compare circuit

Listing A.4: 4-bit counter and compare circuit

```
library IEEE;
use ieee.std_logic_1164.all;
use work.all;

entity BITS is
    port(
    Clk             :in      std_logic;
    Reset           :in      std_logic;
    Counter         :out     std_logic_vector(3 downto 0);
    Comparator      :in      std_logic_vector(3 downto 0);
    Match           :out     std_logic;
    LastCarry       :out     std_logic
    );
end entity;


Architecture Behavior of BITS is
    component BIT is
        port(
            Input    :in      std_logic;
            Carry    :out     std_logic;
            Qout     :out     std_logic;
            Clk      :in      std_logic;
            Reset    :in      std_logic
            );
    end component BIT;

    component FirstBIT is
        port(
            Carry    :out     std_logic;
            Qout     :out     std_logic;
            Clk      :in      std_logic;
            Reset    :in      std_logic
            );
    end component FirstBIT;



    signal  Q             :std_logic_vector(3 downto 0);
    signal  CarryVect     :std_logic_vector(2 downto 0);
    signal  CompXorOut    :std_logic_vector(3 downto 0);
    signal  MatchVect     :std_logic_vector(2 downto 1);

begin
    BIT0:FirstBIT
    port map(CarryVect(0), Q(0), Clk, Reset);

    BIT1:BIT
    port map(CarryVect(0), CarryVect(1), Q(1), Clk, Reset);

    BIT2:BIT
    port map(CarryVect(1), CarryVect(2), Q(2), Clk, Reset);

    BIT3:BIT
    port map(CarryVect(2), LastCarry, Q(3), Clk, Reset);

    CompXorOut(0) <= Q(0) XOR  Comparator(0);
    CompXorOut(1) <= Q(1) XNOR Comparator(1);
    CompXorOut(2) <= Q(2) XOR  Comparator(2);
    CompXorOut(3) <= Q(3) XOR  Comparator(3);

    MatchVect(2) <= CompXorOut(2) NOR  CompXorOut(3);
    MatchVect(1) <= CompXorOut(1) NAND MatchVect(2);
    Match   <= CompXorOut(0) NOR MatchVect(1);
    counter <= Q;

end Architecture;
```

## A.3 Testbench

Listing A.5: Testbench

```
library IEEE;
use ieee.std_logic_1164.all;
use work.all;

entity tb is
end entity;

Architecture behavior of tb is
    component BITS is
        port(
        Clk          :in     std_logic;
        Reset        :in     std_logic;
        Counter      :out    std_logic_vector(3 downto 0);
        Comparator   :in     std_logic_vector(3 downto 0);
        Match        :out    std_logic;
        LastCarry    :out    std_logic
        );
    end component;

    signal  Clock         :std_logic;
    signal  Reset         :std_logic := '1';
    signal  Counter       :std_logic_vector(3 downto 0);
    signal  compReg       :std_logic_vector(3 downto 0) :="
       1001";
    signal  compMatch     :std_logic;
    signal  LastCarry     :std_logic;

    constant clk_half_period: time:= 15 us; --ca 33kHz clk

begin
    ADDER: BITS
    port map(Clock, Reset, Counter, compReg, compMatch,
       LastCarry);

    cl: process
    begin
        Clock<='0';
        wait for clk_half_period;
        Clock<=’1’;
        wait for clk_half_period;
    end process;

    div: process
    begin
        wait for 3*clk_half_period;
        wait until rising_edge(Clock);
        Reset <= ’0’;
    end process;
end architecture;
```

## A.4 Simulation plot

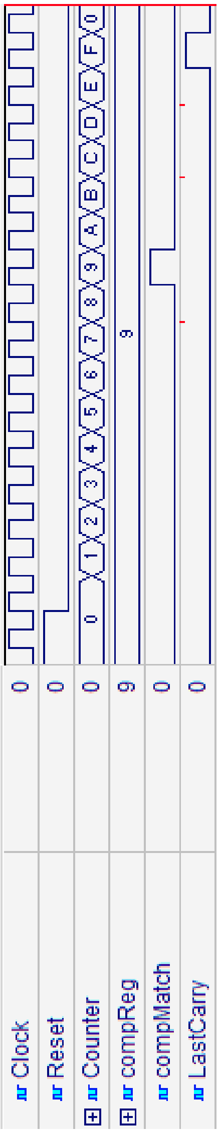


Figure A.1: VHDL simulation of the four first bits.

