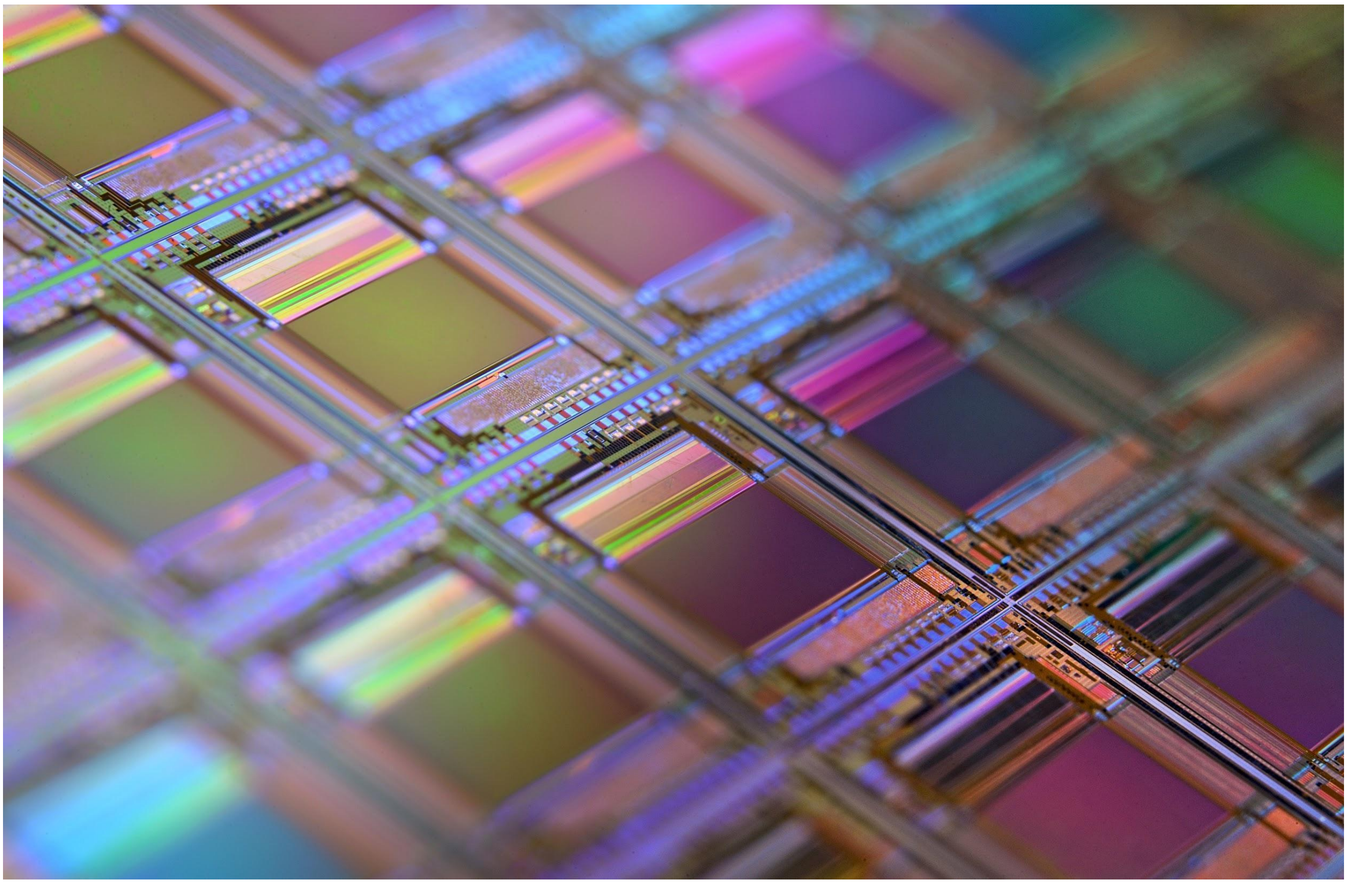
UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

DESIGN/SIMULATION OF MICROELECTRONICS USING VLSI DESIGN TECHNIQUES



Student: Giannakoulis Alexandros

Registration Number: 50106666

Supervisor

George P. Patsis

Professor

ATHENS-EGALEO, MARCH 2021

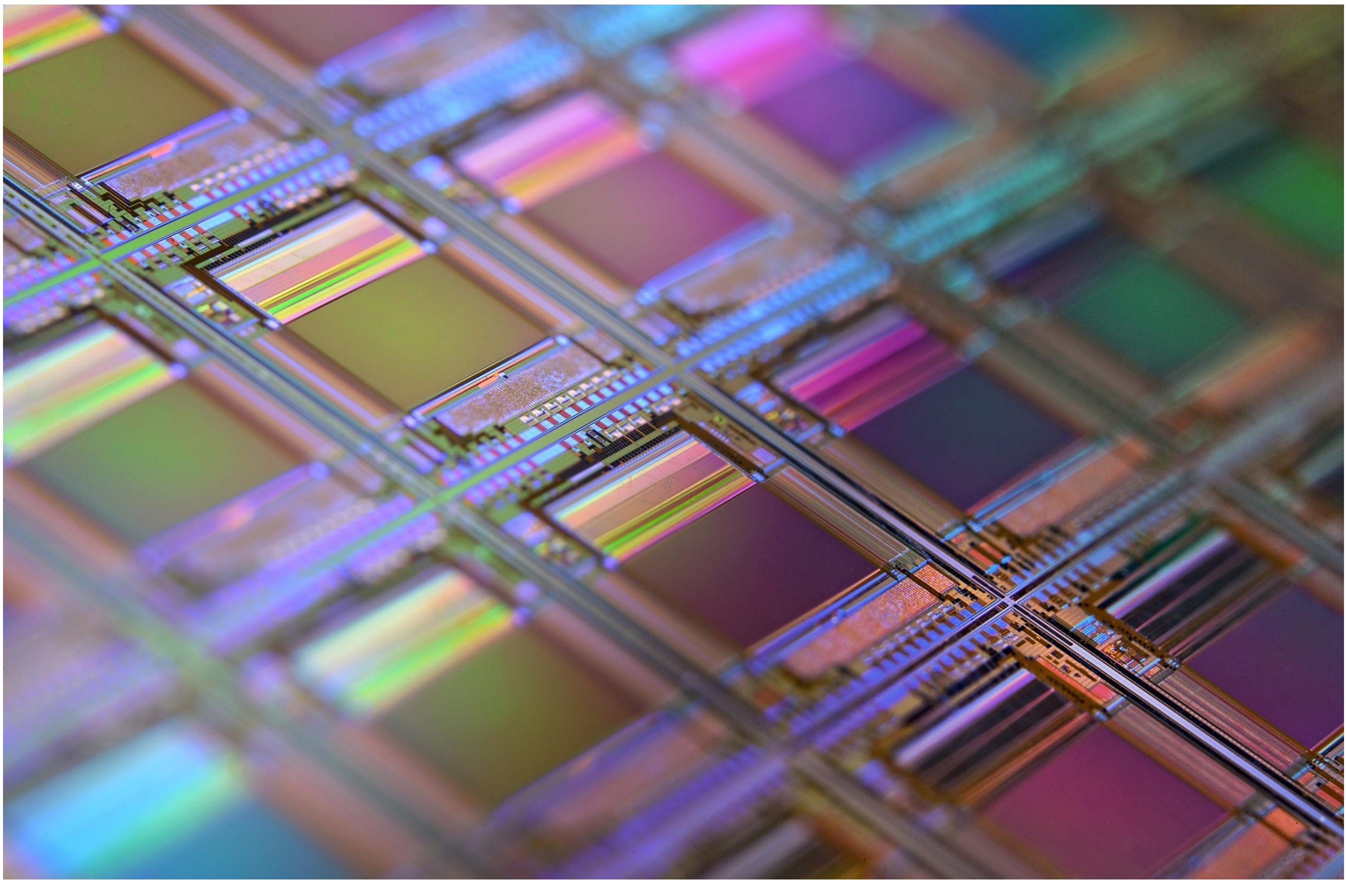
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

ΣΧΕΔΙΑΣΗ/ΠΡΟΣΟΜΟΙΩΣΗ ΜΙΚΡΟΗΛΕΚΤΡΟΝΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ ΜΕ ΤΕΧΝΙΚΕΣ VLSI



Φοιτητής: Γιαννακούλης Αλέξανδρος
ΑΜ: 50106666

Επιβλέπων καθηγητής
Γεώργιος Π. Πάτσης
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΜΑΡΤΙΟΣ 2021

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπόγραφα ότι η παρούσα εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα αποκλειστικά και ότι είμαι ο αποκλειστικός συγγραφέας του κειμένου της.

Η εργασία μου δεν προσβάλλει οποιασδήποτε μορφής δικαιώματα πνευματικής ιδιοκτησίας, προσωπικότητας ή προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής ή λογοκλοπής.

Κάθε βοήθεια που έλαβα για την ολοκλήρωση της εργασίας είναι αναγνωρισμένη και αναφέρεται λεπτομερώς στο κείμενό της. Ειδικότερα, έχω αναφέρει ευδιάκριτα μέσα στο κείμενο και με την κατάλληλη παραπομπή όλες τις πηγές δεδομένων, κώδικα προγραμματισμού Η/Υ, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών που χρησιμοποιήθηκαν, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης, και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Επιπλέον, όλες οι πηγές που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης κατά τα διεθνή πρότυπα.

Τέλος δηλώνω ενυπόγραφα ότι αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της είναι προϊόν λογοκλοπής.

Ημερομηνία 4-3-2021

Γιαννακούλης Αλέξανδρος

ΜΕΛΗ ΕΞΕΤΑΣΤΙΚΗΣ ΕΠΙΤΡΟΠΗΣ

Γεώργιος Π. Πάτσης, καθηγητής	ΓΕΩΡΓΙΟΣ ΠΑΤΣΗΣ Digitally signed by ΓΕΩΡΓΙΟΣ ΠΑΤΣΗΣ DN: cn=ΓΕΩΡΓΙΟΣ ΠΑΤΣΗΣ, c=GR, o=University of West Attica, email=patsisg@uniwa.gr Date: 2021.03.05 07:35:59 +02'00'
Βαλαμόντες Ευάγγελος, καθηγητής	Digitally signed by Grigorios Kaltsas Date: 2021.03.04 15:21:42 +02'00'
Καλτσάς Γρηγόριος, καθηγητής	Evangelos Valamontes Digitally signed by Evangelos Valamontes Date: 2021.03.04 15:46:58 +02'00'

I would like to thank Professor George Patsis for the support and guidance he provided to me during the writing of this Thesis. He helped me become interested in the subject and I am grateful for that opportunity.

I would also like to thank my family, especially my mother, father and brother for their support and assistance throughout my studies.

Abstract

In this paper, simple logic circuits were used to design more advanced, yet basic and essential MOS components. The 4-Stage Binary Counter can count sequentially2^4bits and is used in a variety of applications like program counting, random number generation and as a counter in general. It consists of a simple design, but is important to also be familiar with the basic components that are used in the making of the counter. In order for the reader to have a general idea, there are a lot of design choices and in this paper there are a few different designs of the same components.

The random number generation process is very important especially in cryptography and in randomizing elements like in games. Random number generation can be achieved with software programs and with hardware devices. A hardware random number generation is a device that generates numbers from a physical process, in contrast to algorithmical. A true random number can be obtained only through hardware, because the microscopic phenomena used are, in theory, completely unpredictable.

The program counter and instructions decoder are important components in addressing the programs and processes running in the memory, as well as which system in the integrated circuit needs the attention of the processor and for what period of time. These components work closely with memory and the processor.

Memory is an essential component of every processor and advanced application. There are a lot of types of memories with their advantages and disadvantages depending on the application. This paper deals with Static Random Access Memory cell designs, which are easy to make and comprehensible, as well as generally good and reliable memory types. Additionally, memory needs some extra components in order to work properly, like a decoder to write reliably the new information into the memory cells and a multiplexer to read the information from the memory cells.

However, CMOS circuits aren't only digital, there are a lot of analog signals in a CMOS circuit which need constant conversion to digital as well as to analog and also some kind of comparison between them. Components like the differential amplifier are important in a variety of applications such as memory, for comparison of signals and also as amplifiers.

Keywords

CMOS, VLSI, logic gates, binary counters, integrated circuits, SRAM, analog CMOS components, amplifiers.

Contents

1. The Microwind35 Program

1.1. Introduction

The Microwind software allows the designer to simulate and design an integrated circuit at physical description level. Born in Toulouse (France), Microwind is an innovative CMOS design tool for the educational market.

Microwind is developed as a comprehensive package on windows platform to enable students to learn smart design methods and techniques with more practice. With inbuilt layout editing tools, mix-signal simulator, MOS characteristic viewer and more, it allows students to learn the complete design process with ease.

Microwind unifies schematic entry, pattern based simulator, SPICE extraction of schematic, Verilog extractor, layout compilation, on layout mixed-signal circuit simulation, cross sectional & 3D viewer, netlist extraction, BSIM4 tutorial on MOS devices and sign-off correlation to deliver unmatched design performance and productivity.

1.2. User Interface & Toolbar

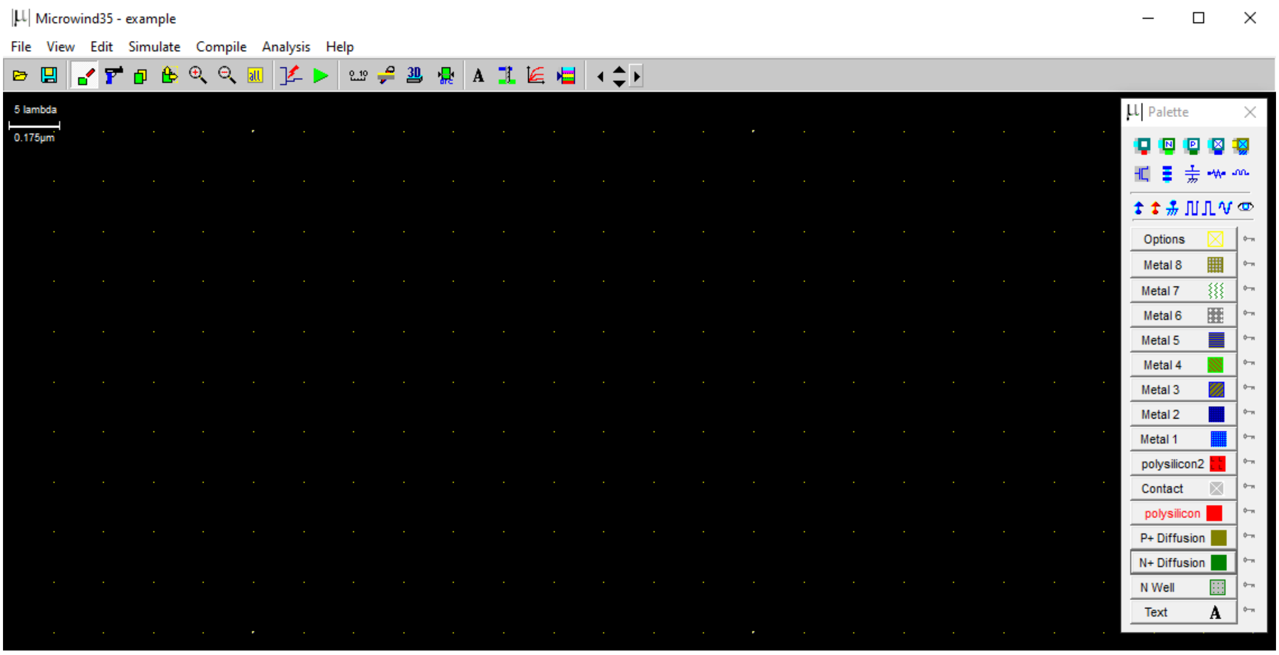


Figure 1.1. Microwind35 Interface and toolbar

The user interface is quite simple and easy to use. All the basic tools are in the Palette menu and it also has submenus in which the user can specify the details of a component. It offers quite a few simulation options and advanced models(BSim4) to accurately reflect the transistor's behavior. The user can change the foundry at any time and also the program offers different design rule checker options for every foundry.

In the simulation tab the user can change the simulation parameters like the temperature of the environment in which the component operates, the MOS model and some scenario options. It can also show the 2D section of the components and the steps of the manufacturing process in 3D.

In conclusion, the Microwind program offers all the basic and even some advanced options for designing and simulating cmos circuits and it is quite easy to use by students and people new to the subject.