# Appendix B

# Schematics

## B.1 Inverter

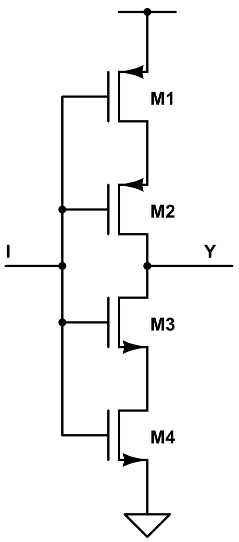


Figure B.1: Inverter

## B.2 Clocked inverter

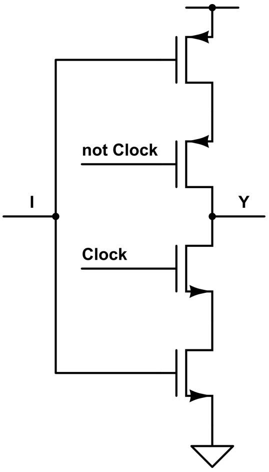


Figure B.2: Clocked Inverter

## B.3 NAND-gate

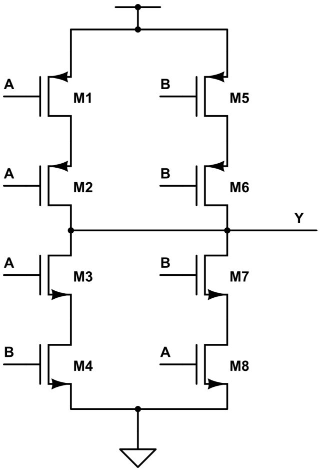


Figure B.3: NAND-gate

## B.4 NOR-gate

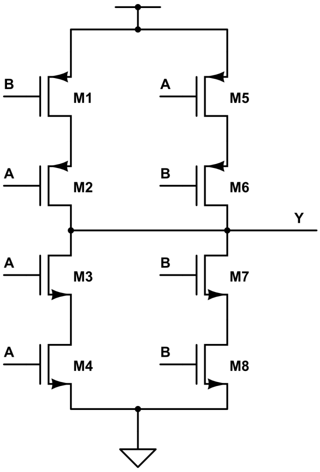


Figure B.4: NOR-gate

## B.5 XOR-gate

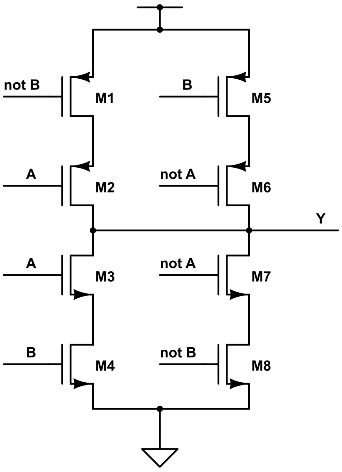


Figure B.5: XOR-gate

## B.6 XNOR-gate

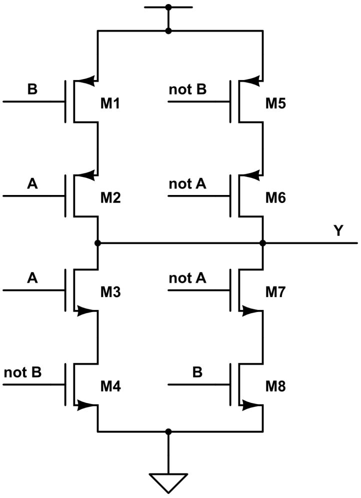


Figure B.6: XNOR-gate

## B.7 Half Adder

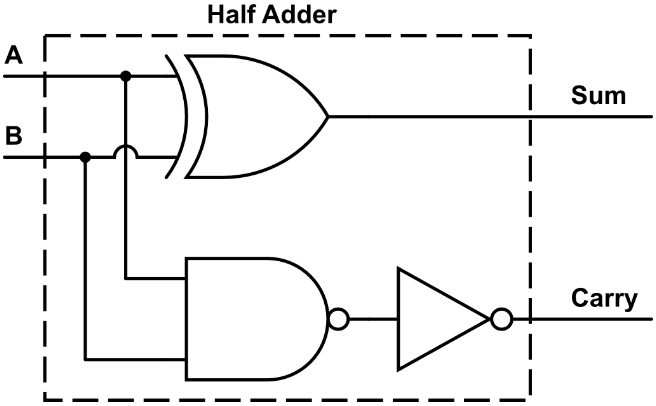


Figure B.7: Half Adder

## B.8 C$^2$MOS Latch

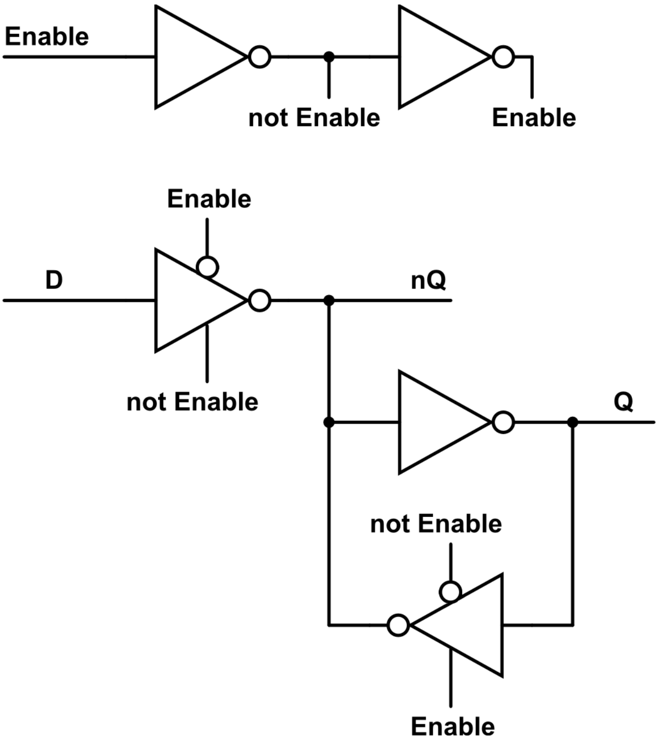


Figure B.8: C$^2$MOS Latch

## B.9 C$^2$MOS D-Flip-Flop

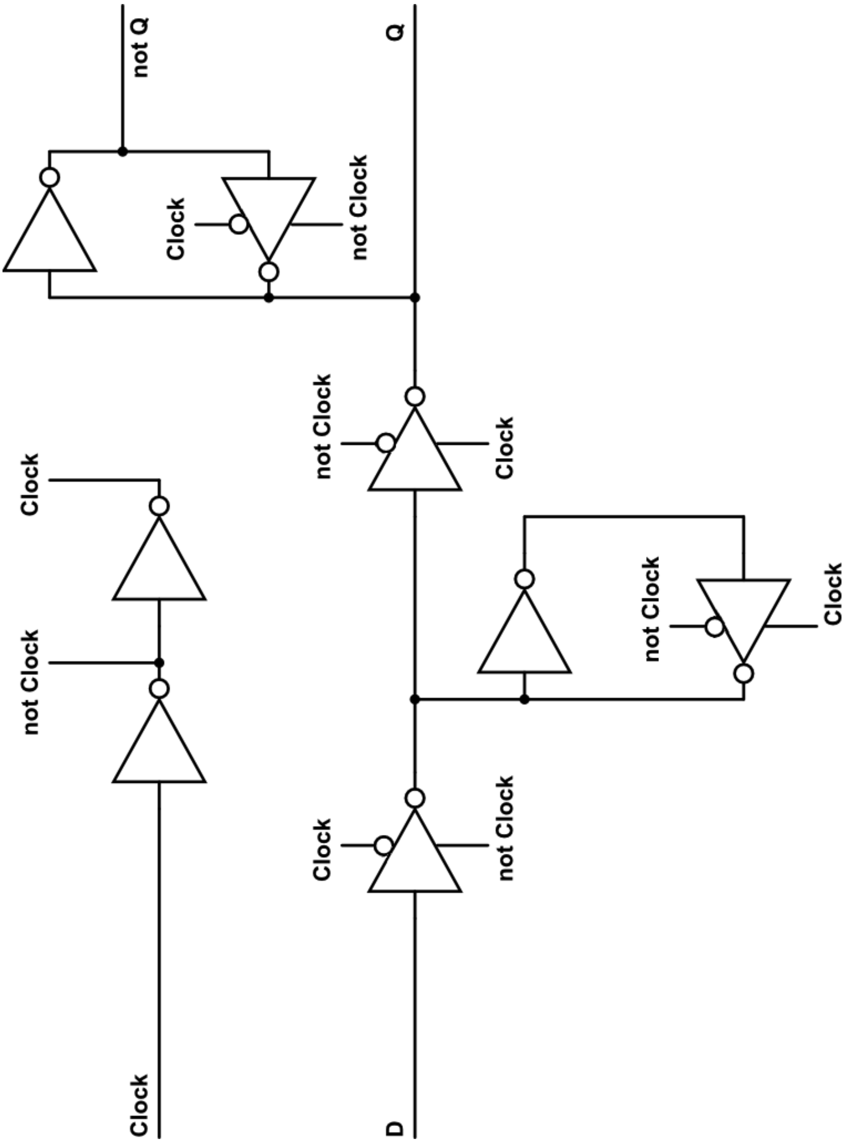


Figure B.9: C$^2$MOS D-Flip Flop

## B.10 C$^2$MOS D-Flip-Flop with asynchronous Reset

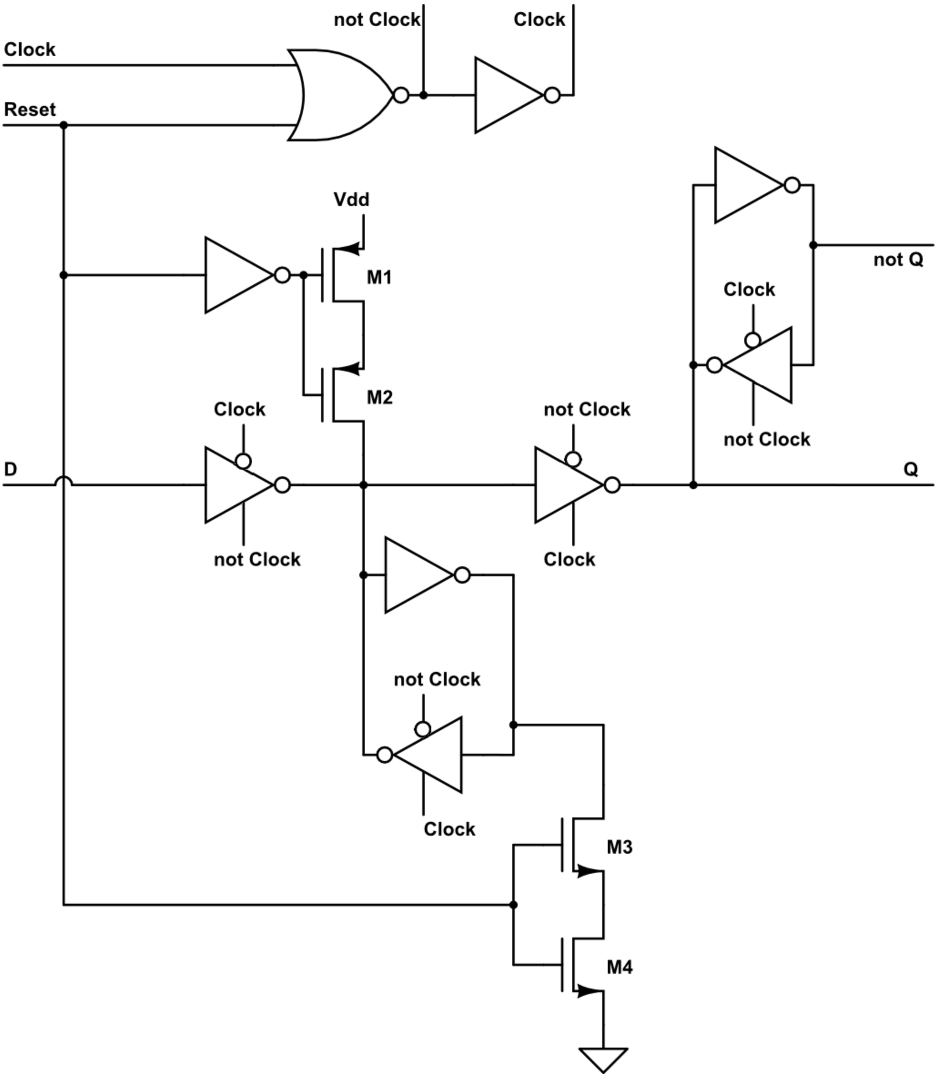


Figure B.10: C$^2$MOS D-Flip Flop with asynchronous Reset

## B.11 Clock-gate

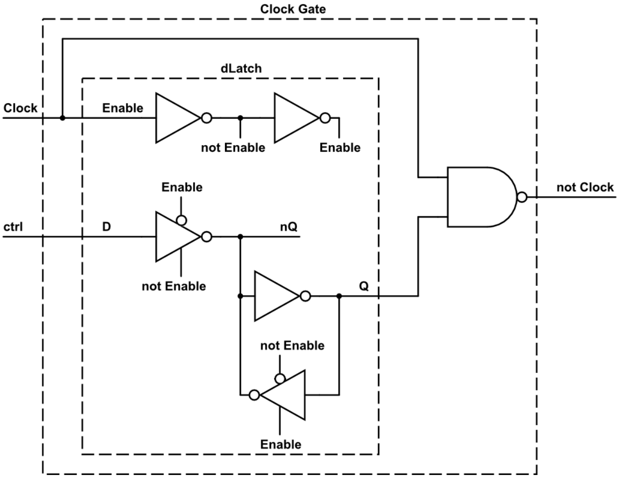


Figure B.11: Clock-gate

## B.12 Compare Bit with Xor

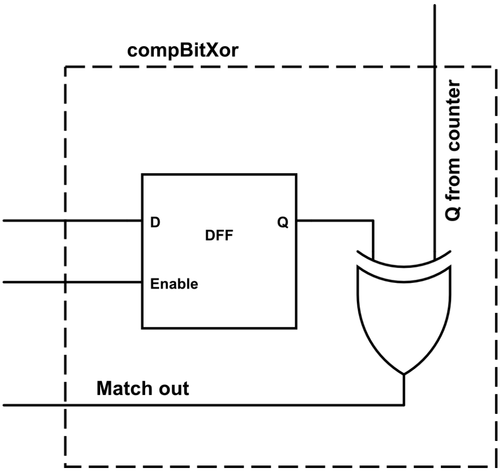


Figure B.12: compBitXor

## B.13 Compare Bit with XOR and NOR

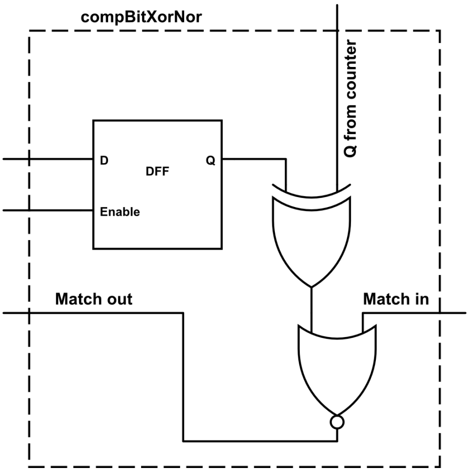


Figure B.13: compBitXorNor

## B.14 Compare Bit with XNOR and NAND

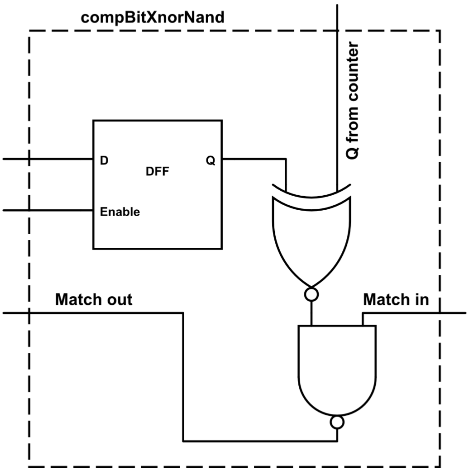


Figure B.14: compBitXnorNand

## B.15 Bit with XOR and NOR

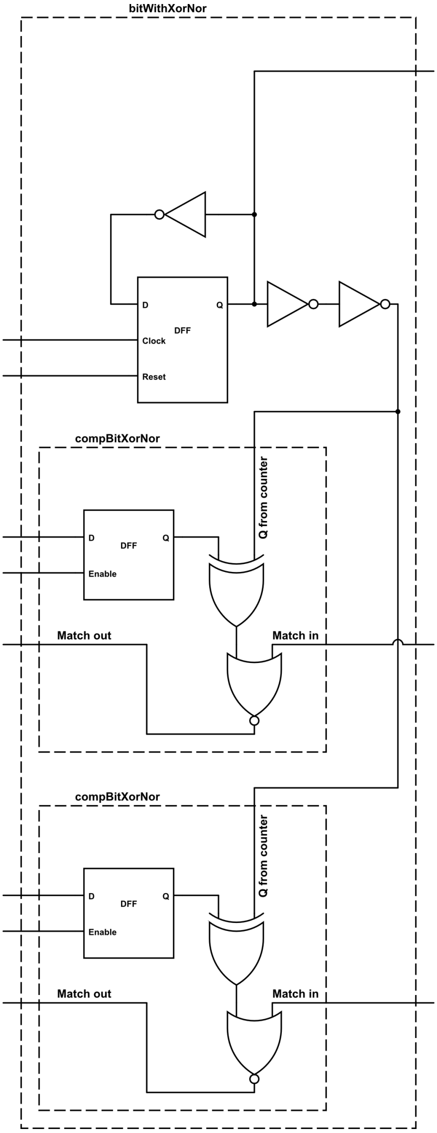


Figure B.15: bitWithXorNor

## B.16 Bit with Adder and XOR

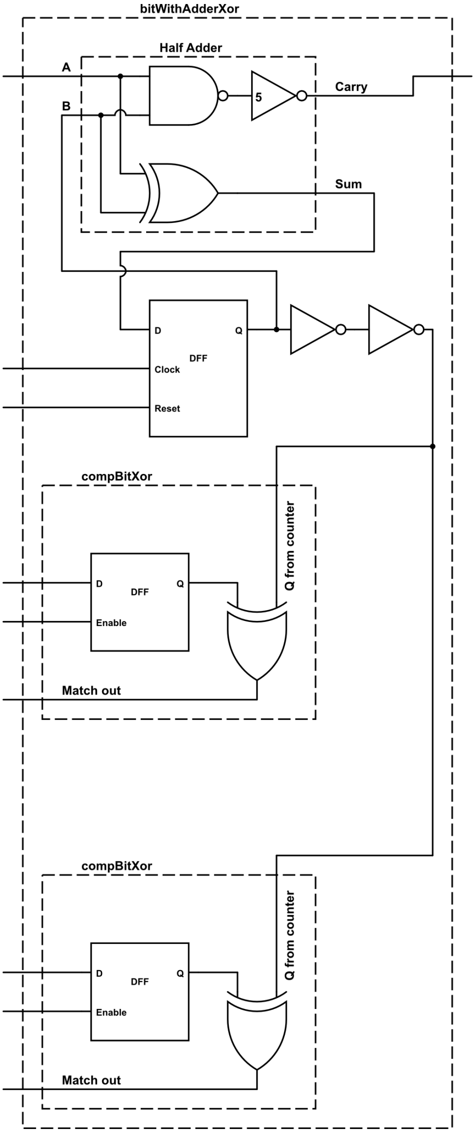


Figure B.16: bitWithAdderXor

## B.17 Bit with Adder, XOR and NOR

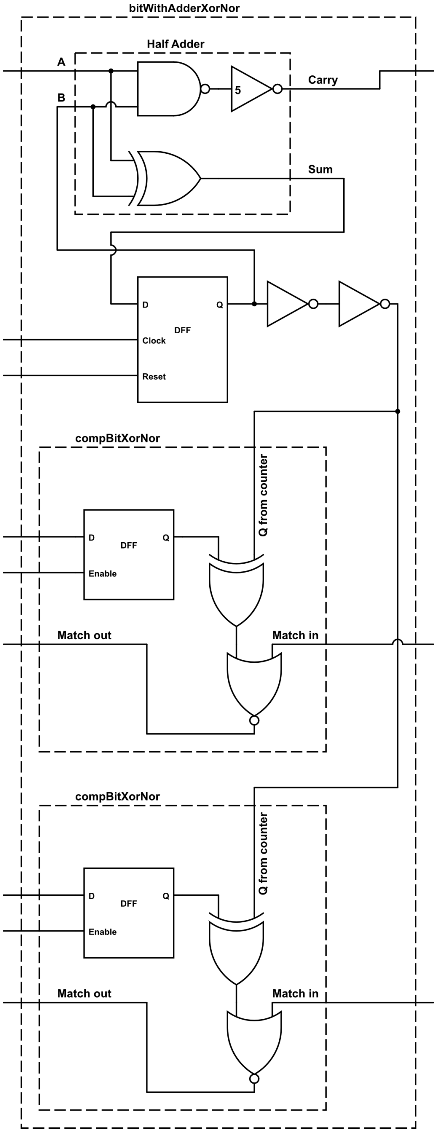


Figure B.17: bitWithAdderXorNor

## B.18 Bit with Adder, XNOR and NAND

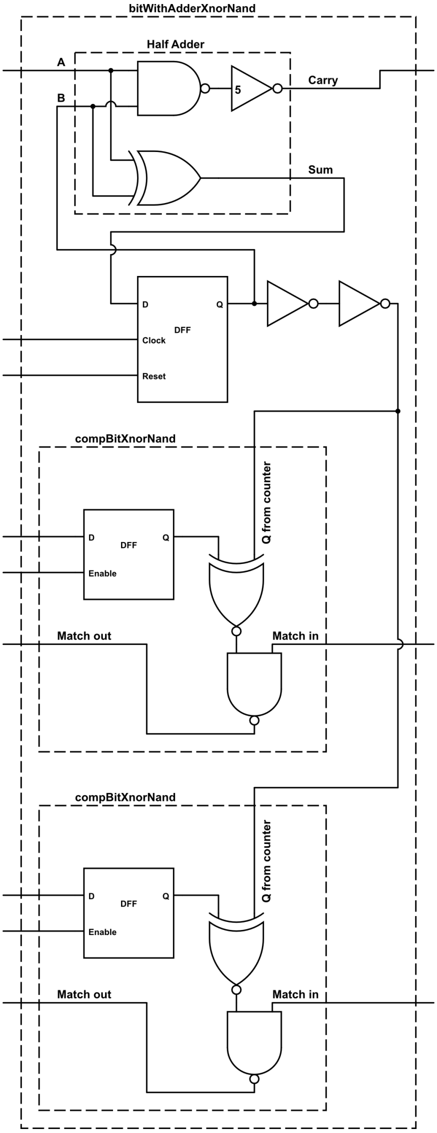


Figure B.18: bitWithAdderXnorNand

## B.19 RTC - whole design

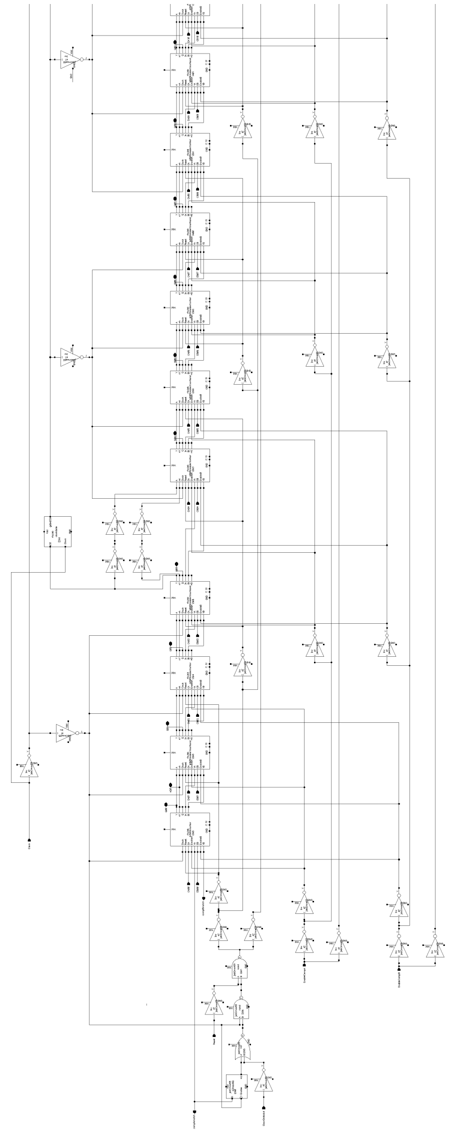