1.3. The Dsch35 Program

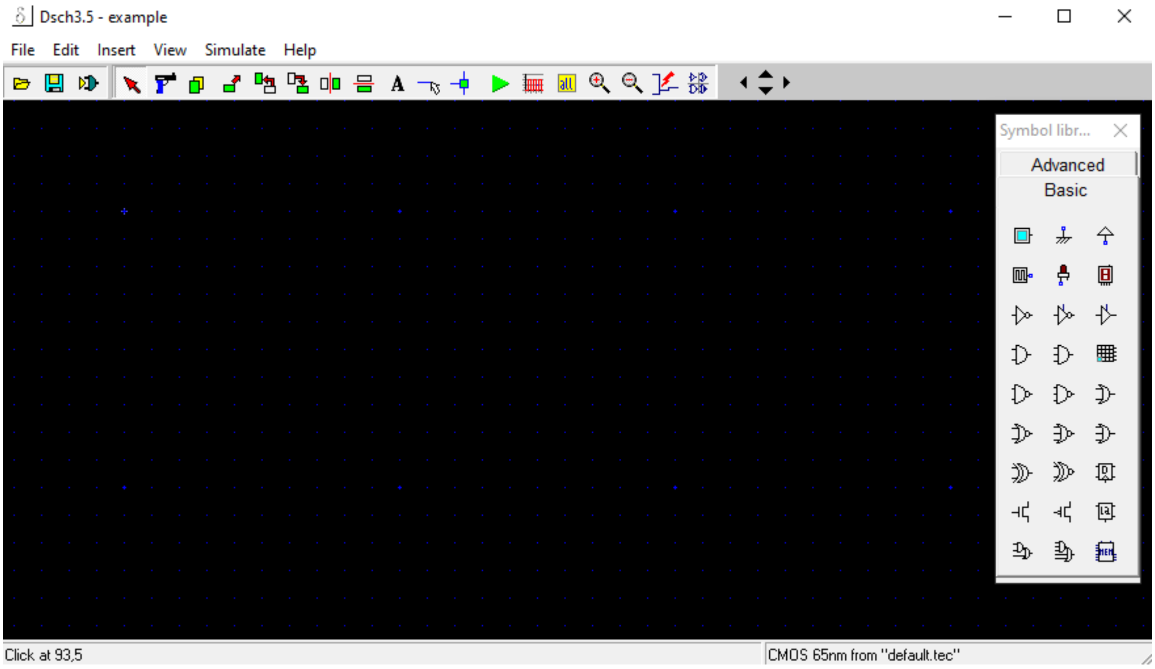

Dsch3.5 - example
File Edit Insert View Simulate Help
Symbol libr...
Advanced
Basic
Click at 93,5
CMOS 65nm from "default.tec"

Figure 1.2. Dschematic35 Interface and toolbar

This program is similar to the Microwind and it is made by the same corporation. It has fewer and simpler options than Microwind, but is a good tool in designing and simulating simple logic circuits. The user can design and simulate a logic circuit or component in Dsch35 before designing it in Microwind35. The components and a variety of logic gates can be found in the Symbol Library and it also offers the option of making new Symbols and Schemas for the designing of more advanced circuits. Lastly, it offers a timing diagram simulation and a few simulation options like making adjustments to the elementary gate current and the gate and wire delay.

2. 4-Stage Binary Counter

The binary counter is made by four D-registers cascaded. The clocking of each stage is simply carried out by the previous counter, to form an asynchronous counter circuit. The 4-stage binary counter displays numbers from 0 to 15, using four D-registers (Figure 2.1). The 4-Stage Binary Counter was made with 0.12um(cmos012.rul) technology.

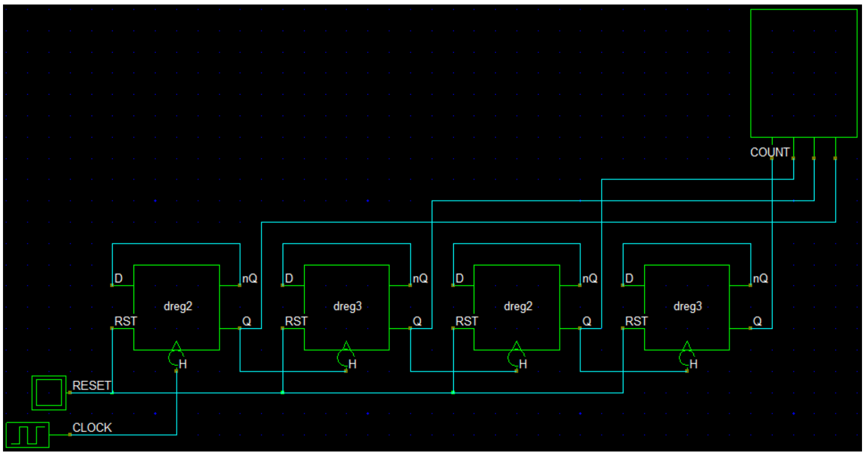


Figure 2.1. 4-Stage Binary Counter with four D-Registers in Dsch35

The D-registers are edge triggered latches, where the information flows from the input 'D' to the output 'Q' only on a rising edge of the clock. The structure of these latches is a Master-Slave structure, where the D-Register is made by two D-Latches and a NOT Gate(Figure 2.2).

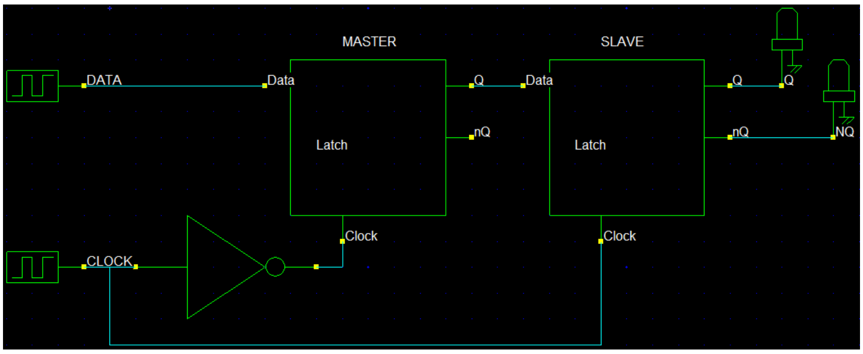


Figure 2.2. D-Register in Dsch35

2.1. The NOT Gate

The NOT Gate is the first step of the design. Its simple design(Figures 2.3 & 2.4) is the base of the project. Its main feature is to perform logical negation on its input. In other words, if the input is true(1), then the output will be false(0). Similarly, a false input results in a true output.

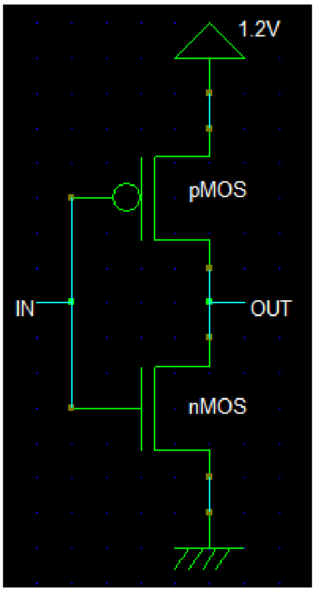


Figure 2.3. NOT Gate in Dsch35

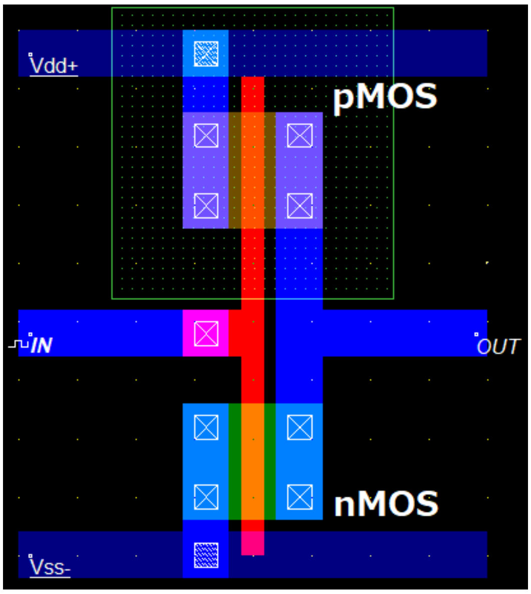


Figure 2.4. NOT Gate in Microwind35

It works as intended in the simulation(Figure 2.5).

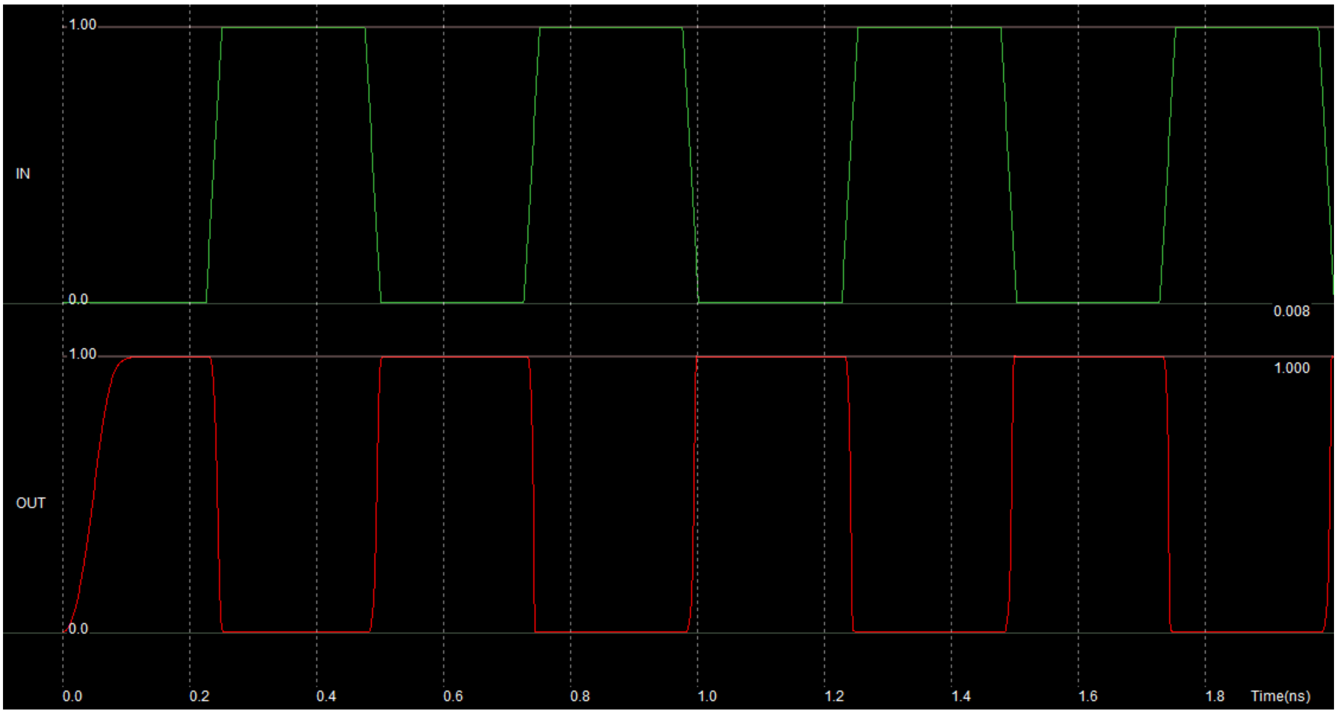


Figure 2.5. NOT Gate simulation in Microwind35

2.2. The D-Latch

Next, is the design of the D-Latch(Figure 2.6). Using the software Dsch35(Figure 2.7), where ‘Clock’ and ‘Data’ are respectively the clock and the input of the latch. The ‘Q’ and ‘NQ’ are the non-inverting and inverting outputs respectively.

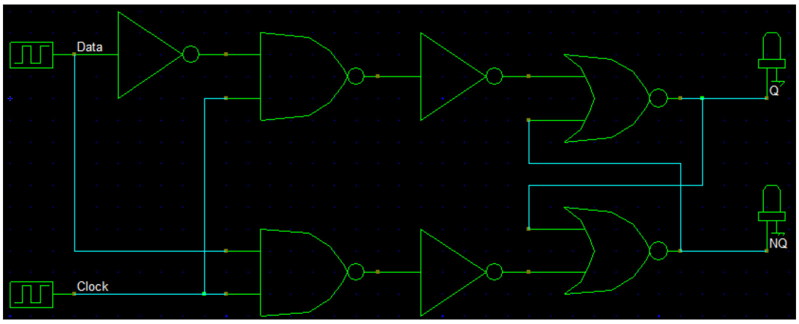


Figure 2.6. D-Latch in Dsch35

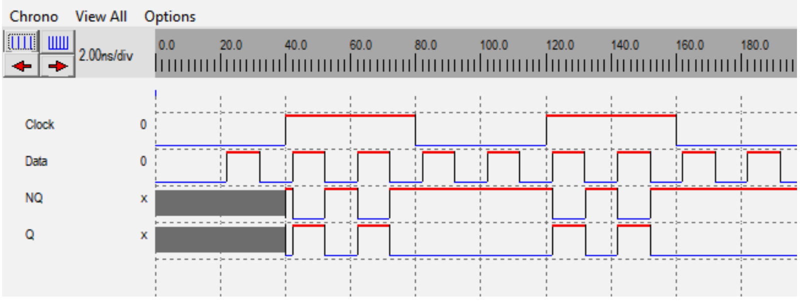


Figure 2.7. D-Latch simulation in Dsch35

When the ‘Clock’ input is high(1), the latch output ‘Q’ follows the changes of the ‘Data’ input. When the ‘Clock’ input is low(0), the latch is in memory mode, meaning that the final input state is latched on the outputs. When performing the logic simulation, ‘Q’ and ‘NQ’ start with an undetermined state(appearing in grey in Figure 2.7).

The design of the D-Latch(Figure 2.6), although it works great, can become more compact. The direct implementation uses 22 MOS devices. In order to obtain a compact design, complex gates should be used(Figures 2.8 & 2.9). The design with the complex gates uses only 14 MOS devices(Figure 2.10). Additionally, the complex gate solution leads to shorter propagation delay.

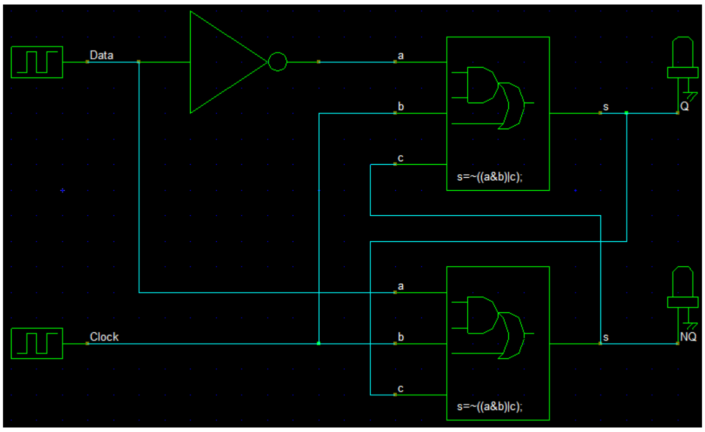


Figure 2.8. The compact design of the D-Latch in Dsch35

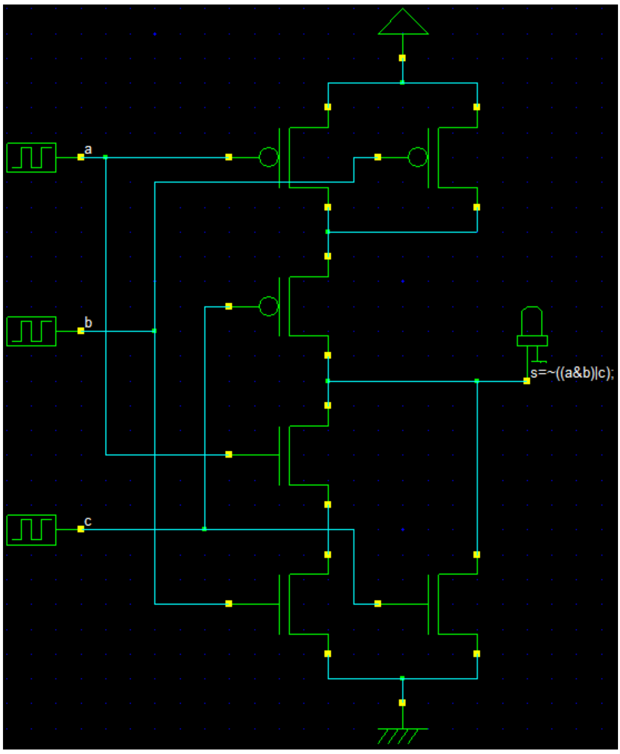


Figure 2.9. The detailed design of the D-Latch in Dsch35

Now, is the time to design the D-latch which is composed of two components(complex gates) as shown on Figure 2.8 and a NOT Gate. From the result of the simulation(Figure 2.11), it is noticeable that the D-latch works properly.

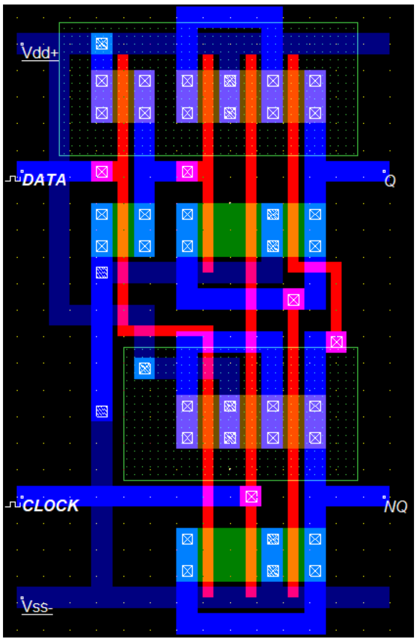


Figure 2.10. D-Latch in Microwind35

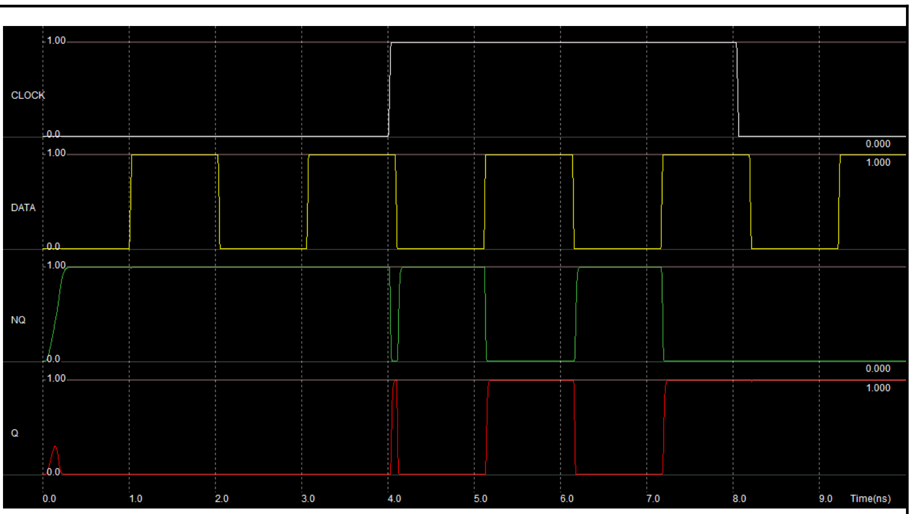


Figure 2.11. D-Latch simulation in Microwind35

2.3. The D-Register

The design of the D-Register is based on a Master-Slave structure combining two D-Latches and a NOT Gate following the schematic below.