Figure B.19: RTC - without $V_{dd}$ and $GND$

## B.20 RTC - whole design

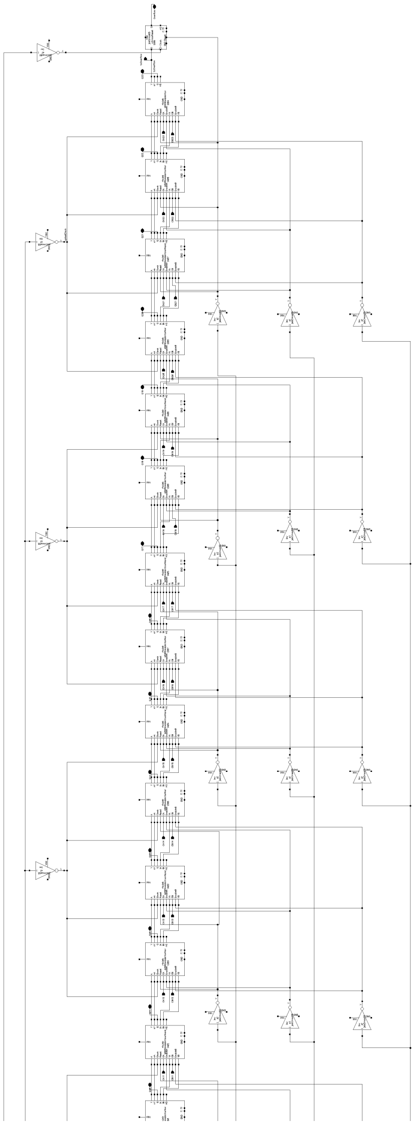


Figure B.20: RTC - without $V_{dd}$ and $GND$

## B.21 RTC structure

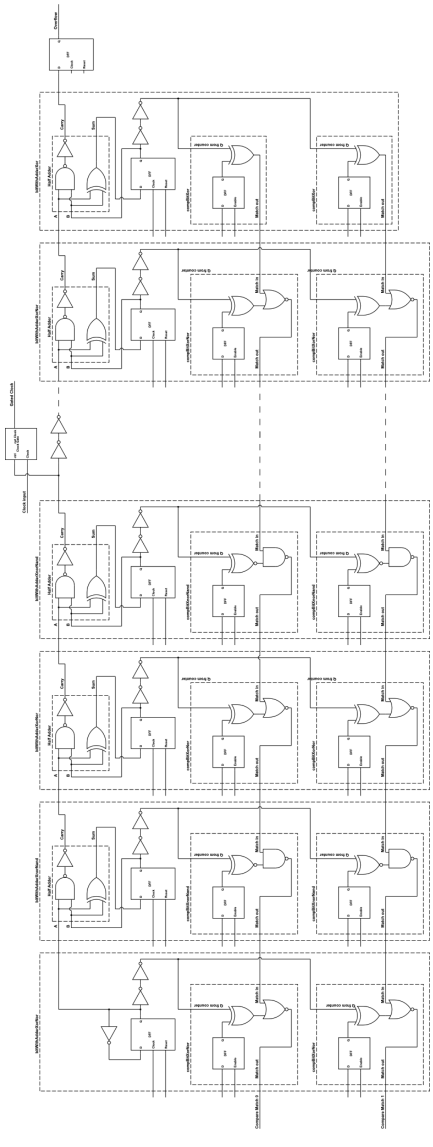


Figure B.21: RTC structure

# Appendix C

# Layout

## C.1 Inverter

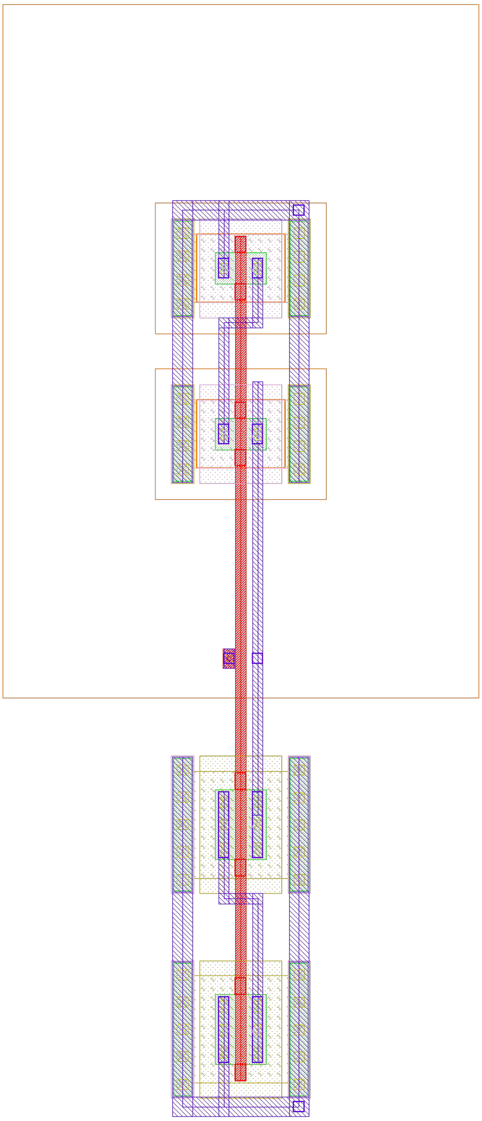


Figure C.1: Inverter

## C.2 Clocked Inverter

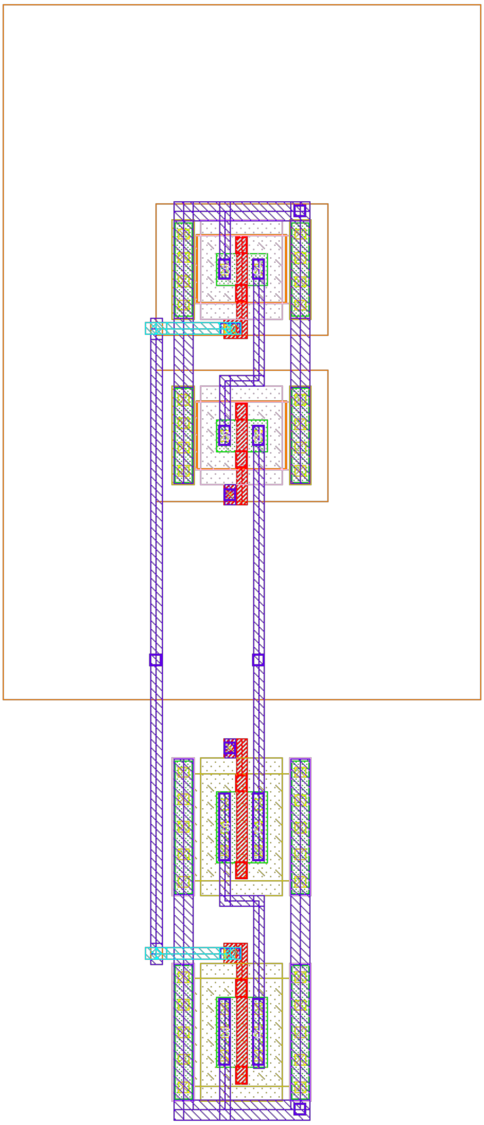


Figure C.2: Clocked Inverter

## C.3 NAND-gate

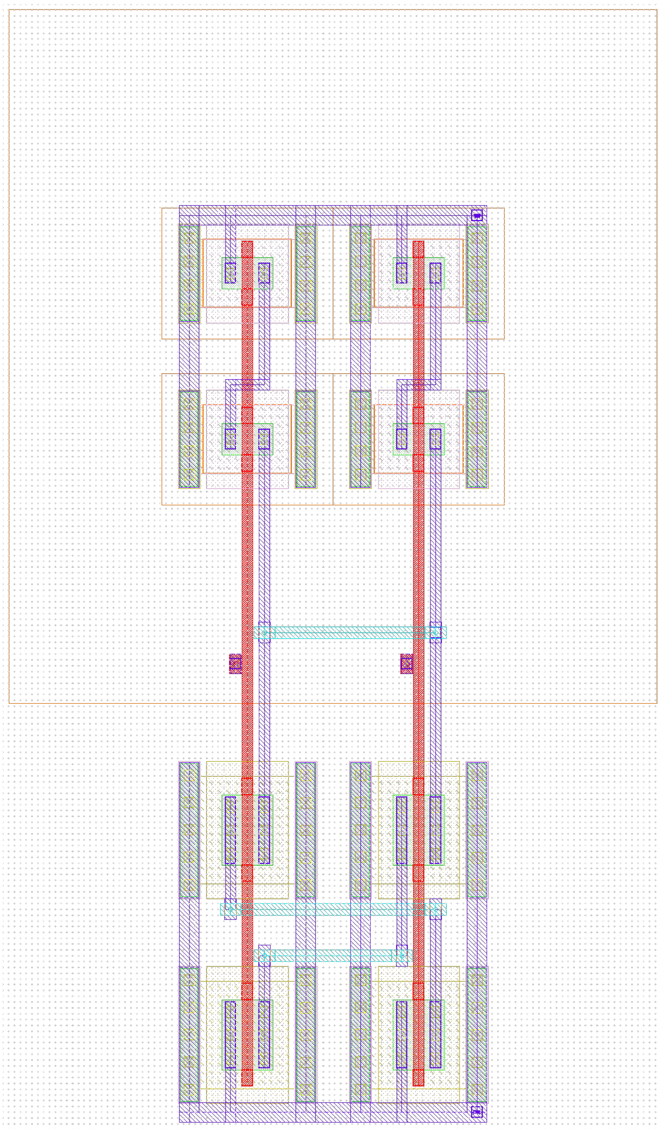


Figure C.3: NAND-gate

## C.4 NOR-gate

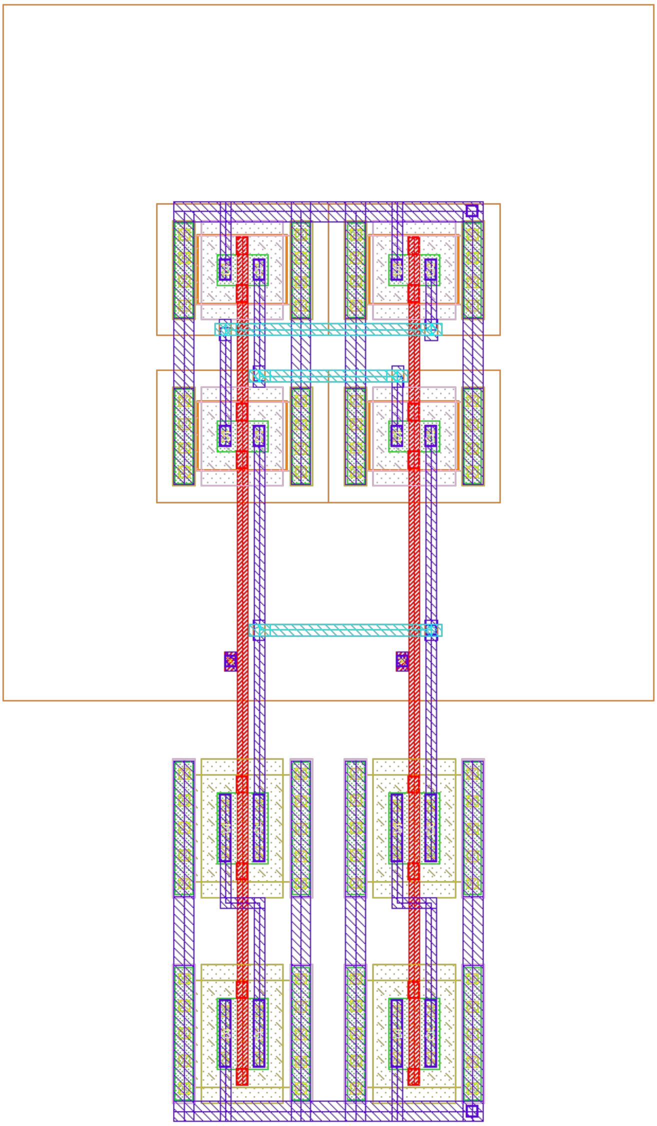


Figure C.4: NOR-gate

## C.5 XOR-gate

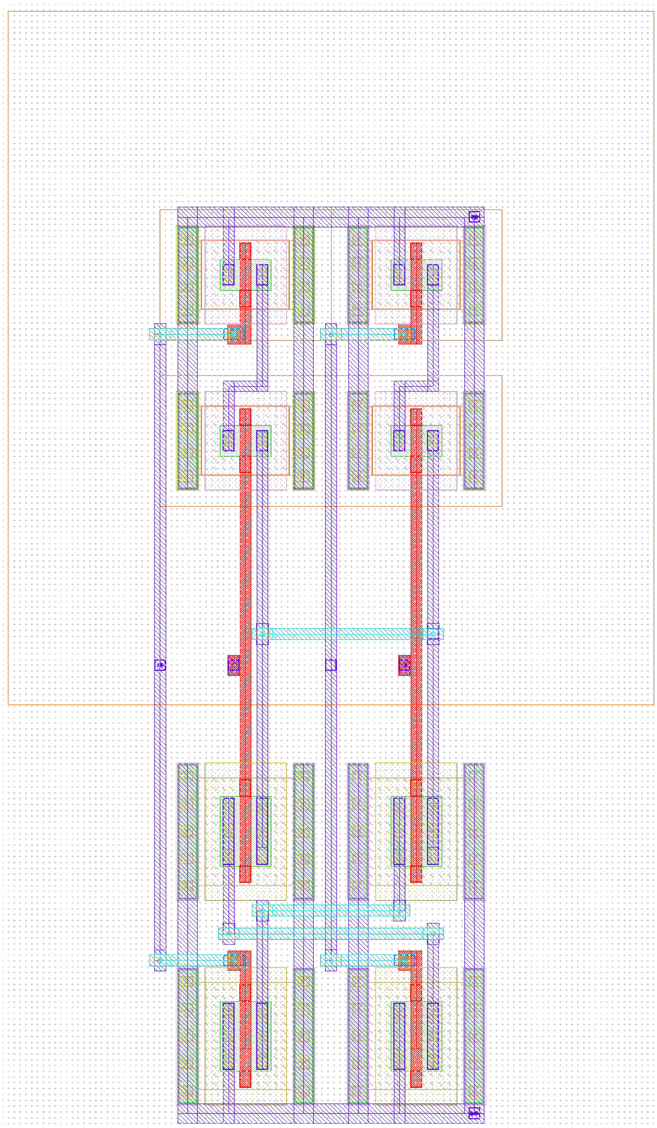


Figure C.5: XOR-gate

## C.6 XNOR-gate

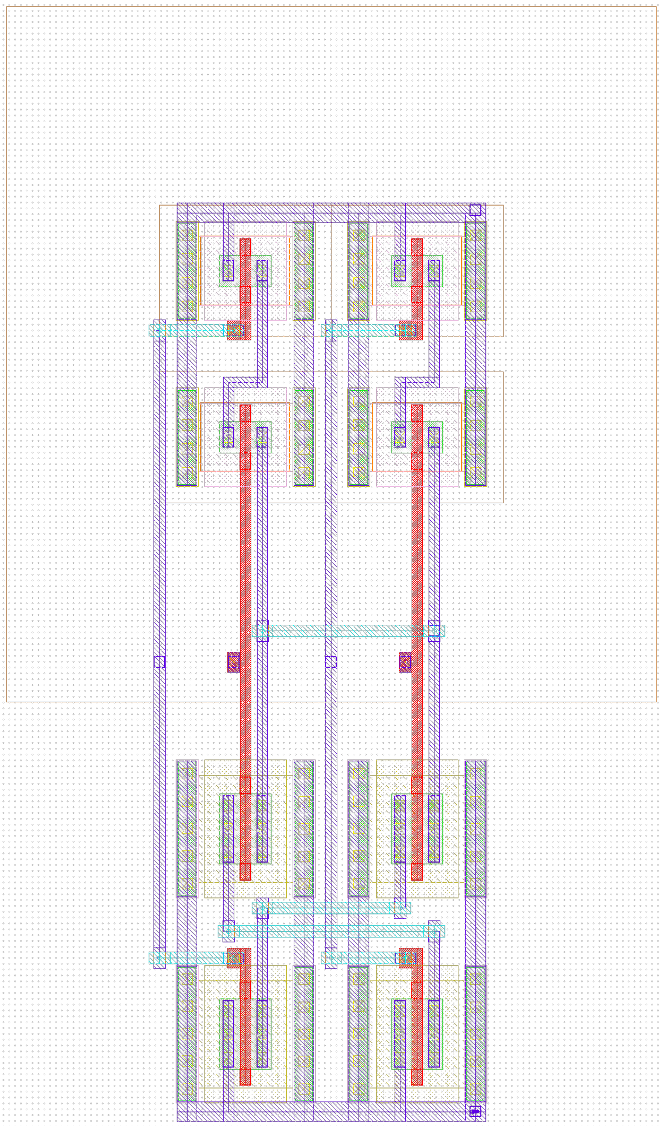


Figure C.6: XNOR-gate

## C.7 Half Adder

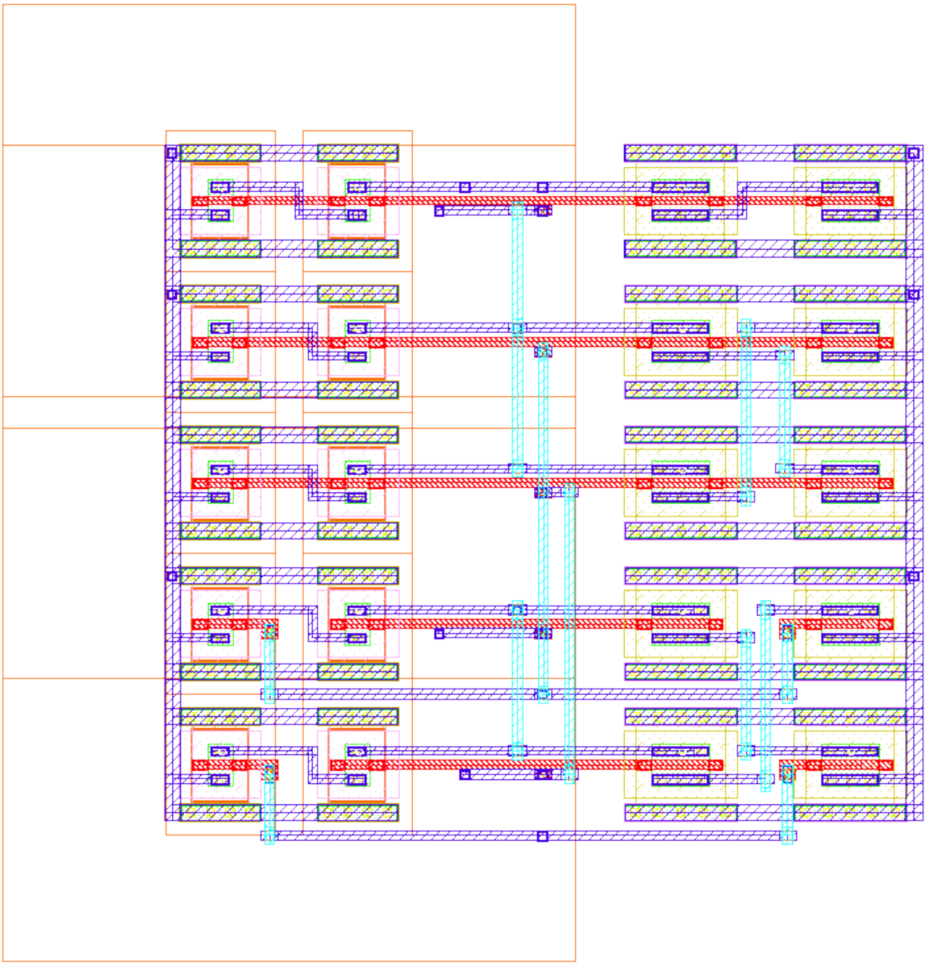


Figure C.7: Half Adder

## C.8 C$^2$MOS Latch

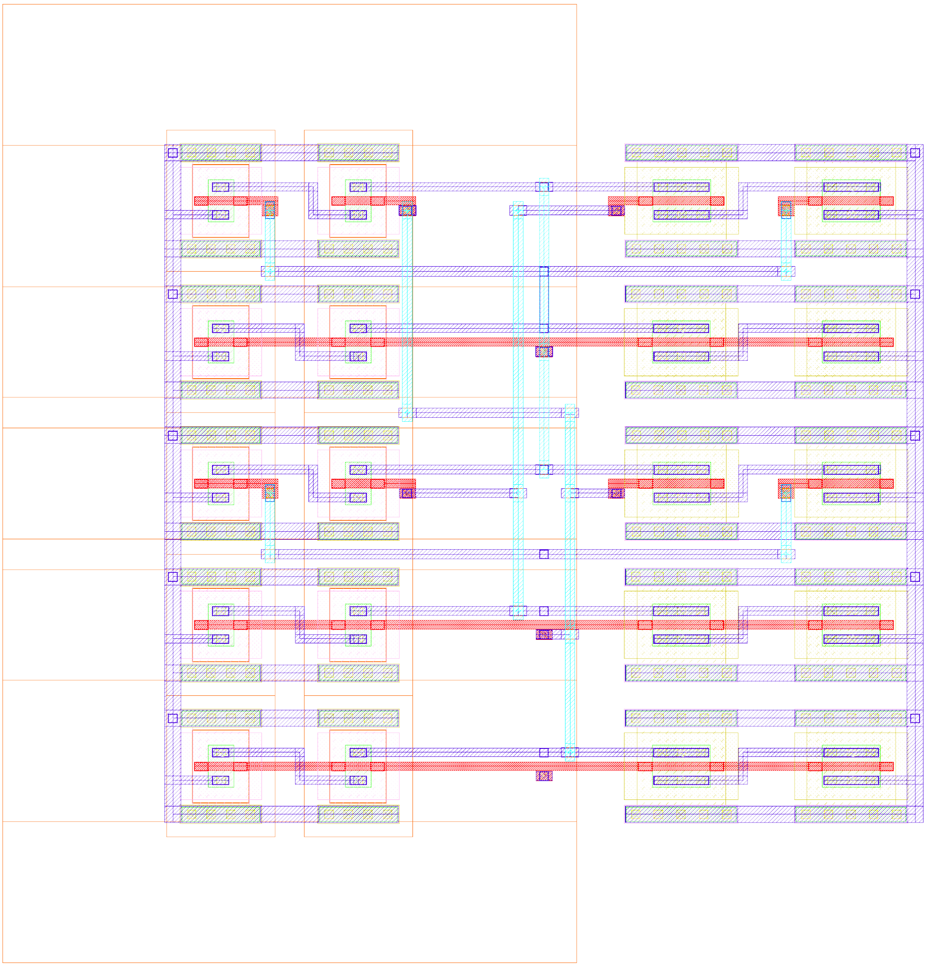