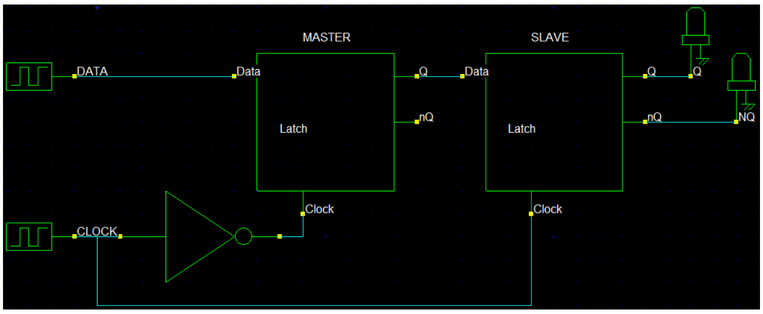


Figure 2.12. D-Register in Dsch35

The first D-Latch's output is assigned to the second D-Latch's input and the ‘Clock’ enters directly into the second D-Latch and through a NOT Gate into the first one. The design of the D-Register as well as its simulation, is shown in the Figures below.

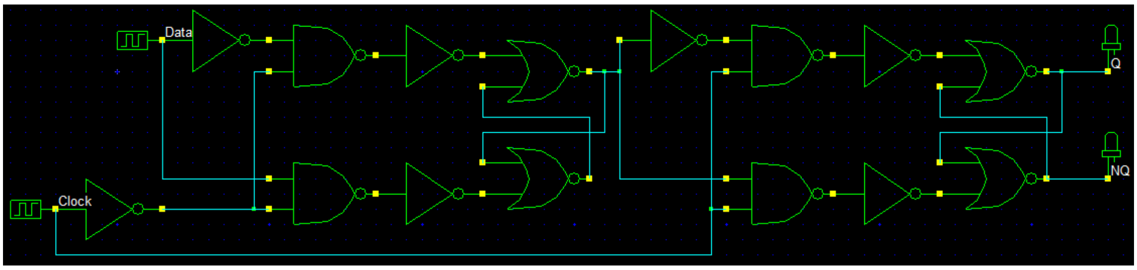


Figure 2.13. The detailed design D-Register in Dsch35

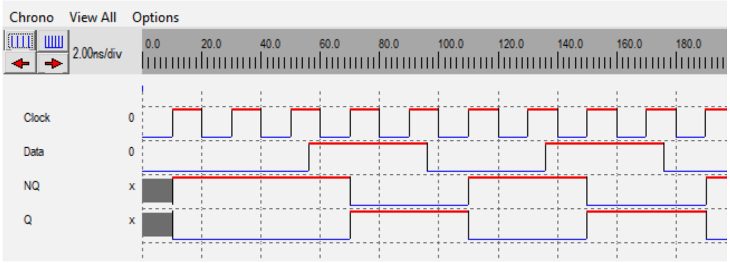


Figure 2.14. D-Register simulation in Dsch35

From the simulation(Figure 2.14), the input 'Data' state is transferred to the output 'Q' on every rising edge of the 'Clock'.

The first D-Register's metal output is linked to the second D-Register's NOT Gate's silicon input and the ‘Clock’ enters directly on the second D-Register's silicon clock input and through a NOT Gate on the first one's silicon clock input.

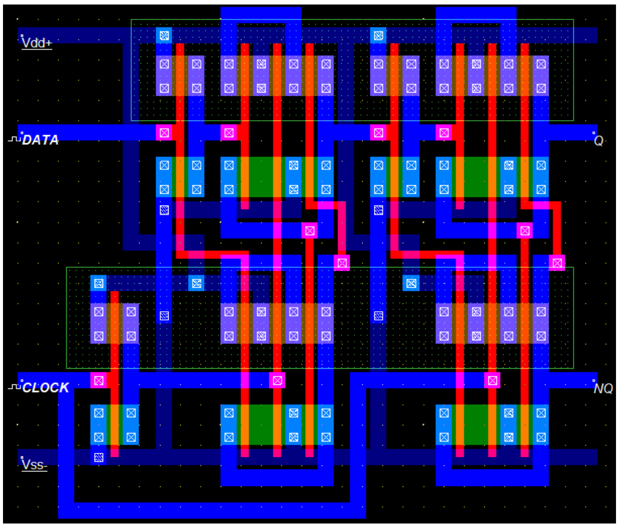


Figure 2.15 D-Register in Microwind35

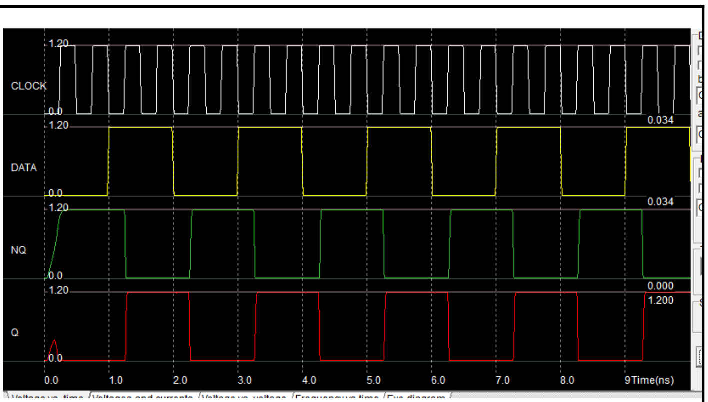


Figure 2.16 D-Register simulation in Microwind35

From the simulation(Figure 2.16), the results are similar to the ones obtained using Dsch35. The input ‘Data’ state is transferred to the output ‘Q’ on every rising edge of the ‘Clock’ and the output ‘NQ’ is the opposite of ‘Q’.

2.4. 4-Stage Binary Counter

Figure 2.17 shows how four D-Registers are linked(like in Figure 2.1), in order to assemble the 4-Stage Binary Counter. The negative output ‘NQ’ of each D-Register returns to its input ‘Data’ and to the following D-Register’s ‘Clock’ input. The positive output from every D-Register is one of the counter’s outputs.

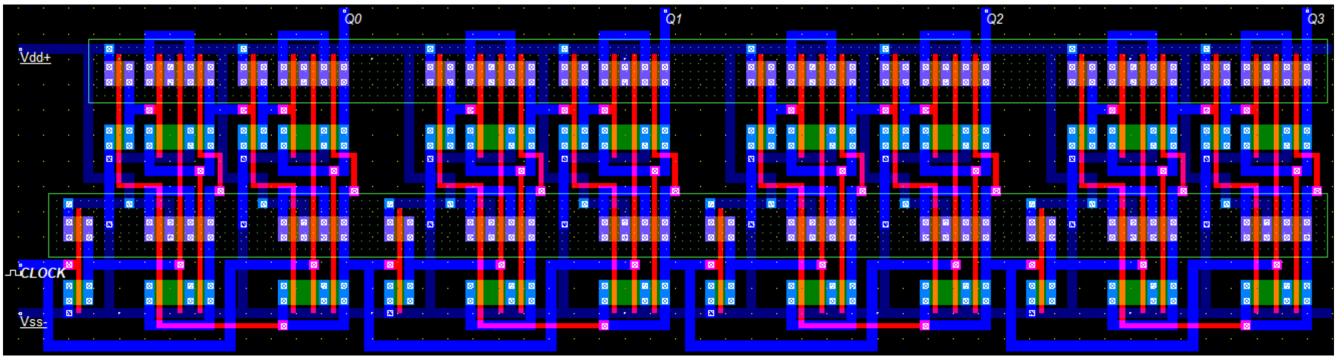


Figure 2.17. 4-Stage Binary Counter in Microwind35

There is only one input ‘Clock’ which is the first D-Register's clock input and four outputs which are ‘Q0’, ‘Q1’, ‘Q2’, ‘Q3’(one output for each D-Register). Finally, in the simulation(Figure 2.18) the 4-Stage Binary Counter increased from 0 to 15 on every clock’s rising edge and back to 0 to count again.

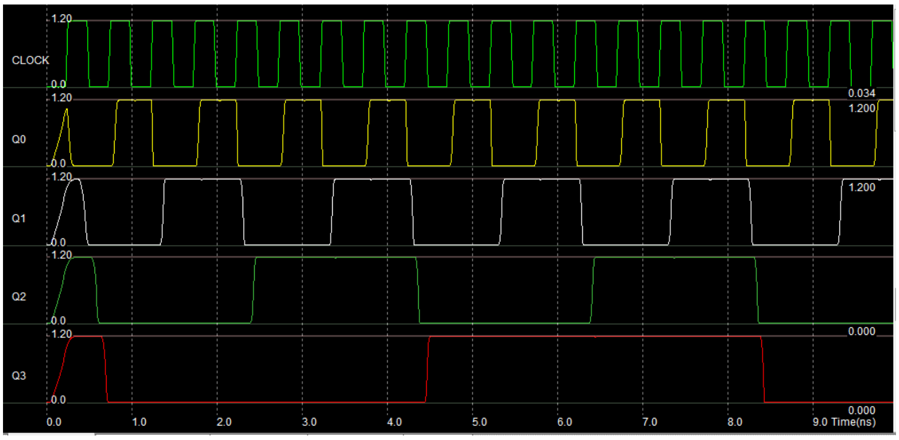


Figure 2.18. 4-Stage Binary Counter simulation in Microwind35

Now that the design is complete and it works well, there are extra functions that could be added to the design. A reset function could have been easily implemented to reset the counter. A new input could also be added, to select whether to increase or decrease the count in every clock’s rising edge.

A few logic gates and some simple MOS designs, are the building blocks of the 4-Stage Asynchronous Binary Counter which is ready to be realised on a silicon wafer(Figure 2.19). Based on 12um technology, its dimensions are about 8um x 32um. The design of this counter utilizes all the basic concepts of the transistor’s operation.

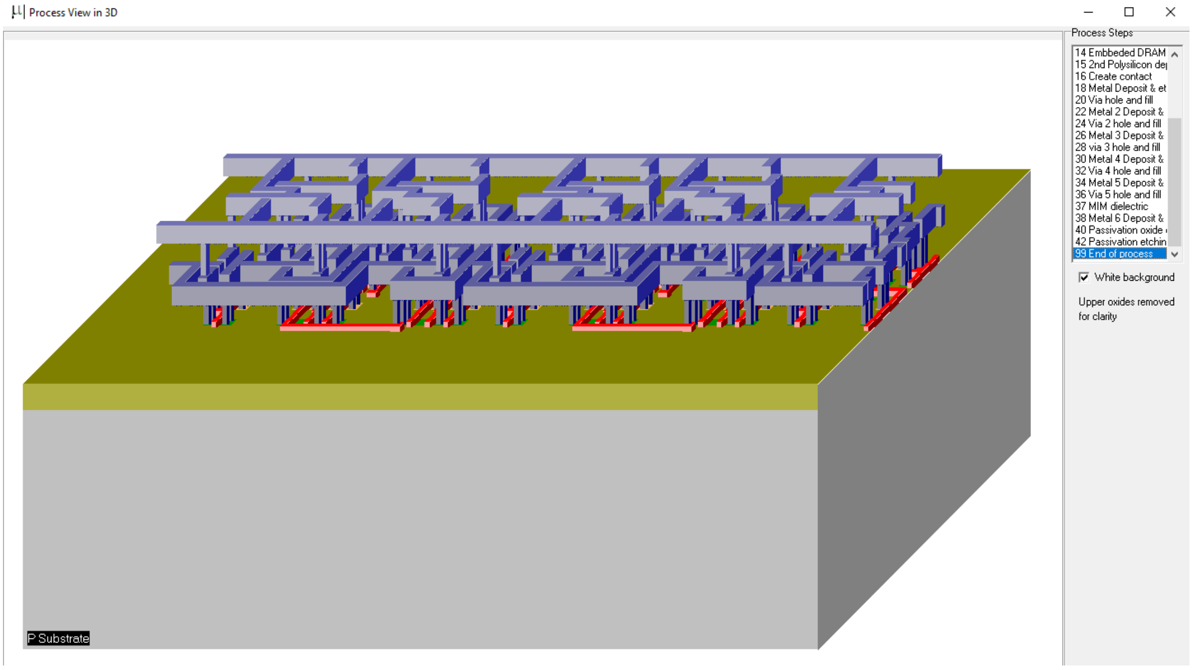


Figure 2.19. 3D Process View of the 4-Stage Binary Counter in Microwind35

3. Random Number Generator

Most numbers are generated with the help of pseudorandom number generation software, even though they are problematic(lack of uniformity, correlation between values, repeatability of sequences).

On the left, is the design of an integrated circuit for random number generation. There are some specification that the circuit must have:

- The component needs to be easy to use with one or two inputs and with only parallel outputs.
- The component must give hexadecimal random numbers.
- The component must be able to generate random numbers at the specific frequency of 10MHz.

Firstly, as in the previous chapter, every component of the integrated circuit needs to be designed separately. In addition, these components need to function properly together. Lastly, it is important to observe how the integrated circuit operates, what its limits are and what could be done to improve it.

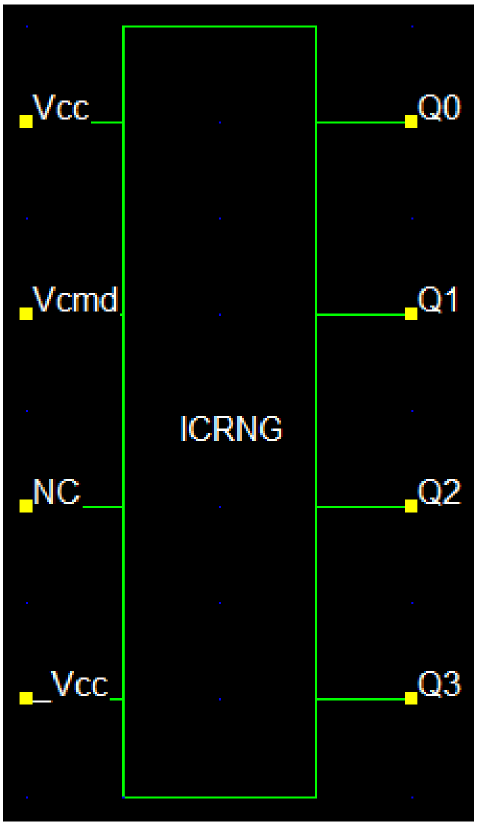


Figure 3.1. The external design of the Integrated Circuit for Random Number Generation (ICRNG) in Dsch35

The principle of operation is that the generator is a fast asynchronous counter, which counting will stop by an external action(very low frequency compared to the counter period), like the push of button. The external design of the component can be seen in Figure 3.1. The 'Vcmd' input allows for the start and pause of the counting, according to its level('1' allows counting).

The Integrated Circuit for Random Number Generation (ICRNG) consists of four D-Registers, an AND Gate and an Oscillator(Figure 3.2). The oscillator frequency will be defined later. It needs to be noted that the ICRNG and its elements were made with 65nm(cmos65n.rul) technology.

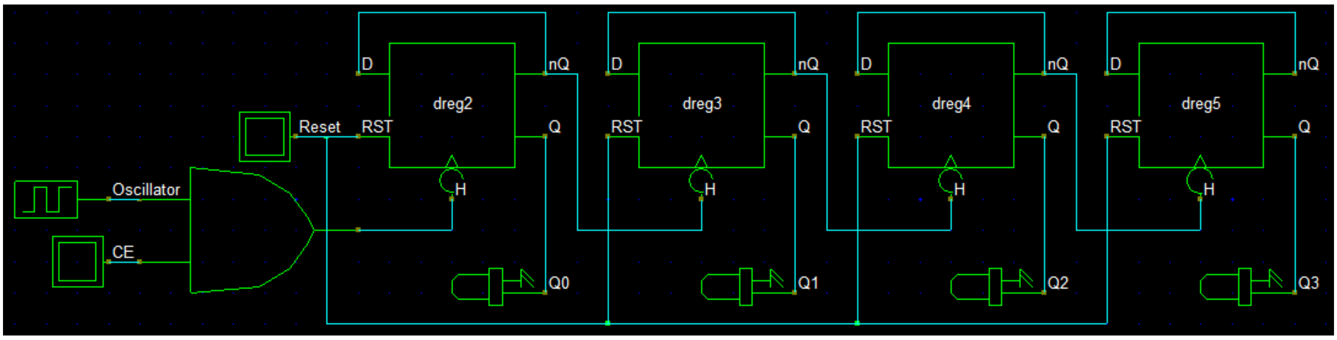


Figure 3.2. The internal design of the ICRNG in Dsch35

3.1. The Inverter

The Inverter(Figures 3.3. & 3.4.), also known as NOT Gate, has a very simple design and consists of a p-type and an n-type MOS transistors connected to each other by their 'bases' and 'sources'. The 'drain' of the pMOS is connected to high voltage(Vdd+) and the 'drain' of the nMOS is connected to the ground.

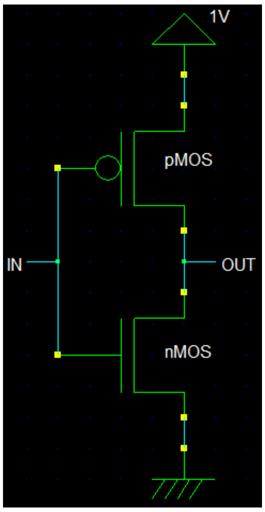


Figure 3.3. Inverter in Dsch35

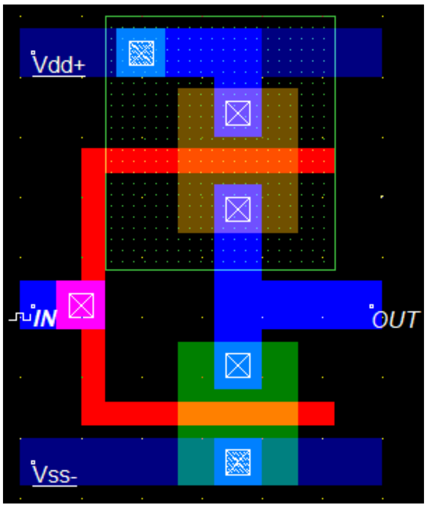


Figure 3.4. Inverter in Microwind35

From the simulation(Figure 3.5), the Inverter works properly.

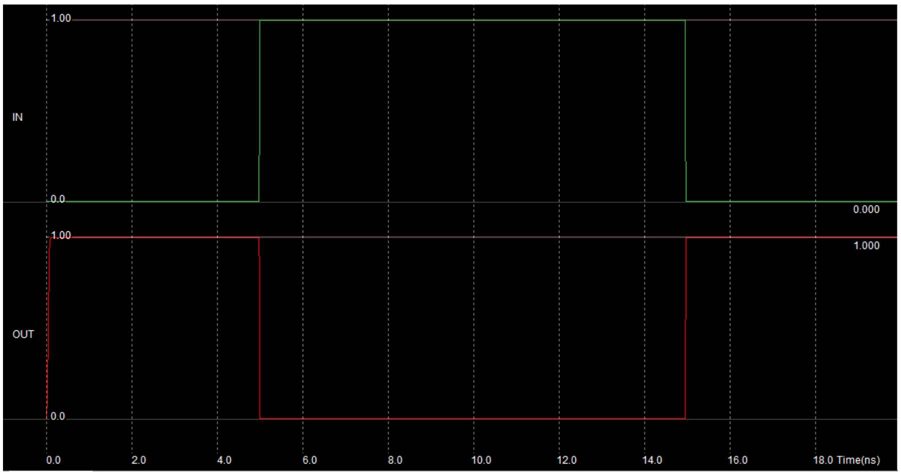


Figure 3.5. Inverter simulation in Microwind35

3.2. The OR Gate

The OR Gate is implemented in the NAND Gate and is made with several nMOS and pMOS.

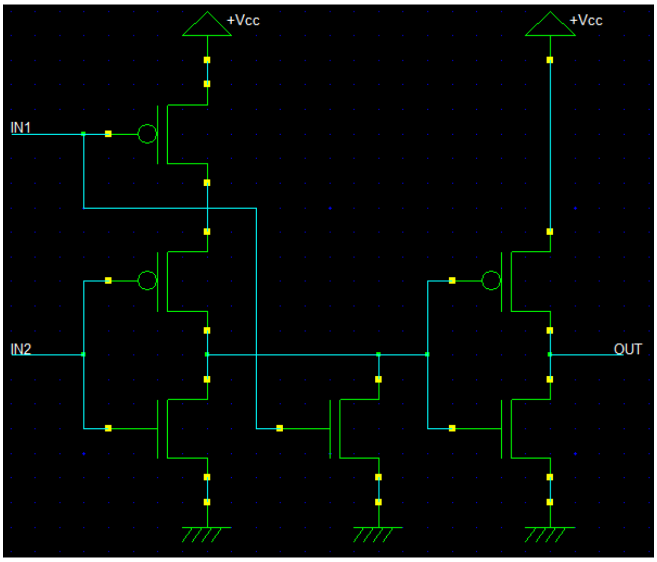


Figure 3.6. OR Gate in Dsch35

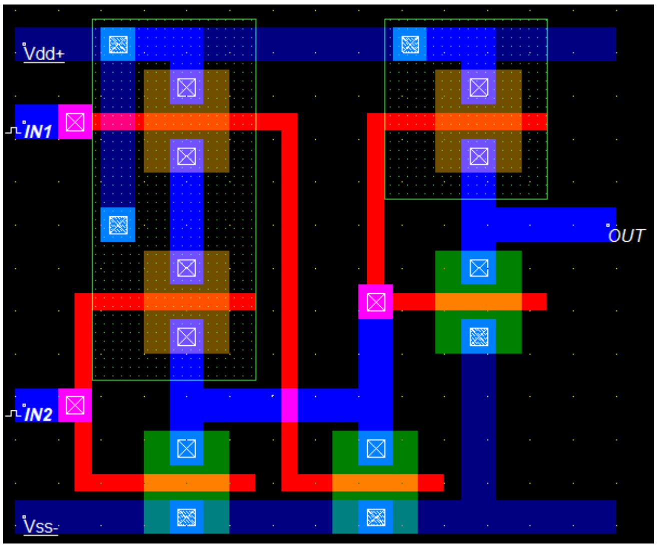


Figure 3.7. OR Gate in Microwind35

From the simulation(Figure 3.8), the OR Gate works as intended.

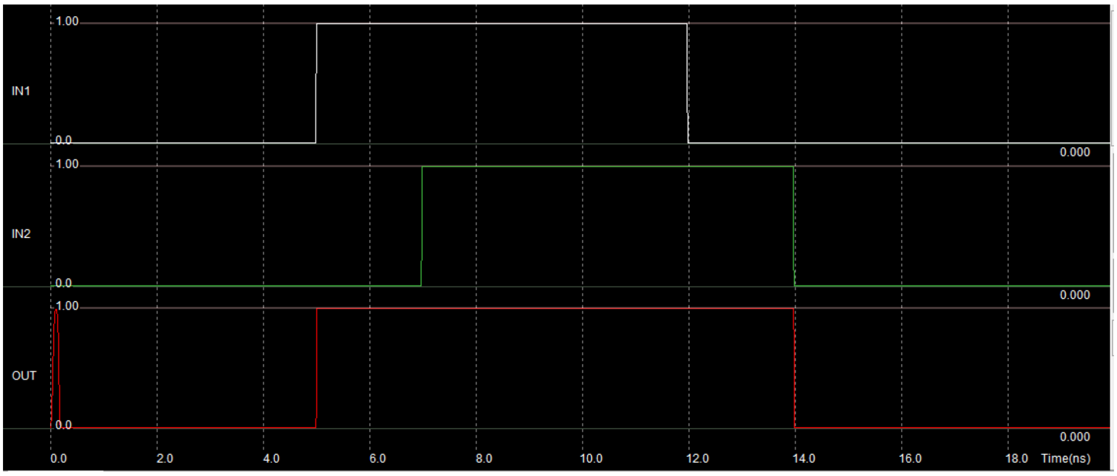


Figure 3.8. OR Gate simulation in Microwind35

3.3. The NAND Gate

The NAND Gate is made by inverting the inputs of an OR Gate.

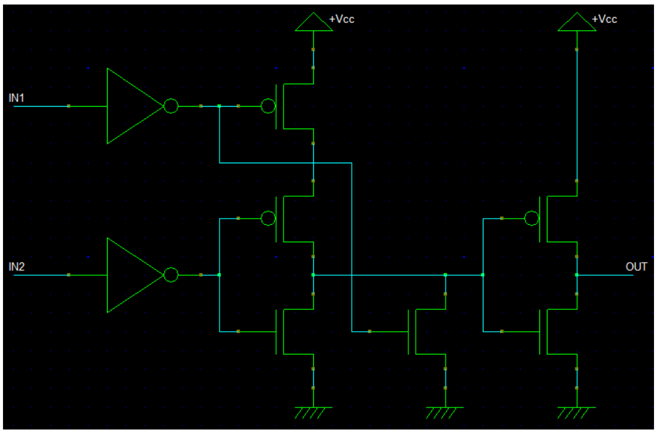


Figure 3.9. NAND Gate in Dsch35

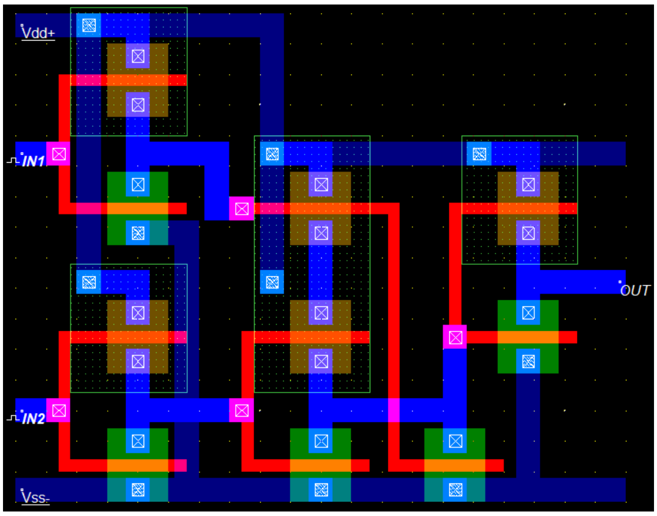


Figure 3.10. NAND Gate in Microwind35

The component is working properly in the simulation(Figure 3.11)

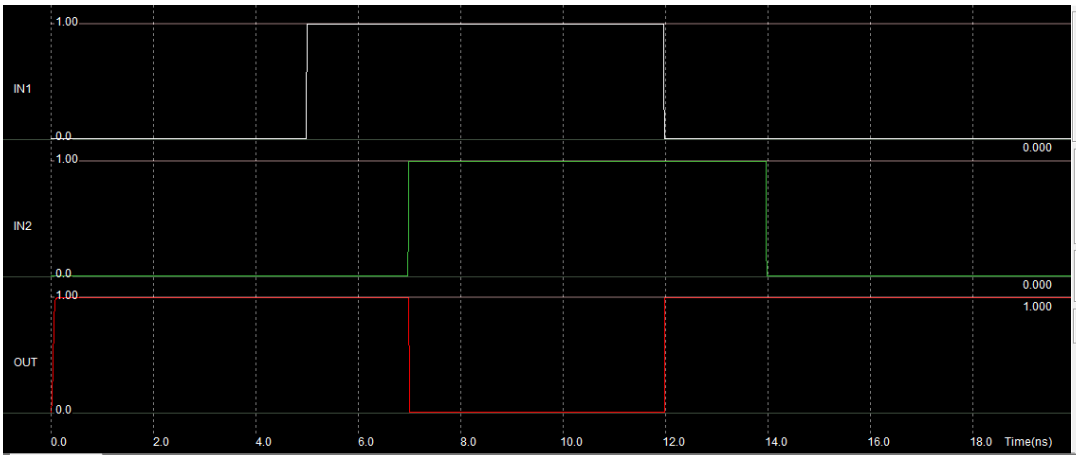


Figure 3.11. NAND Gate simulation in Microwind35

3.4. The AND Gate

The AND Gate is made by placing two NAND Gates in series.

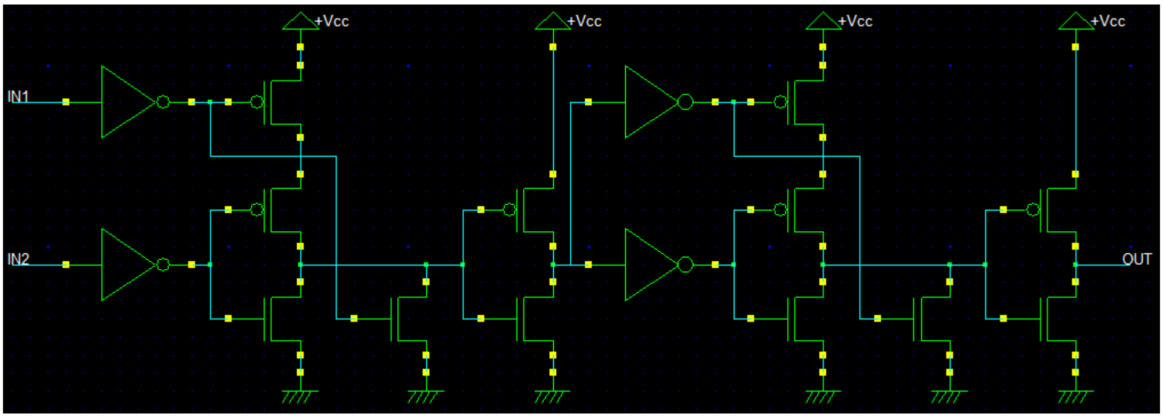


Figure 3.12. AND Gate in Dsch35

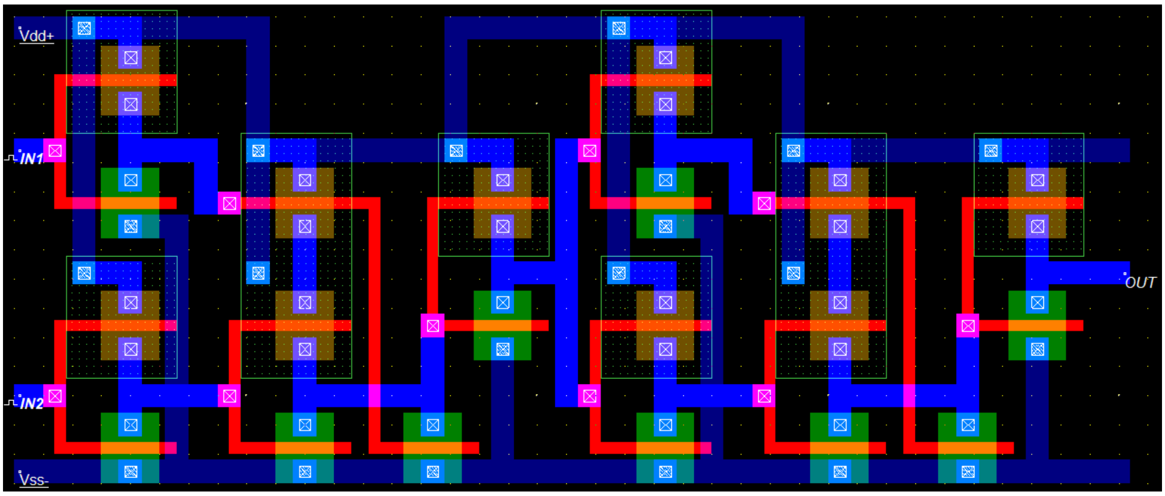


Figure 3.13. AND Gate in Microwind35

From the simulation(Figure 3.14), the AND Gate works as intended.

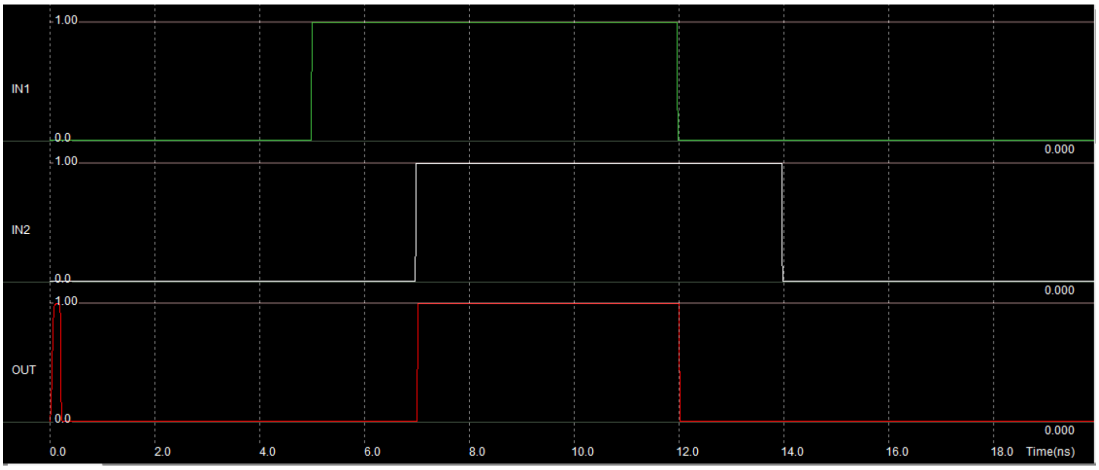


Figure 3.14. AND Gate in Microwind35

3.5. The Oscillator

The design of the oscillator(Figure 3.15) is simple and consists of an odd number of Inverters in series. This particular design of oscillator, is called 'ring oscillator', because the output of the last Inverter is fed back into the input of the first.

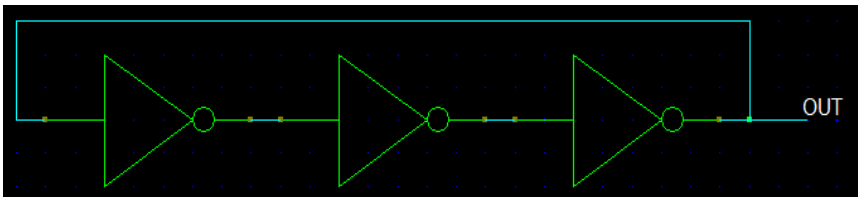


Figure 3.15. Oscillator in Dsch35

The oscillation frequency needs to be slow enough for the counter and for that, five Inverters were used instead of three and also the 'gate' has been widened in order to slow down the response of every Inverter.