Figure C.8: C$^2$MOS Latch

## C.9 C$^2$MOS D-Flip-Flop

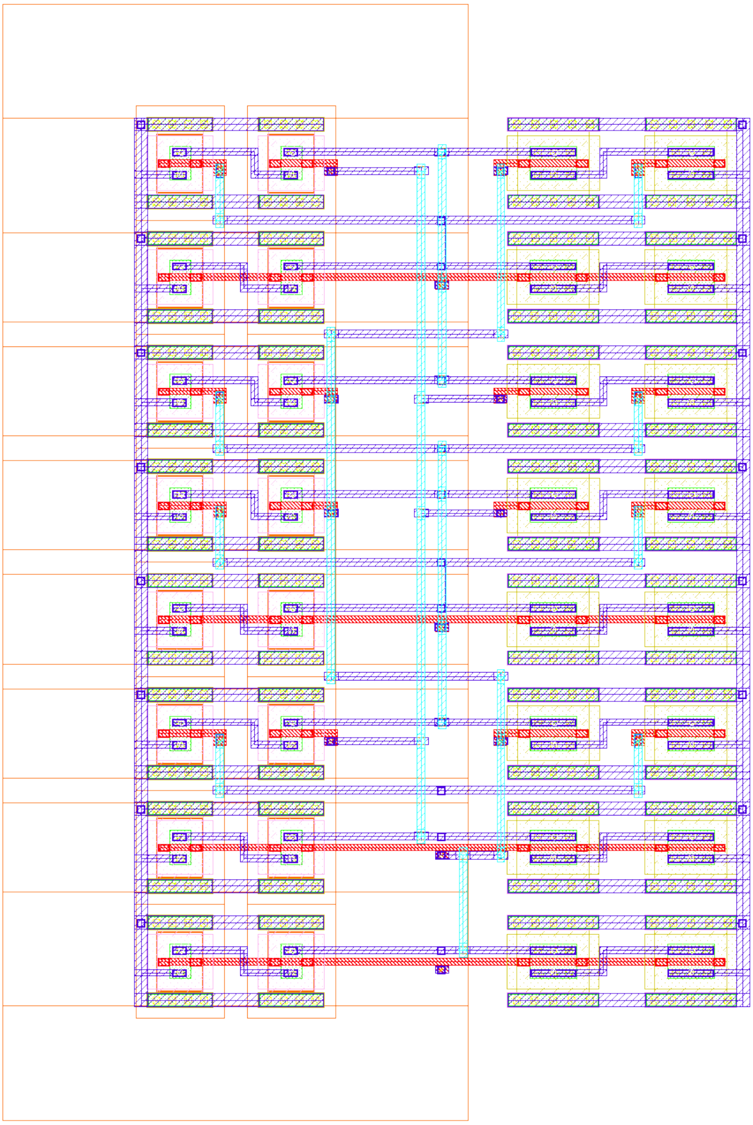


Figure C.9: C$^2$MOS D-Flip Flop

## C.10 C$^2$MOS D-Flip-Flop with async. reset

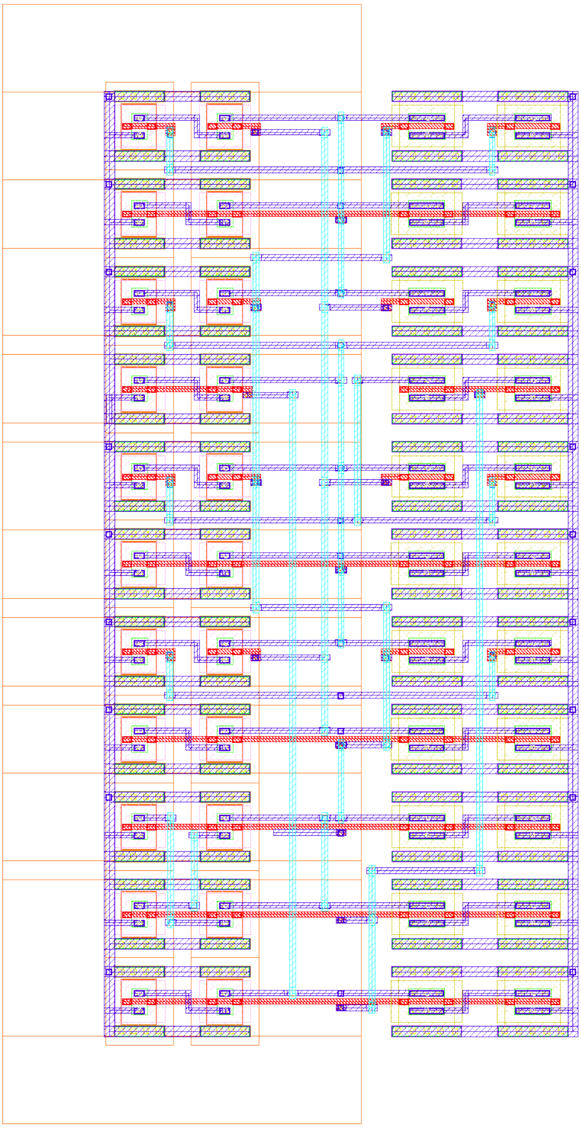


Figure C.10: C$^2$MOS D-Flip-Flop with async. Reset

## C.11 Clock-gate

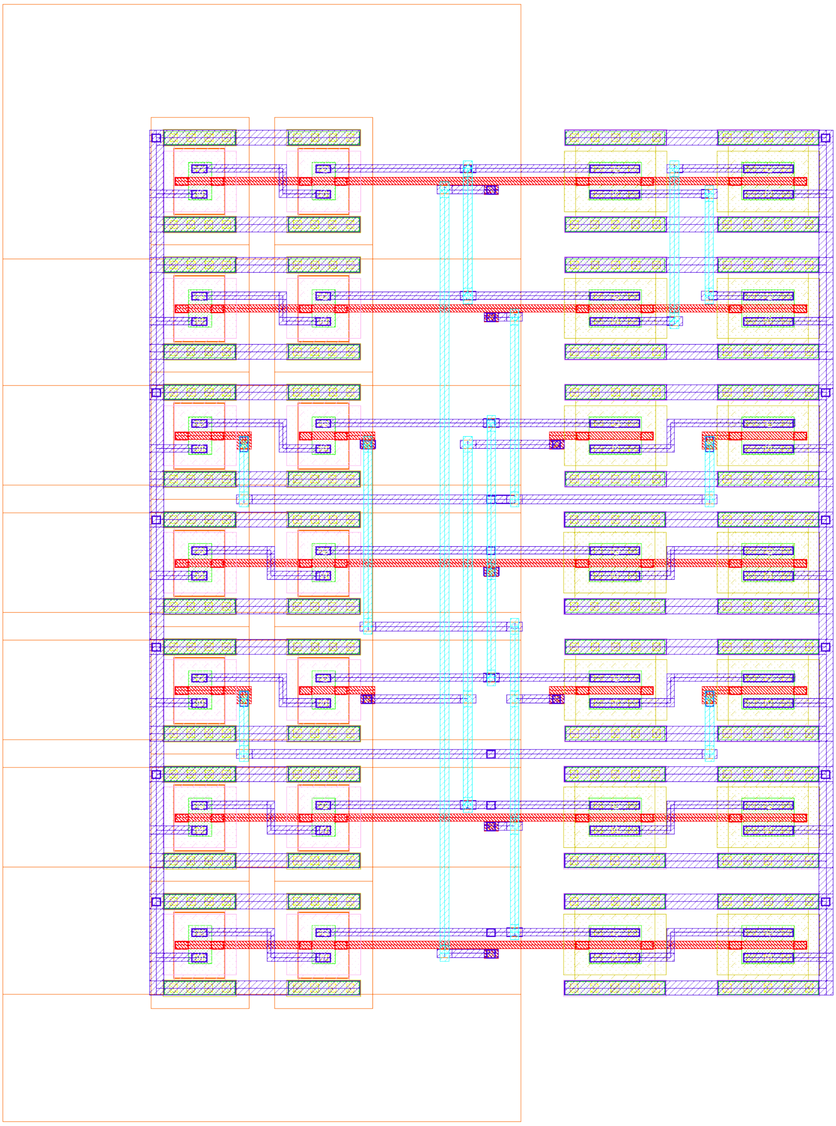


Figure C.11: Clock-gate

## C.12 Compare Bit with Xor

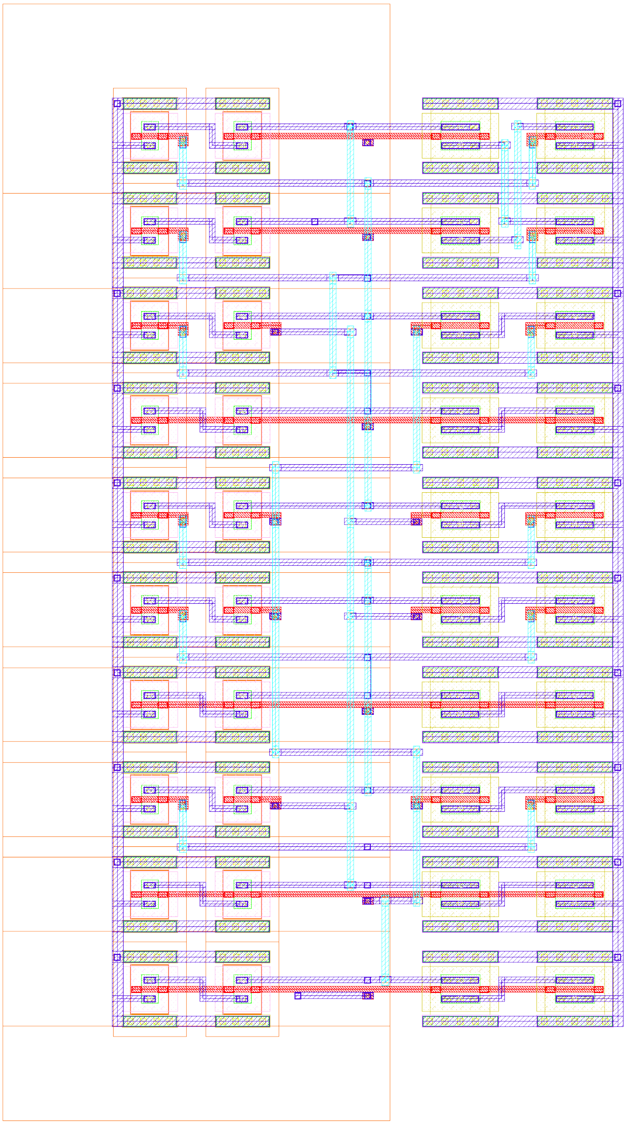


Figure C.12: compBitXor

## C.13 Compare Bit with XOR and NOR

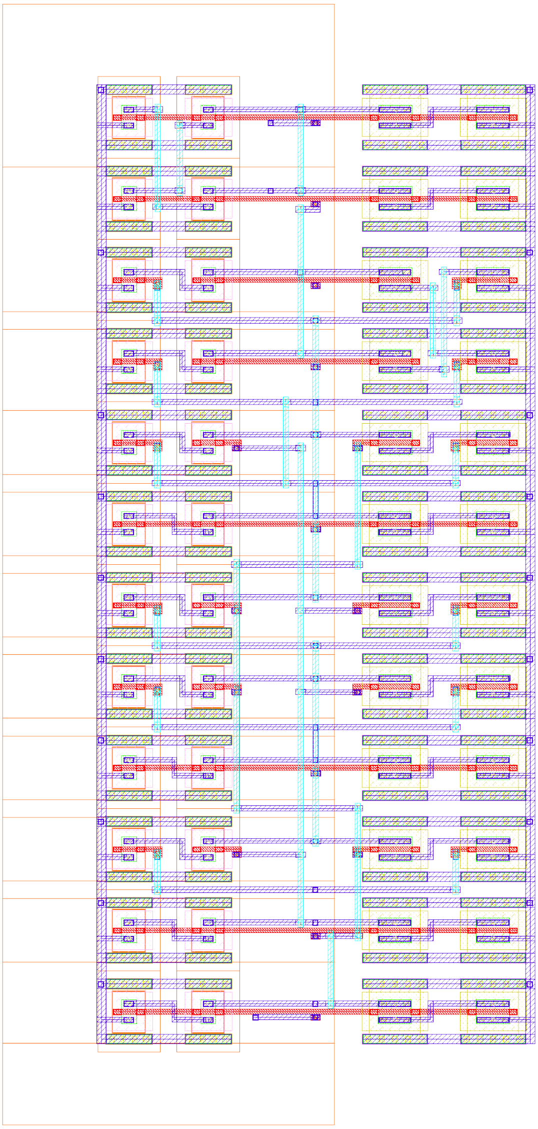


Figure C.13: compBitXorNor

## C.14 Compare Bit with XNOR and NAND

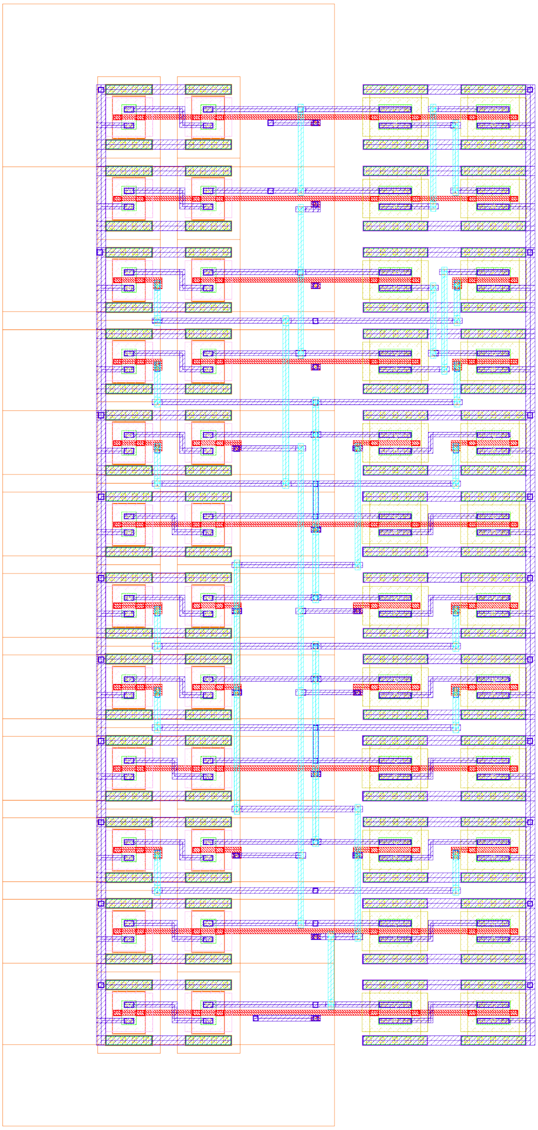


Figure C.14: compBitXnorNand

## C.15 Bit with XOR and NOR

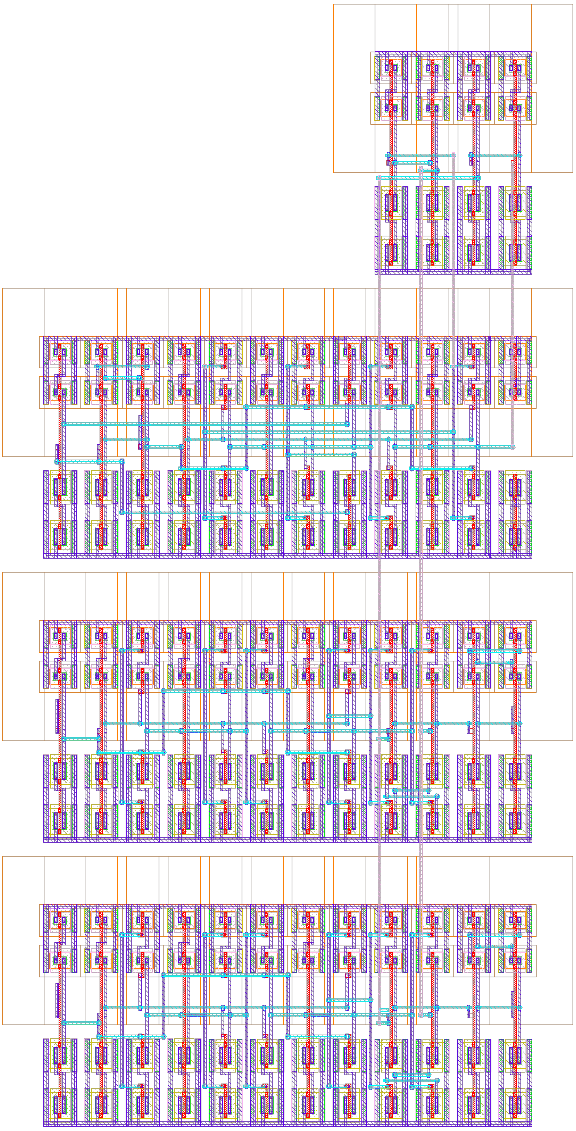


Figure C.15: bitWithXorNor

## C.16 Bit with Adder and XOR

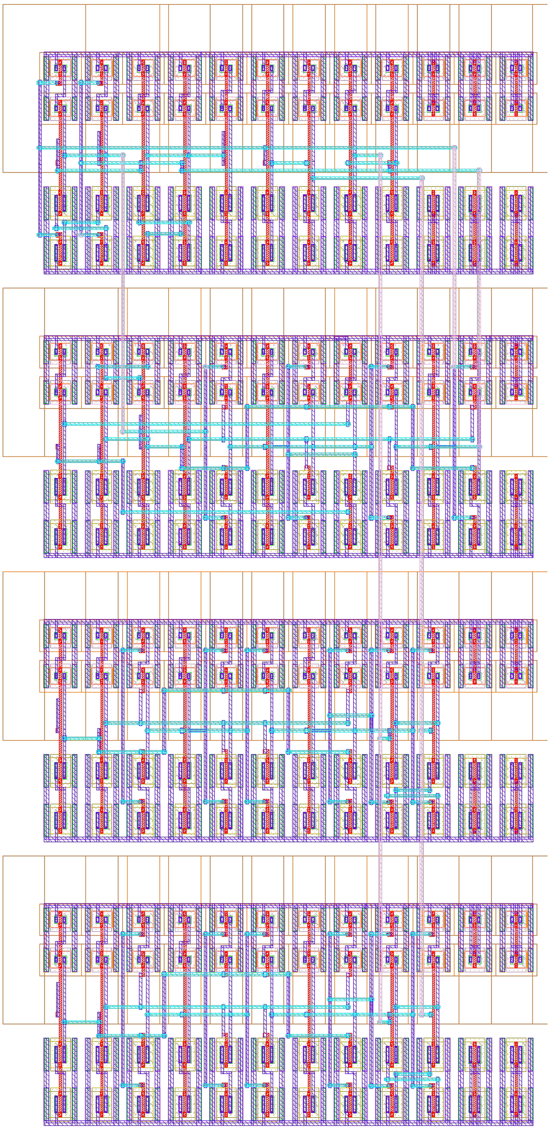


Figure C.16: bitWithAdderXor