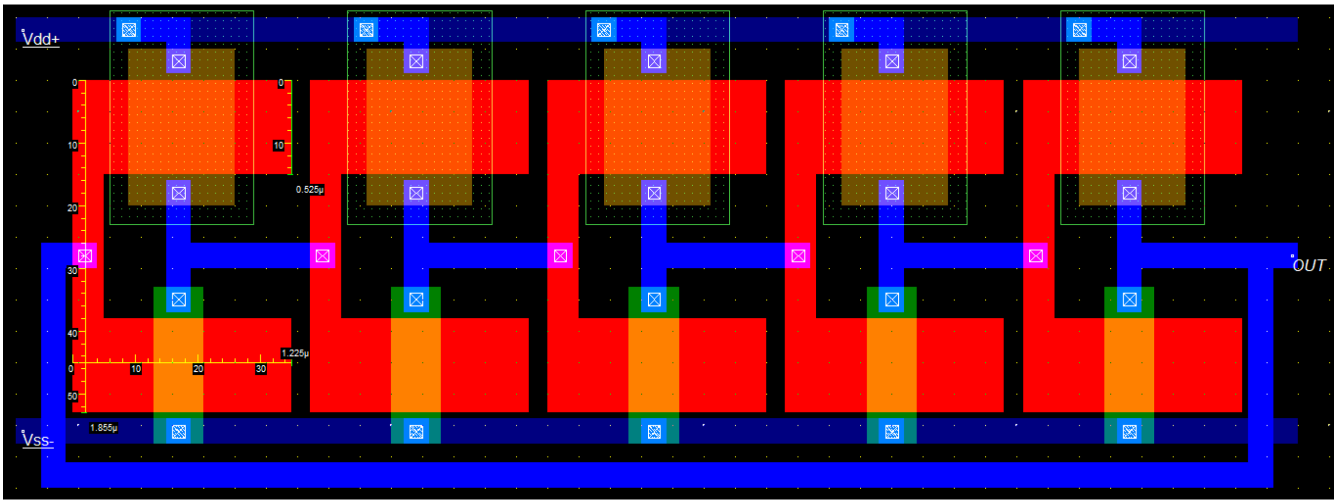


Figure 3.16. Oscillator in Microwind35

From the simulation it is noticeable that the oscillator has a frequency of 300MHZ.

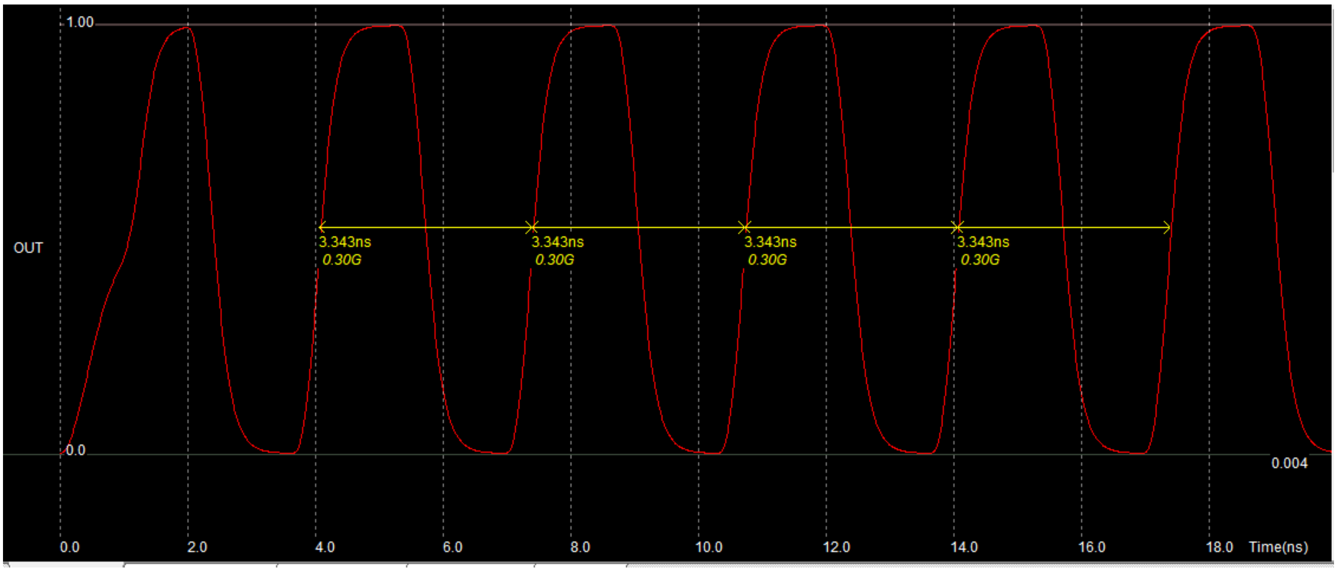


Figure 3.17. Oscillator simulation in Microwind35

3.6. The D-Latch

The D-Latch is made with four NAND Gates(Figure 3.18) and it’s a very important component of the counter, because its main feature is to remember its previous state.

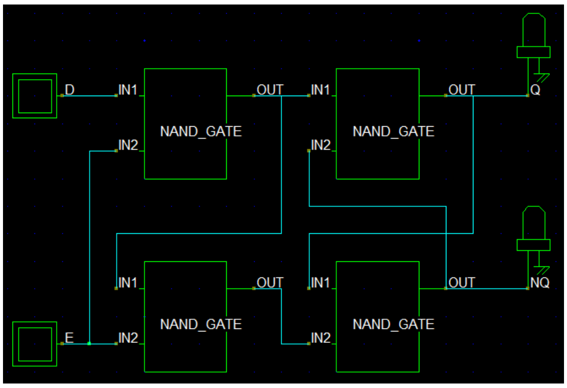


Figure 3.18. D-Latch in Dsch35

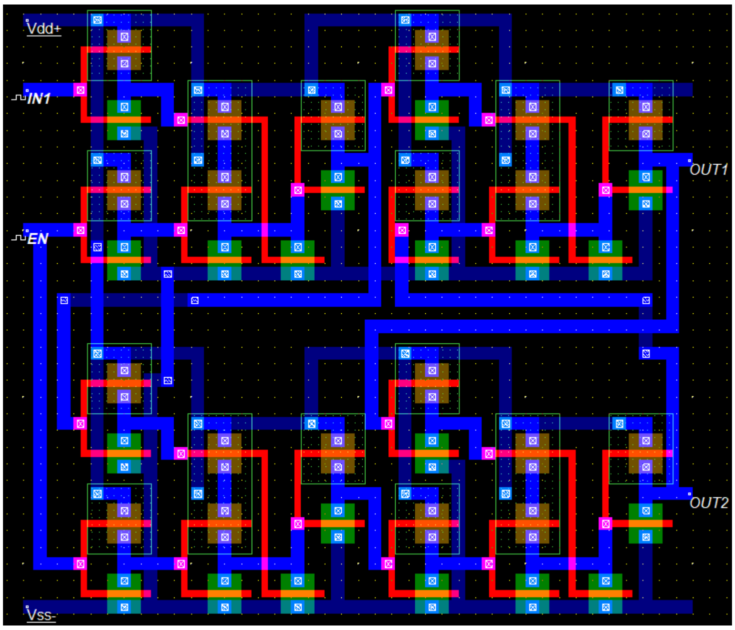


Figure 3.19. D-Latch in Microwind35

In the simulation(Figure 3.20), when the input ‘EN’ is high(1), the non-inverting output ‘OUT1’ copies the input ‘IN1’. The enable input ‘EN’ has to be set ‘high’ to ensure the functioning of the latch, which explains the behavior on both outputs at the very beginning of the simulation.

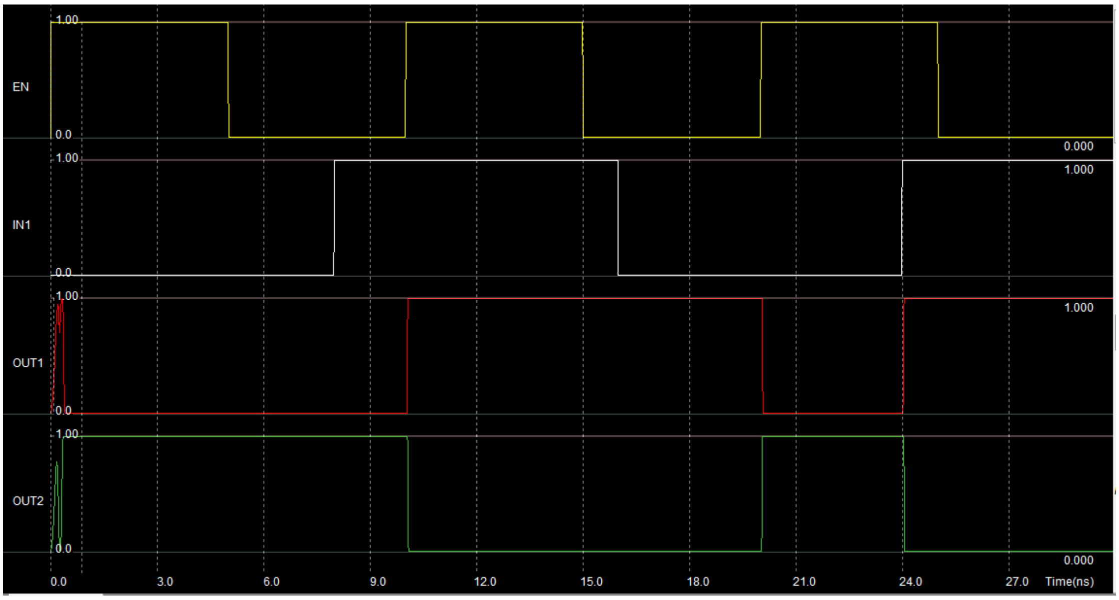


Figure 3.20. D-Latch simulation in Microwind35

3.7. The D-Flip-Flop

The D-Flip-Flop(Figure 3.21) is composed of two D-Latches in series and an Inverter.

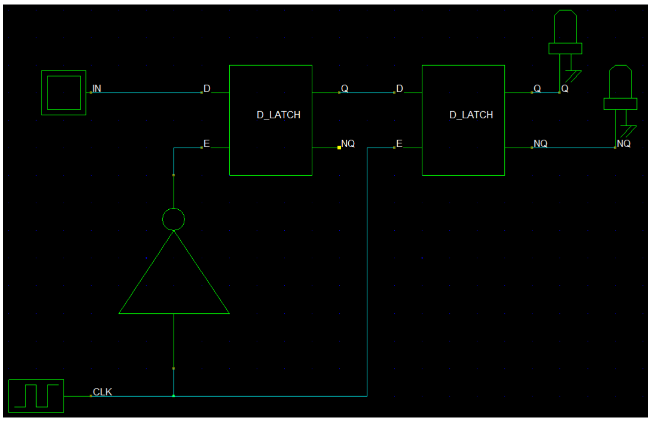


Figure 3.21. D-Flip-Flop in Dsch35

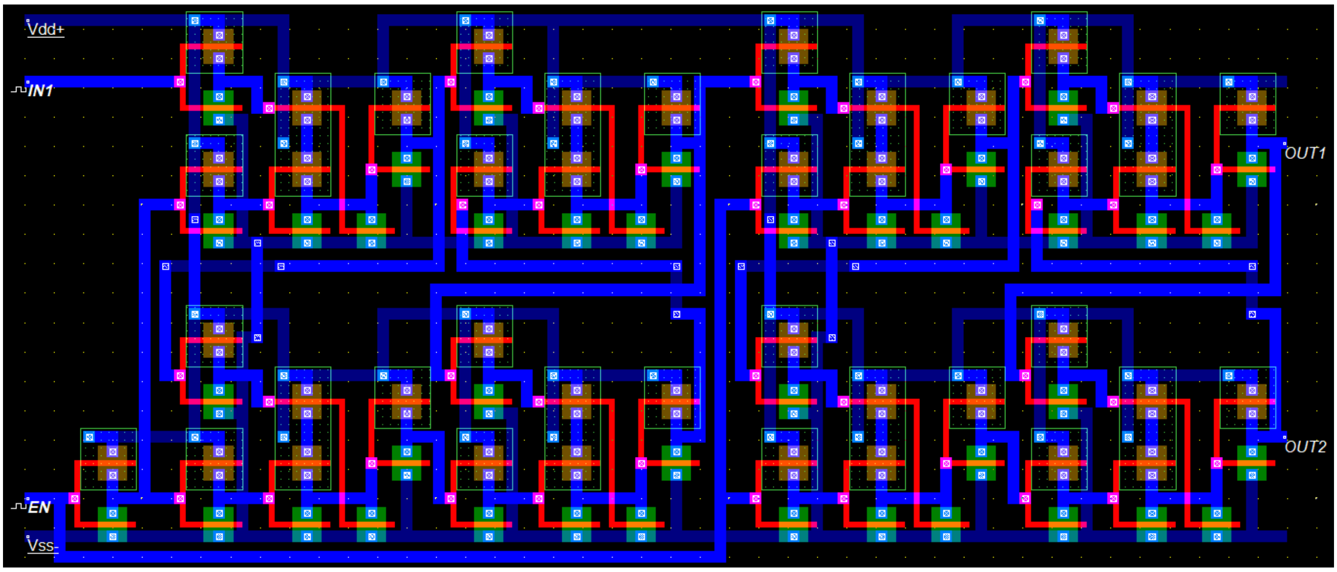


Figure 3.22. D-Flip-Flop in Microwind35

The purpose of such a Flip-Flop is to have the same behavior previously observed for the D-Latch but on the rising edge of the clock.

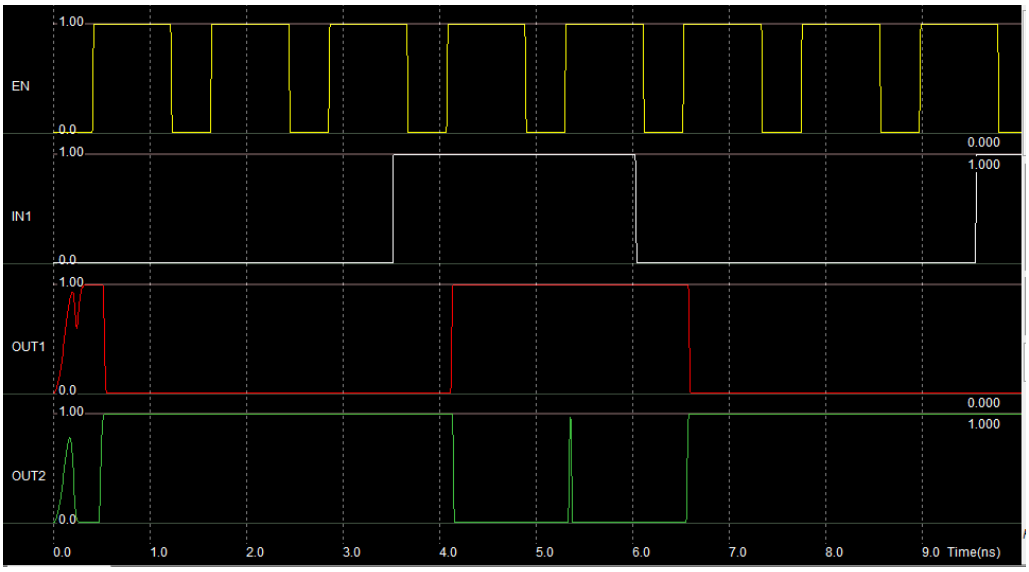


Figure 3.23. D-Flip-Flop simulation in Microwind35

General annotations

There are methods that could be used in order to enhance the efficiency of every component.

- The drain-to-source current IDS is proportional to $\frac{W}{L}$, with 'W' the width of a channel and 'L' its length(distance separating source and drain). Making the MOS channels wider and with a short drain-to-source distance, the switching frequency of the components can be improved.
- Short interconnections can reduce the propagation time.
- Reduction of the MOS surface can decrease the stray capacitance, thus reducing the rise and fall time and increasing the switching frequency.

3.8. Integrated Circuit for Random Number Generation (ICRNG)

The reduction of the surface of the components would also reduce the cost, as less silicon would need to be used to build it. Additionally, by limiting the number of different layers of metal to three, the cost is reduced significantly as every additional layer is extremely expensive.