## C.17 Bit with Adder, XOR and NOR

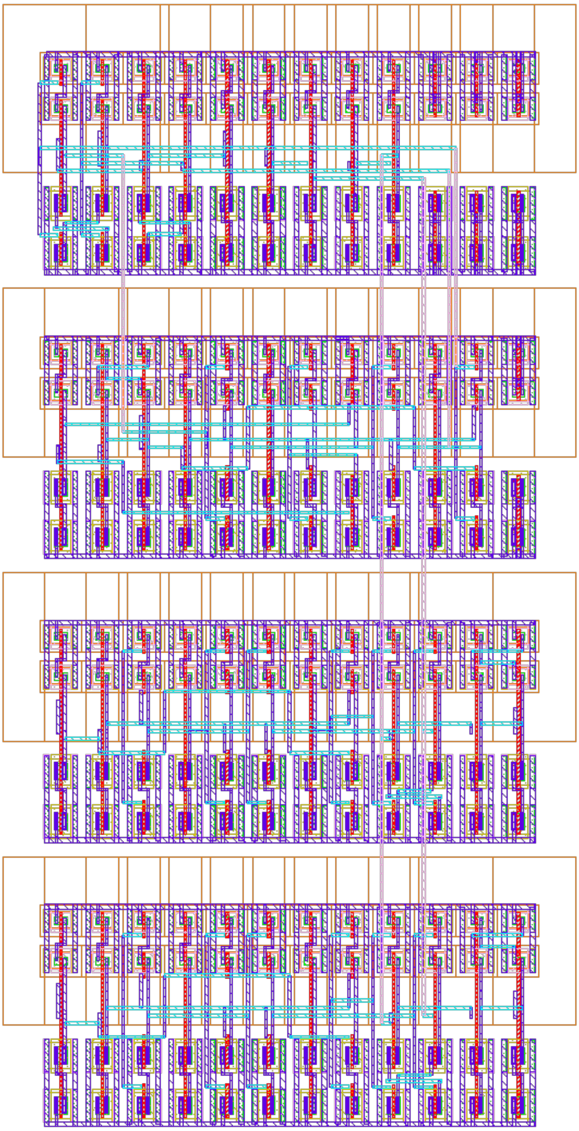


Figure C.17: bitWithAdderXorNor

## C.18 Bit with Adder, XNOR and NAND

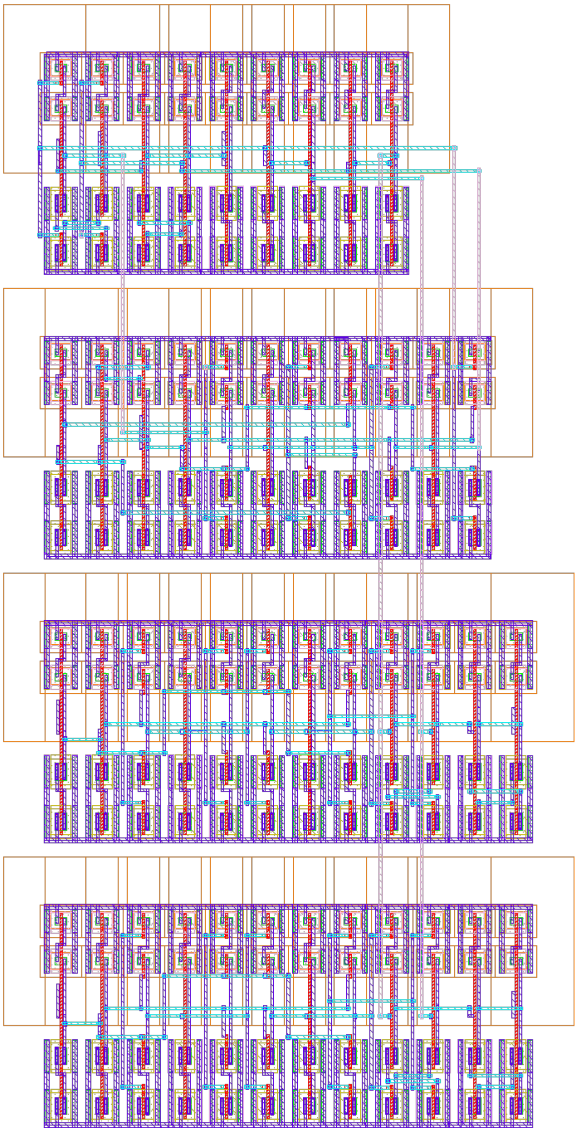


Figure C.18: bitWithAdderXnorNand

## C.19 RTC

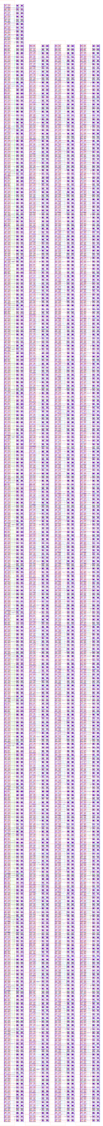


Figure C.19: RTC

# Bibliography

[1] M. Alioto. Ultra-low power vlsi circuit design demystified and explained: A tutorial. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(1):3–29, Jan. 1, 7, 9

[2] H.P. Alstad and S. Aunet. Seven subthreshold flip-flop cells. In *Norchip, 2007*, pages 1–4, 2007. 13

[3] Energy Micro AS. Efm32 referance manual. *Online: http://www.energymicro.com.* i, ix, 11, 12

[4] A. Bellaouar, A. Fridi, M.I. Elmasry, and K. Itoh. Supply voltage scaling for temperature insensitive cmos circuit operation. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 45(3):415–417, 1998. 9