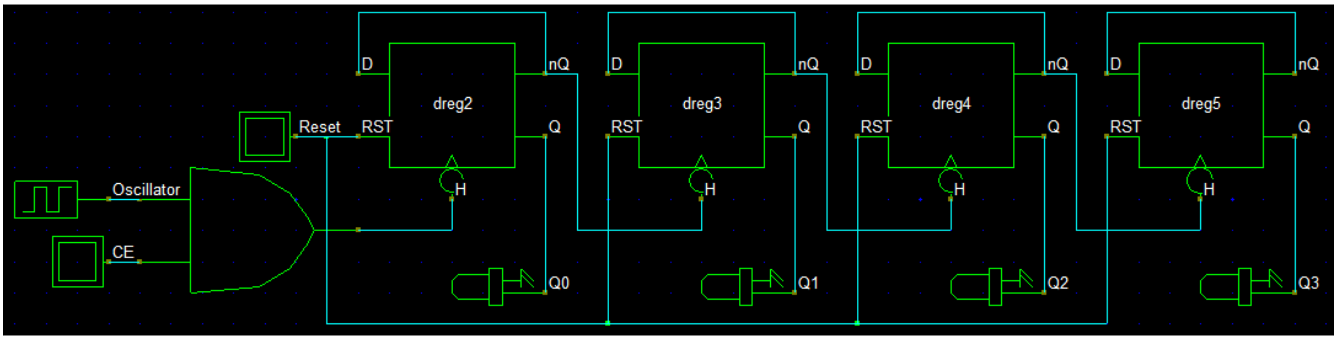


Figure 3.24. Internal schematic of the ICRNG

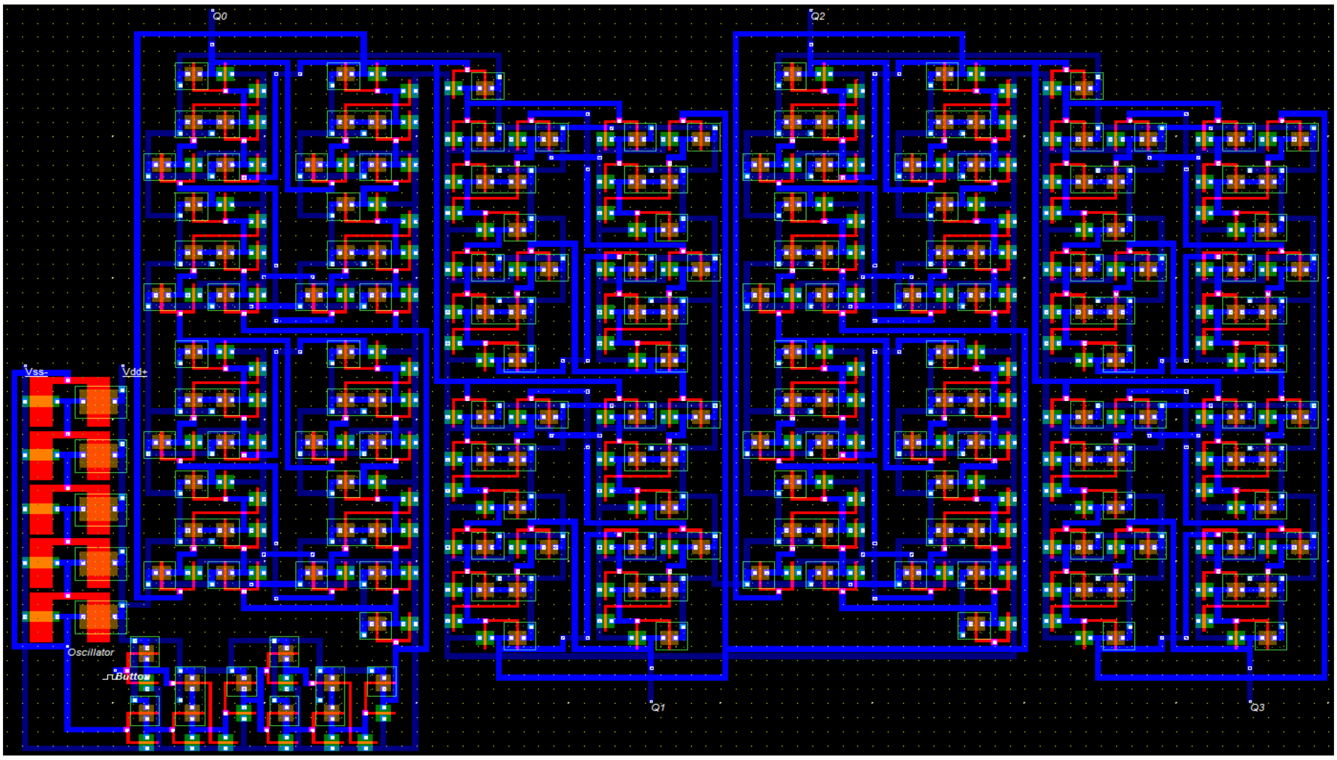
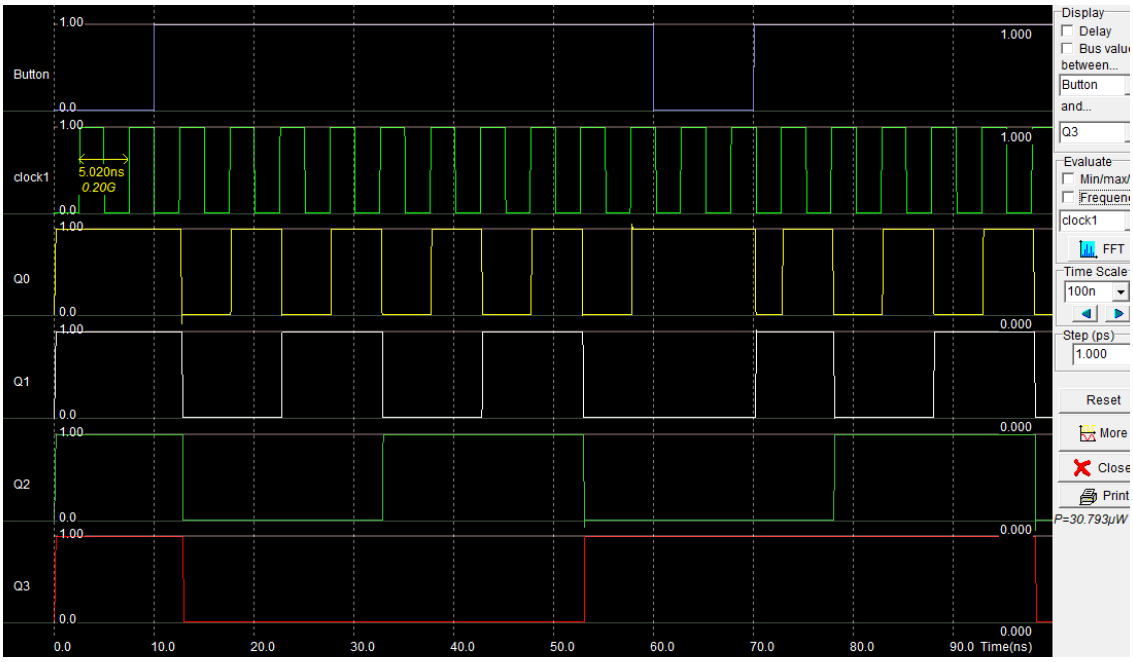


Figure 3.25. ICRNG with all the components in Microwind35

The completed device has quite a small area(width 18.375um and length 30.310um). Additionally it consumes rather low power(40.577uW at 27℃), probably thanks to the organization of the interconnections.

The first simulation(Figure 3.26) of the device, used a software clock, with a frequency comparable to the one of the designed oscillator, to observe the functionality of the device at low frequency(200MHz).



between... Button and... Q3

clock1

Time Scale 100n

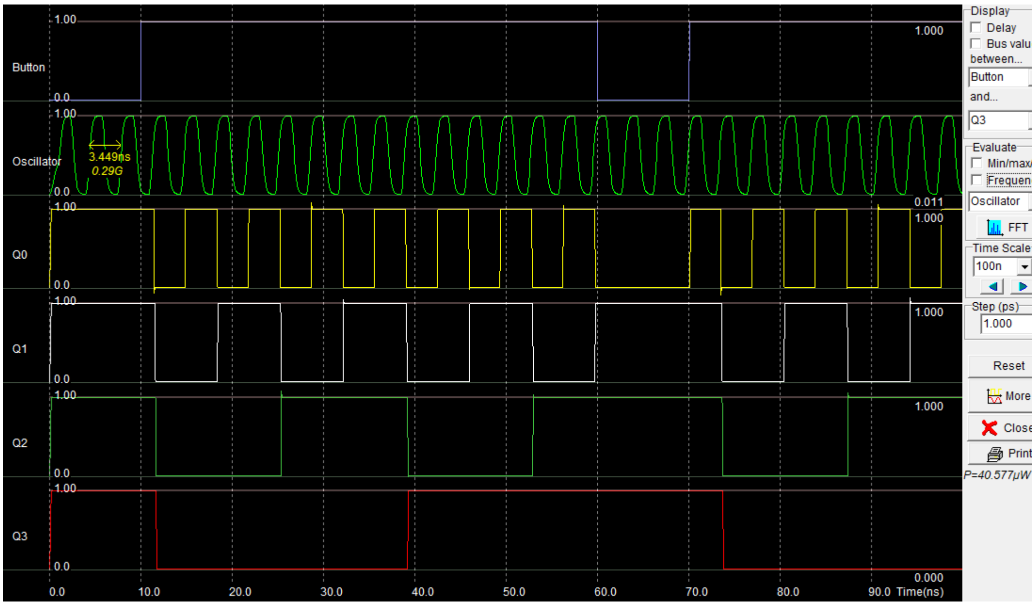
Step (ps) 1.000

P=30.793µW

Figure 3.26. ICRNG simulation using software clock in Microwind35

It is noticeable that the counter works properly and that it counts at every rising edge of the clock, as long as the 'Button' is high(pressed by the user). The maximum frequency of the counter can also be observed, by increasing the software clock, up until the frequency at which the device stops working properly. In this case, that frequency was 500MHz.

In the following simulation(Figure 3.27), the ICRNG works properly with the designed oscillator and with a frequency of 300MHz.



between... Button and... Q3

Oscillator

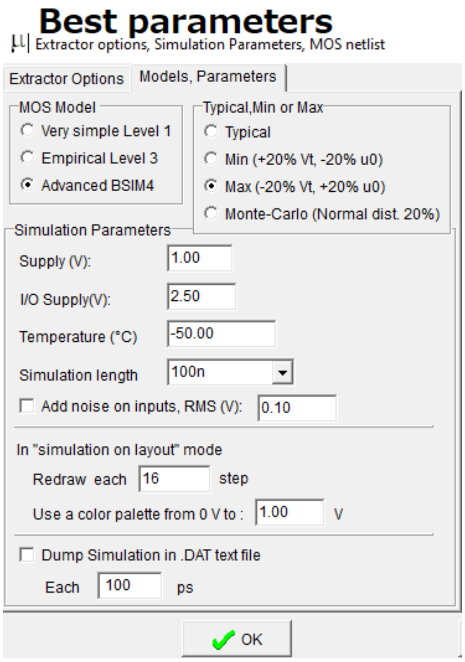
Time Scale 100n

Step (ps) 1.000

P=40.577µW

Figure 3.27. ICRNG simulation using designed oscillator in Microwind35

Lastly, the device was simulated to a few extreme situations(scenarios), to observe its range of operation.



Best parameters

Extractor options, Simulation Parameters, MOS netlist

MOS Model: ○ Very simple Level 1 ○ Empirical Level 3 ● Advanced BSIM4

Typical,Min or Max: ○ Typical ○ Min (+20% Vt, -20% u0) ● Max (-20% Vt, +20% u0) ○ Monte-Carlo (Normal dist. 20%)

Simulation Parameters

Supply (V): 1.00

I/O Supply(V): 2.50

Temperature (°C): -50.00

Simulation length: 100n

□ Add noise on inputs, RMS (V): 0.10

In "simulation on layout" mode

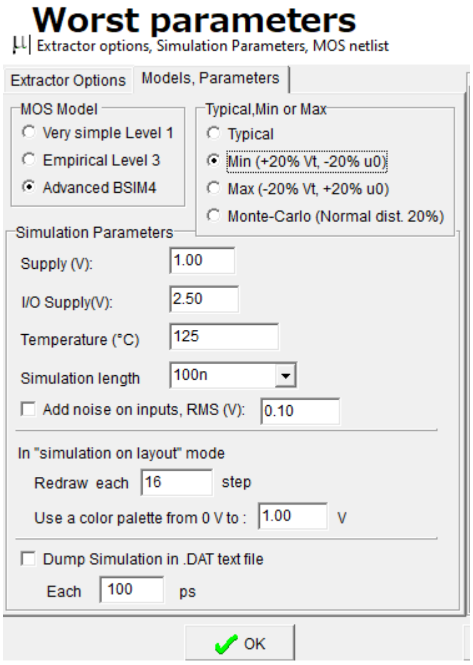
Redraw each 16 step

Use a color palette from 0 V to : 1.00 V

□ Dump Simulation in .DAT text file

Each 100 ps

Figure 3.28. Best case scenario parameters



Worst parameters

Extractor options, Simulation Parameters, MOS netlist

MOS Model: ○ Very simple Level 1 ○ Empirical Level 3 ● Advanced BSIM4

Typical,Min or Max: ○ Typical ● Min (+20% Vt, -20% u0) ○ Max (-20% Vt, +20% u0) ○ Monte-Carlo (Normal dist. 20%)

Simulation Parameters

Supply (V): 1.00

I/O Supply(V): 2.50

Temperature (°C): 125

Simulation length: 100n

□ Add noise on inputs, RMS (V): 0.10

In "simulation on layout" mode

Redraw each 16 step

Use a color palette from 0 V to : 1.00 V

□ Dump Simulation in .DAT text file

Each 100 ps

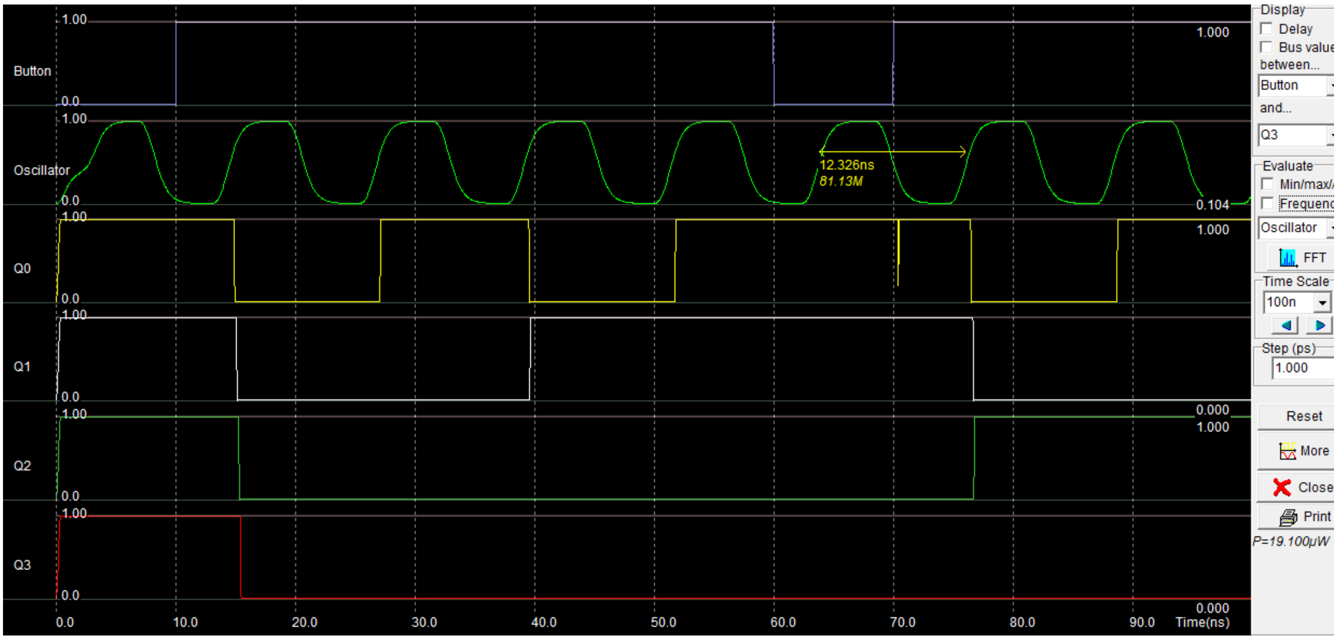
Figure 3.29. Worst case scenario parameters

Thinner oxide layer, good supply and low temperature give very fast switching frequency in every MOS of the device, which explains such good performances in the tables below. The consumption power of a C-MOS based circuit is proportional to the supply voltage and the circuit frequency. $P \times a \times Vsupply^2 \times f$

	Worst parameters (125℃)	Best parameters (-50℃)
Power consumption (W)	17.084u	109u
Counting frequency (Hz)	110M	680M

	Worst parameters (200℃)	Best parameters (-238℃)
Power consumption (W)	19.100u	921u
Counting frequency (Hz)	81.13M	10.34G

The simulation(Figure 3.30) of the device at an extremely high temperature.



Display: □ Delay □ Bus value between... Button and... Q3

Evaluate: □ Min/max □ Frequency Oscillator

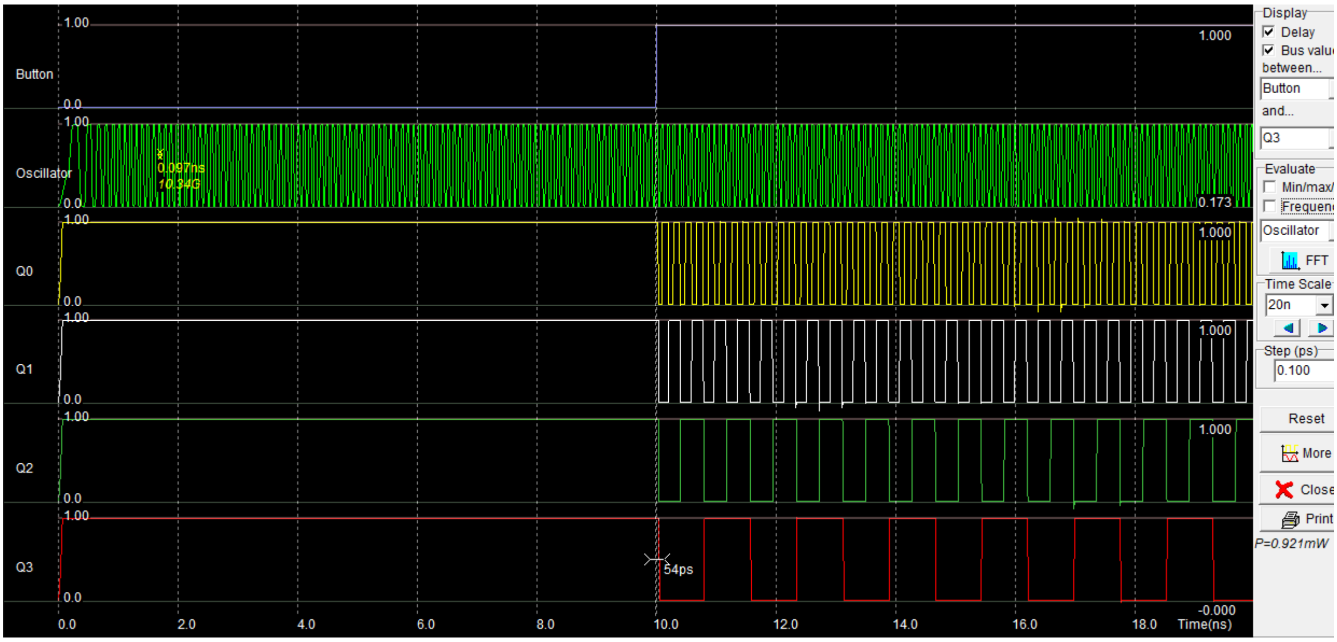
Time Scale: 100n

Step (ps): 1.000

P=19.100µW

Figure 3.30. ICRNG simulation at 200℃ in Microwind35

The simulation(Figure 3.31) of the device at an extremely low temperature.



Display: ☑ Delay ☑ Bus value between... Button and... Q3

Evaluate: □ Min/max □ Frequency Oscillator

Time Scale: 20n

Step (ps): 0.100

P=0.921mW

Figure 3.31. ICRNG simulation at -238℃ in Microwind35

From the observations of the simulations above, the results are satisfying as the range of operation is quite large.

3.9. Improvements of the ICRNG

- **Asynchronous or synchronous counter:** The period of the oscillator must be shorter than the duration of the user signal(Button). Considering that the operator using the device will keep his finger at least 0.01s on the button, a slow clock(300MHz/period: 3.3ns) allows good results. Although a synchronous counter can be significantly faster than an asynchronous one, its design is more complicated, as it needs more components and therefore is more expensive and consumes more energy. In the end the choice of an asynchronous counter was a good one, since the speed is satisfying.
- **Big number generator:** The best option would be to use several different ICRNGs. Also, it is known for a fact that each silicon etching is unique and every ICRNG has its own oscillation frequency. If two ICRNGs were to be placed in parallel, a perfect 8-Bit random number could be accomplished.

4. Program Counter and Instructions Decoder

This chapter features the designs and simulations of four components which are going to be part of an integrated circuit. The designs were made with 45nm(cmos45n.rul) technology. The components are; a 4-Bit Adder, a 16-Bit Counter, an 8-Bit Program Counter and an Instructions Decoder. Additionally their analysis will include three different cases; typical, best and worst-case scenarios.

The program counter and the instructions decoder are circuits that work closely with the memory. While the decoder reads the instructions from the memory and gives the enabling pulse to the other components(the ALU, inputs and outputs and RF circuit), the counter will work as a clock for the entire integrated circuit.

The first component that needs to be designed is the Full Adder, which is essential for the design of the 4-Bit Adder. The Full Adder(Figure 4.1) consists of two XOR Gates with three different inputs ‘A’, ‘B’, ‘C’ to give the output ‘SUM’ and four NAND Gates to give the output ‘CARRY’.

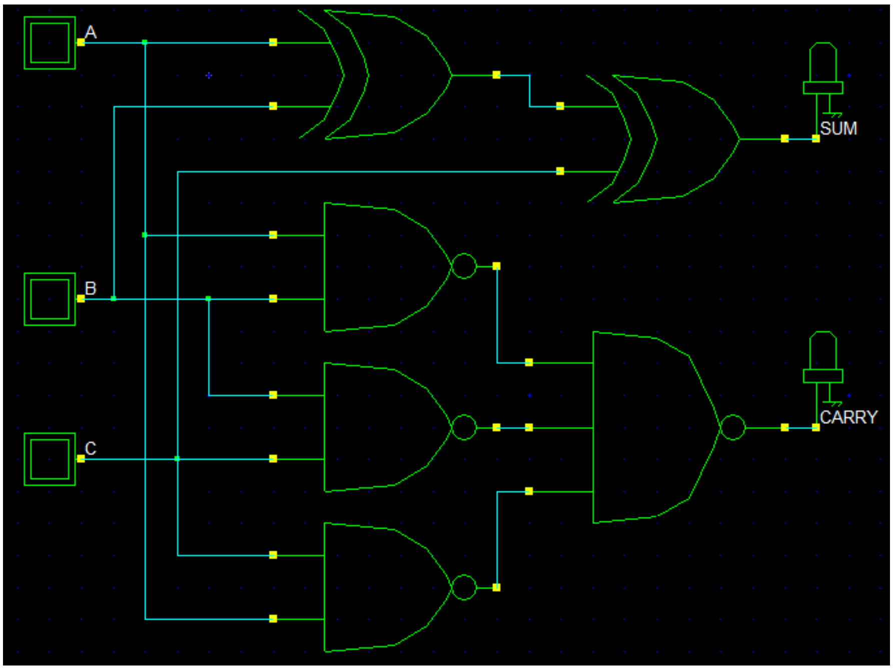


Figure 4.1. Full Adder in Dsch35

4.1. The NAND & XOR Gates

First are the designs of two NAND Gates with two and three inputs respectively and an XOR Gate with two inputs. It is best to begin by designing the NAND Gate(Figure 4.2), because it’s a versatile component which, in different combinations, can be used to make every other Logic Gate(Figure 4.3). Below, is the design of the NAND Gate with two inputs and its simulation.

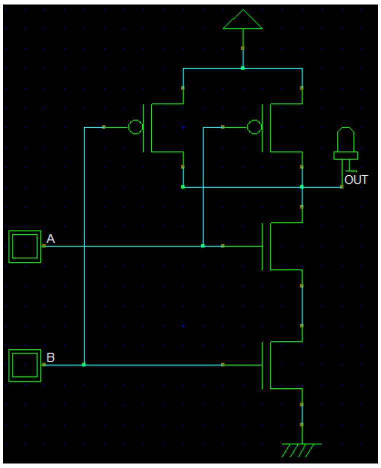


Figure 4.2. Two inputs NAND Gate in Dsch35

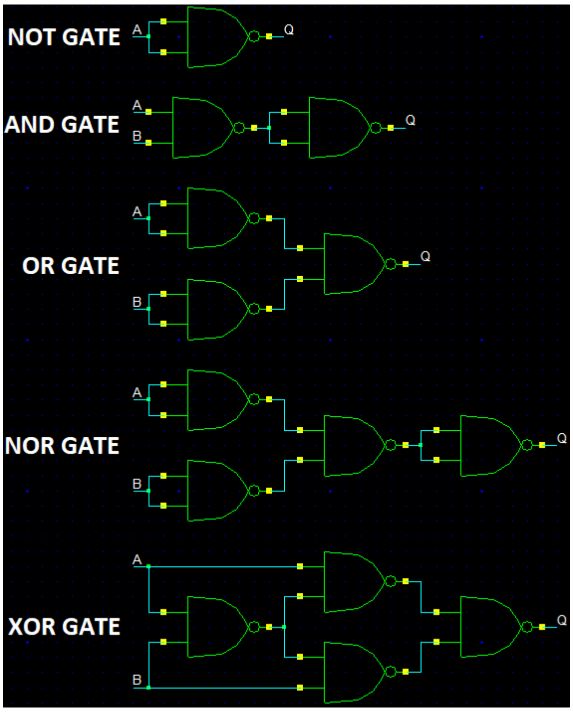


Figure 4.3. Logic gates made using only NAND Gates

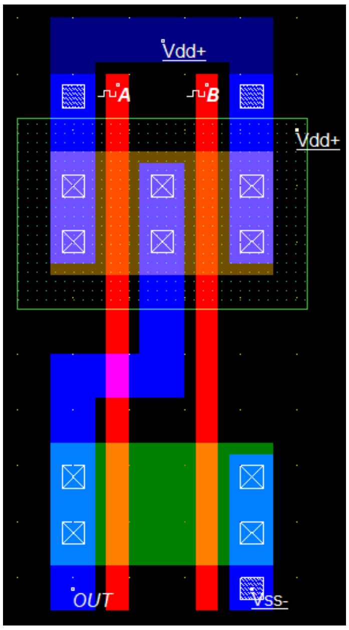


Figure 4.4. Two inputs NAND Gate in Microwind35

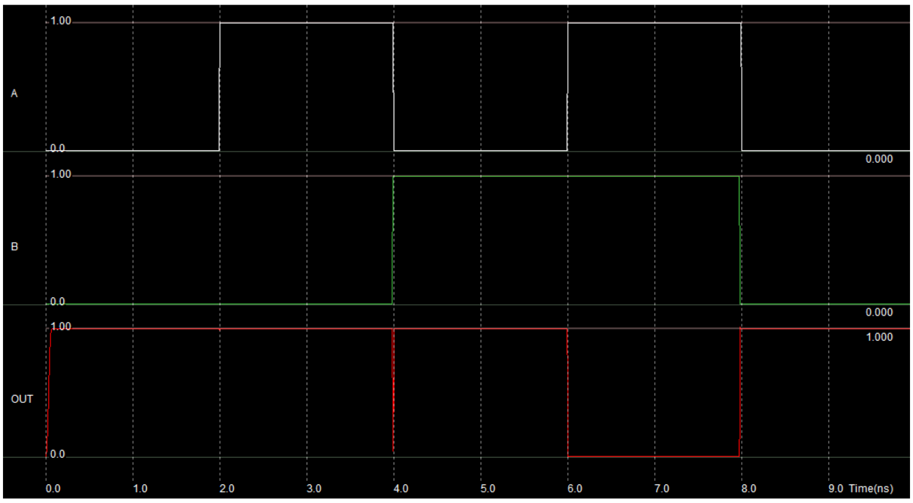


Figure 4.5. Two inputs NAND Gate simulation in Microwind35

The NAND Gate with three inputs is the same as with the two, with the addition of another couple of p-MOS and n-MOS, parallel to the other four. The design of the XOR Gate(Figure 4.8) is based on Figure 4.3.

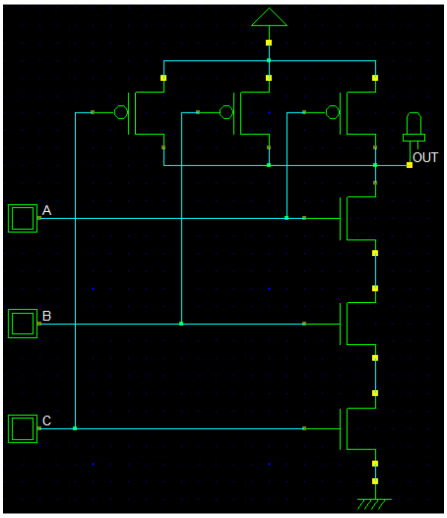


Figure 4.6. Three inputs NAND Gate in Dsch35

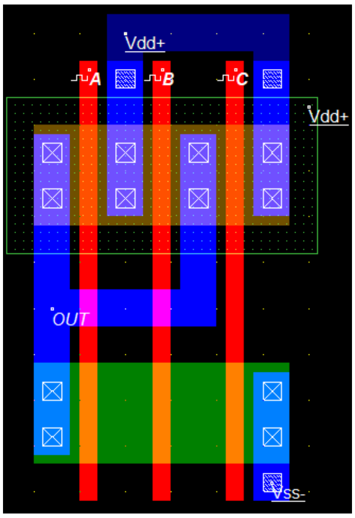


Figure 4.7. Three inputs NAND Gate in Microwind35

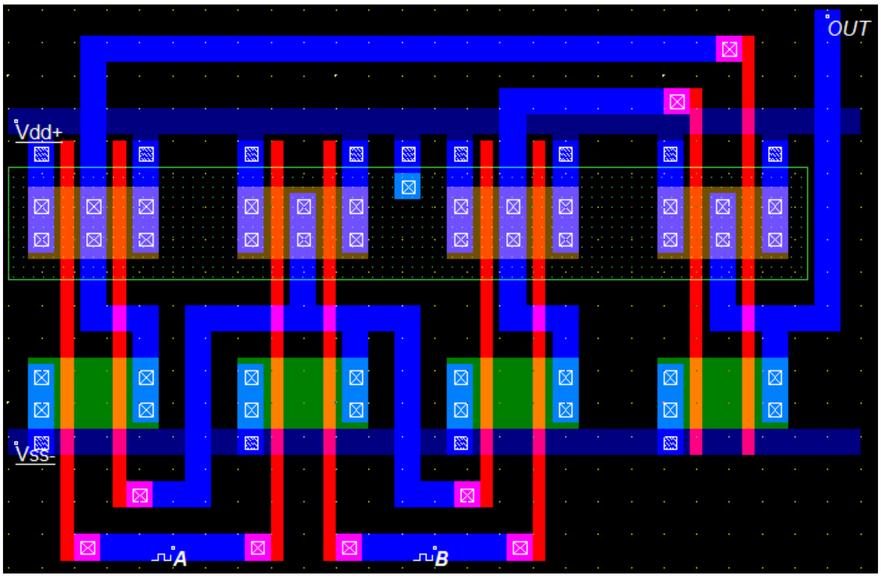


Figure 4.8. XOR Gate in Microwind35

From the simulation(Figure 4.9) of the XOR Gate, it seems to be working properly. The XOR Gate gives high output(logic ‘1’), only when the inputs are different.

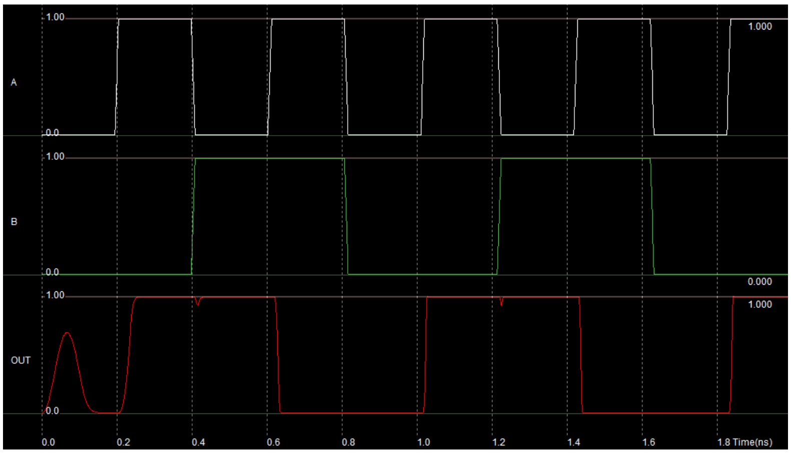


Figure 4.9. XOR Gate simulation in Microwind35

4.2. The Full Adder

Now that all the essential Gates are designed, it is time to design the Full Adder(Figure 4.10).

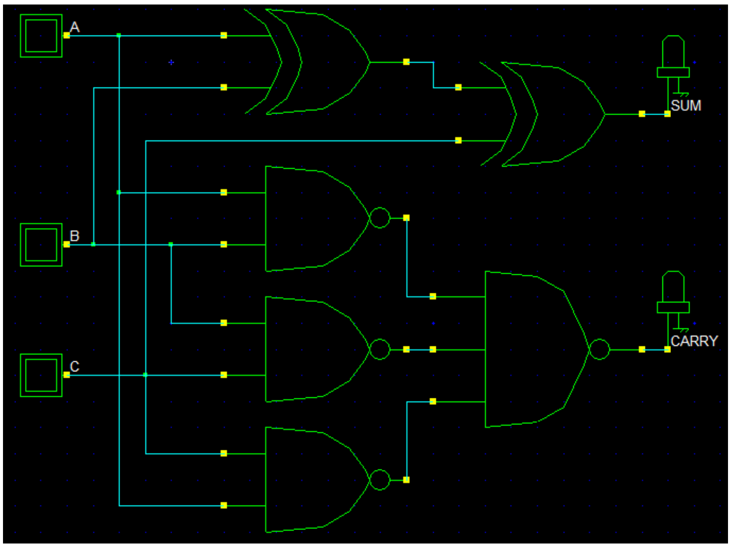


Figure 4.10. Full Adder in Dsch35

From the simulation(Figure 4.12), it is noticeable that the output 'SUM' is high when only one or all inputs are high and that the output 'CARRY' is high when two or all inputs are high. When the output 'CARRY' is high, it means that the resulting number, from the addition of the three input numbers, needs more bits than the circuit can show. We need more Full Adders working together in order to show bigger number additions.

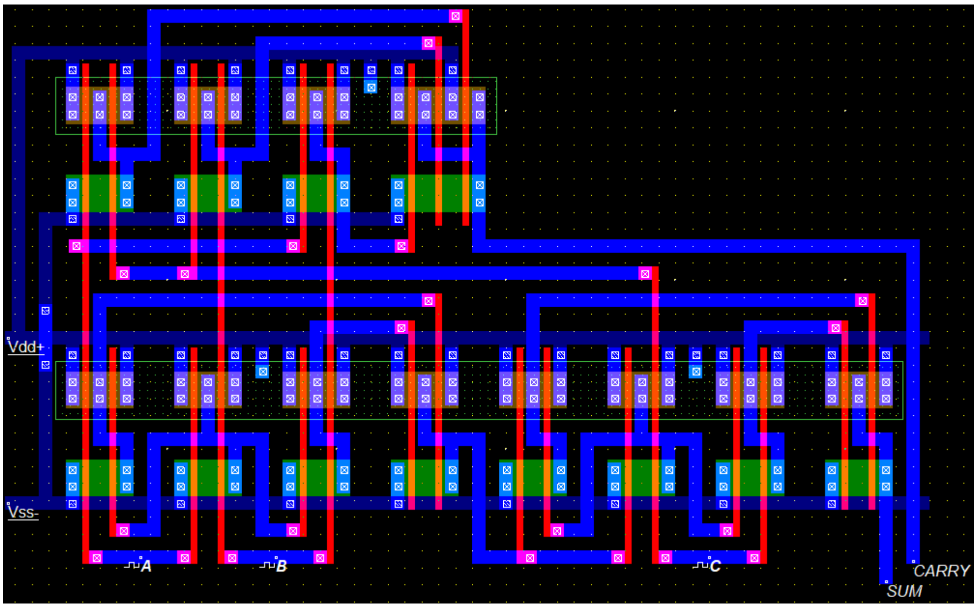


Figure 4.11. Full Adder in Microwind35

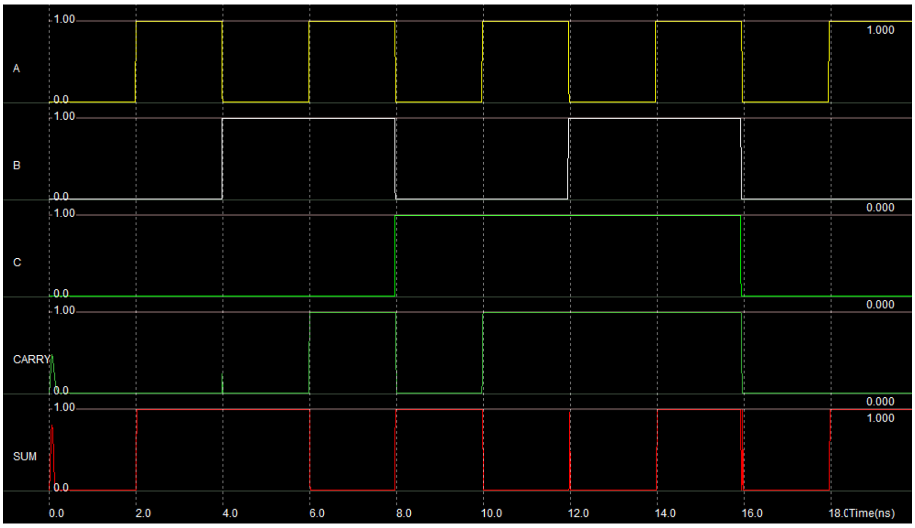


Figure 4.12. Full Adder simulation in Microwind35

4.3. The 4-Bit Adder

The 4-Bit Adder schematic consists of four Full Adders. Each ‘A’ and ‘B’ input represents a bit from the hexadecimal number A and B respectively. Each ‘CARRY’ output signal connects with the ‘C’ input of the next Full Adder. The ‘SUM’ output of the first Full Adder connects to an AND Gate which is connected at the other pin to ‘high’ voltage. The logic result connects to the next AND Gate with the ‘SUM’ output of the next Full Adder and the cycle continues through all the following Full Adders of the design.