[5] M. Blesken, S. Lutkemeier, and U. Ruckert. Multiobjective optimization for transistor sizing sub-threshold cmos logic standard cells. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1480–1483, 2010. 37

[6] A. Chavan, G. Dukle, B. Graniello, and E. MacDonald. Robust ultra-low power subthreshold logic flip-flop design for reconfigurable architectures. In *Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference on*, pages 1–7, 2006. 13

[7] D. Huang and W. Li. On cmos exclusive or design. In *Circuits and Systems, 1989., Proceedings of the 32nd Midwest Symposium on*, pages 829–832 vol.2, 1989. 28

[8] H. Kristian, O. Berge, and S. Aunet. Multi-objective optimization of minority-3 functions for ultra-low voltage supplies. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2313–2316, 2011. 21

[9] L.L. Lewyn, T. Ytterdal, C. Wulff, and K. Martin. Analog circuit design in nanoscale cmos technologies. *Proceedings of the IEEE*, 97(10):1687–1714, 2009. 37

[10] Preeti Ranjan Panda, B. V. N Silpa, and Krishnaiah Shrivastava, Aviral A1 Gummidipudi. *Power-efficient System Design*. Springer US, Boston, MA, 2010. ix, 5, 6, 7

[11] Xiaoning Qi, S.C. Lo, A. Gyure, Yansheng Luo, M. Shahram, Kishore Singhal, and D.B. MacMillen. Efficient subthreshold leakage current optimization - leakage current optimization and layout migration for 90- and 65- nm asic libraries. *Circuits and Devices Magazine, IEEE*, 22(5):39–47, 2006. 6

[12] V. Suzuki, K. Odagawa, and T. Abe. Clocked cmos calculator circuitry. *Solid-State Circuits, IEEE Journal of*, 8(6):462–469, 1973. 27

[13] John P. Uyemura. *Introduction to VLSI circuits and systems*. Wiley, New York, 2002. 6

[14] Eric A. Vittoz. Design of VLSI Circuits for Telecommunication and Signal Processing, Micropower techniques. 1994. 8

[15] Alice Wang, Benton H Calhoun, and Anantha P Chandrakasan. *Subthreshold Design for Ultra Low-Power Systems*. Springer US, Boston, MA, 2006. 1, 8, 9, 10, 13, 60