Through the simulation, it is clear that the result of the addition of the two hexadecimal numbers is shown at the ‘SUM’ outputs. The ‘SUM’ output of the first Full Adder represents the least significant bit and the ‘SUM’ output of the last Full Adder represents the most significant bit. In Figure 4.14 is the design of the AND Gate used in the design of the 4-Bit Adder(Figure 4.13). The AND Gate consists of a NAND Gate which result goes through a NOT Gate to do the ‘NOT(NAND(A,B))’ operation.

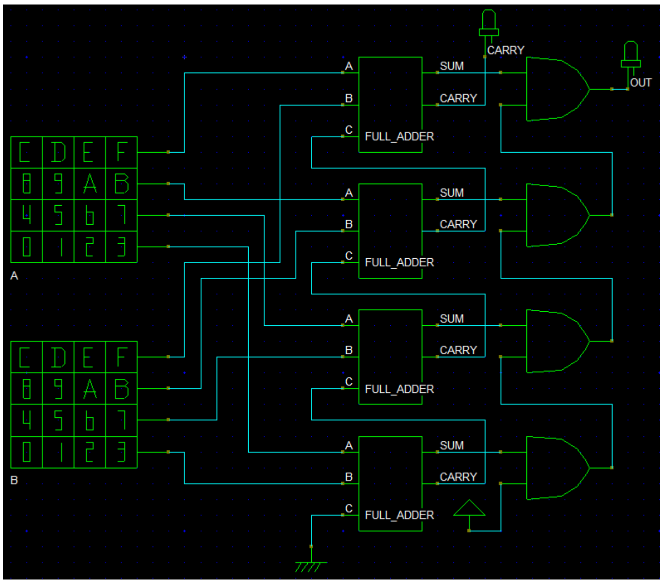


Figure 4.13. 4-Bit Adder in Dsch35

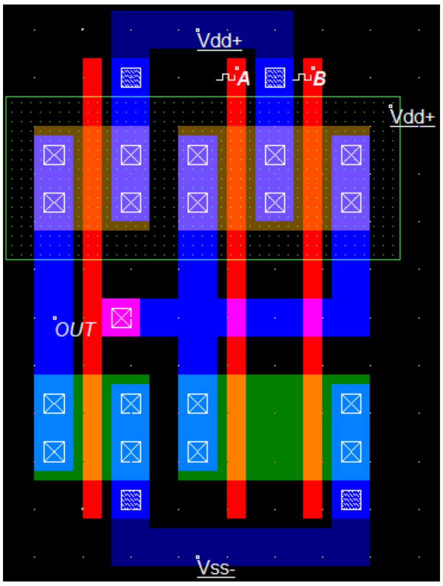
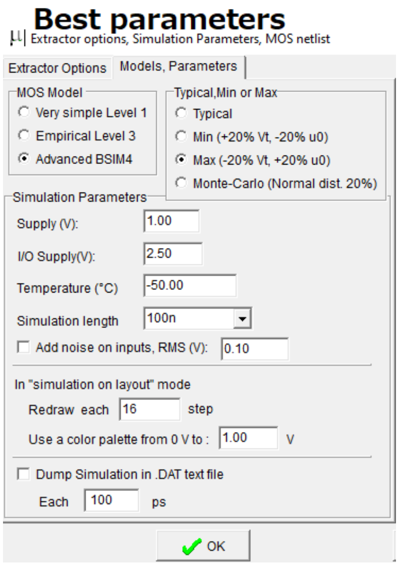


Figure 4.14. AND Gate in Microwind35

It is also important to note that the parameters below were used, in the following simulations, for the best and worst case scenarios.



Best parameters

Extractor options, Simulation Parameters, MOS netlist

Extractor Options | Models, Parameters

MOS Model
○ Very simple Level 1
○ Empirical Level 3
● Advanced BSIM4

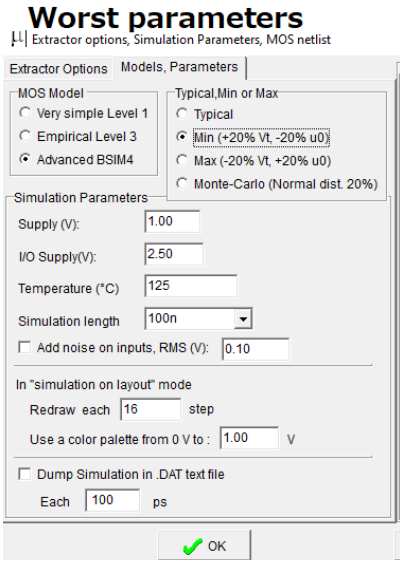
Typical,Min or Max
○ Typical
○ Min (+20% Vt, -20% u0)
● Max (-20% Vt, +20% u0)
○ Monte-Carlo (Normal dist. 20%)

Simulation Parameters
Supply (V): 1.00
I/O Supply(V): 2.50
Temperature (°C) -50.00
Simulation length 100n
□ Add noise on inputs, RMS (V): 0.10

In "simulation on layout" mode
Redraw each 16 step
Use a color palette from 0 V to : 1.00 V

□ Dump Simulation in .DAT text file
Each 100 ps

Figure 4.15. Best case scenario parameters



Worst parameters

Extractor options, Simulation Parameters, MOS netlist

Extractor Options | Models, Parameters

MOS Model
○ Very simple Level 1
○ Empirical Level 3
● Advanced BSIM4

Typical,Min or Max
○ Typical
● Min (+20% Vt, -20% u0)
○ Max (-20% Vt, +20% u0)
○ Monte-Carlo (Normal dist. 20%)

Simulation Parameters
Supply (V): 1.00
I/O Supply(V): 2.50
Temperature (°C) 125
Simulation length 100n
□ Add noise on inputs, RMS (V): 0.10

In "simulation on layout" mode
Redraw each 16 step
Use a color palette from 0 V to : 1.00 V

□ Dump Simulation in .DAT text file
Each 100 ps

Figure 4.16. Worst case scenario parameters

The Figure 4.17. depicts the design of the 4-Bit Adder in the Microwind program which is made using the design of the Full Adder(Figure 4.11.).

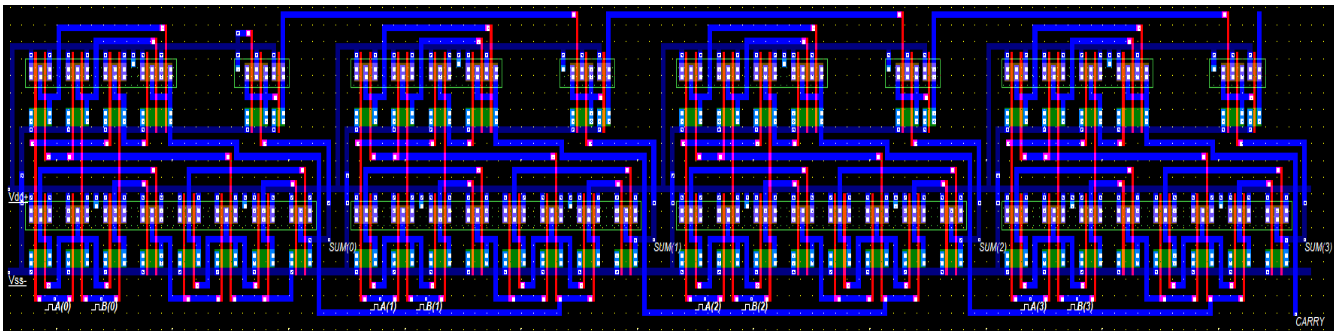


Figure 4.17. 4-Bit Adder in Microwind35

Below is the simulation of the 4-Bit Adder component in typical operating conditions(27℃). The result shows a delay of 0.11ns and frequency of 5GHz.

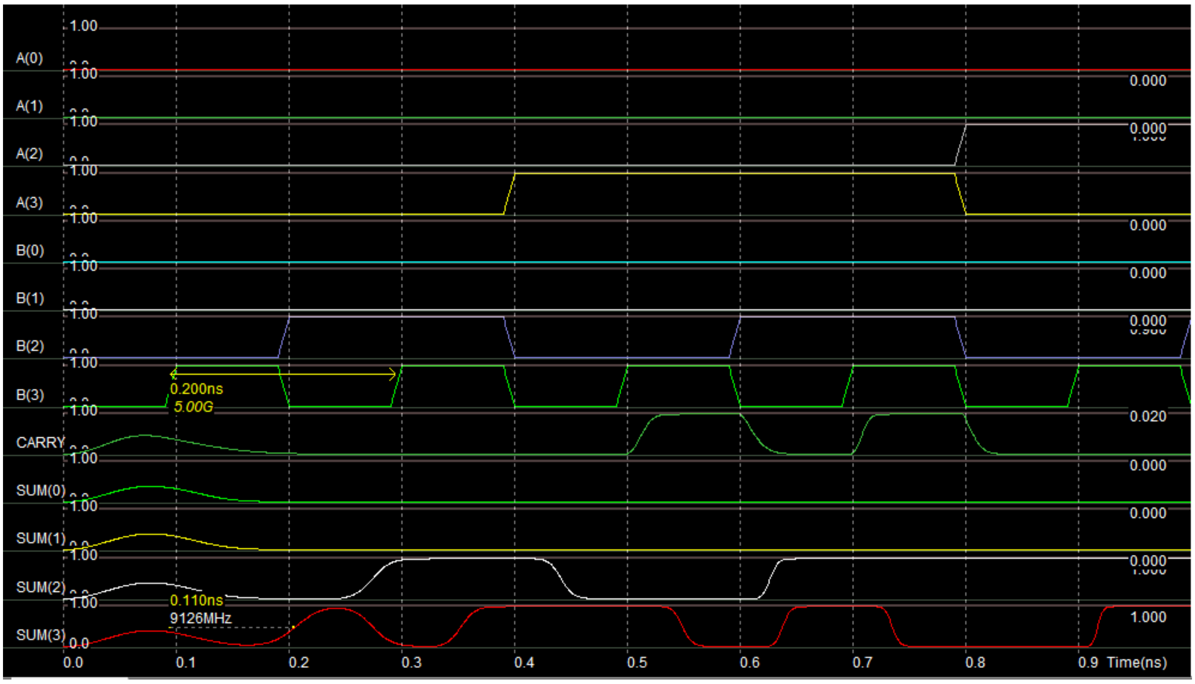


Figure 4.18. 4-Bit Adder simulation in Microwind35

The results from the simulation of the worst and best case scenarios are shown in the next table. The counting frequency didn't exhibit any difference in the various scenarios, but the delay is almost doubled in the worst case in contrast to the best case scenario, but is still a tolerant delay. Additionally, the component stops working properly for voltages lower than 0.5V.

	Typical parameters (27℃)	Worst parameters (125℃)	Best parameters (-50℃)
Delay (s)	0.11n	0.154n	0.079n
Counting frequency (Hz)	5G	5G	5G

4.4. The 16-Bit Counter

The design of the 16-Bit Counter is the same as the design of the 4-Stage Binary Counter in chapter 2.4(page 15). However, in this chapter the design uses only NAND Gates to accomplish the same task. The schematics below depict the design of the 16-Bit Counter in Dsch and in Microwind respectively. It follows the same design steps as the 4-Stage Binary Counter and is made with four D-Registers in series.

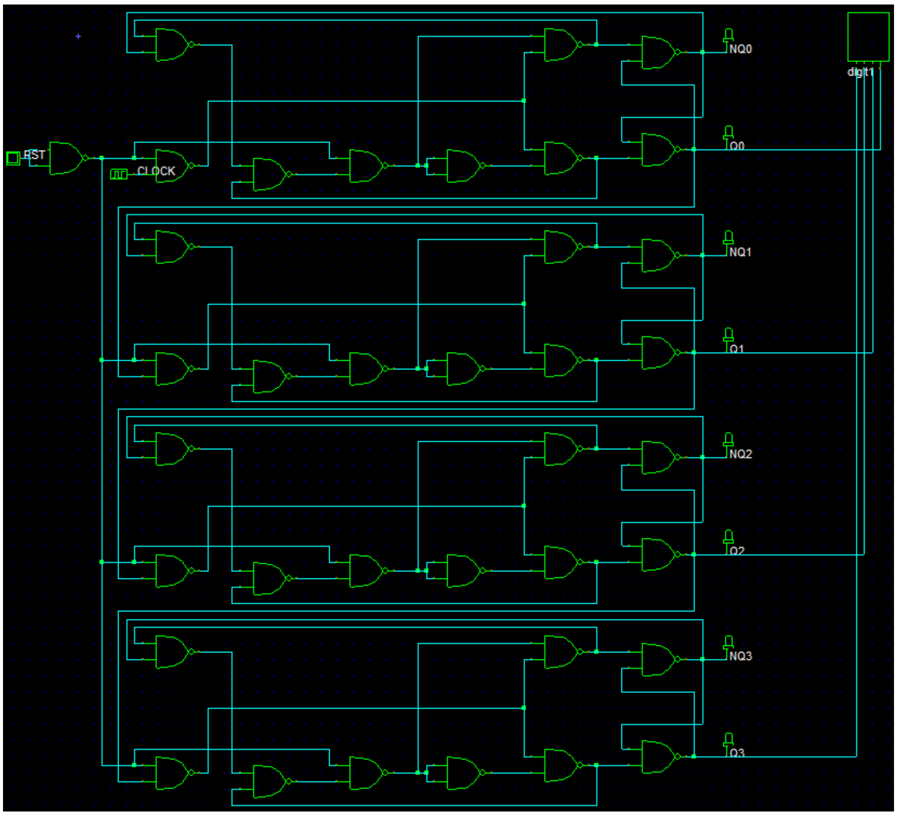


Figure 4.19. 16-Bit Counter in Dsch35

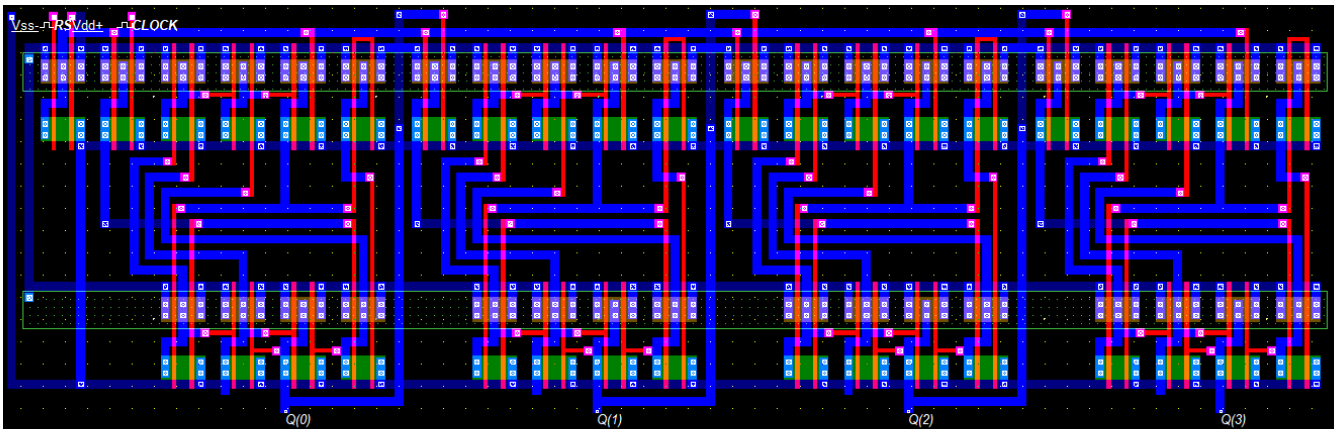


Figure 4.20. 16-Bit Counter in Microwind35

The simulation(Figure 4.21) of the 16-Bit Counter is similar to the one of the 4-Stage Binary Counter(Figure 2.18). The only difference is the 'RST' signal, which is implemented in this design, in order to allow the user to reset the counting from the beginning.

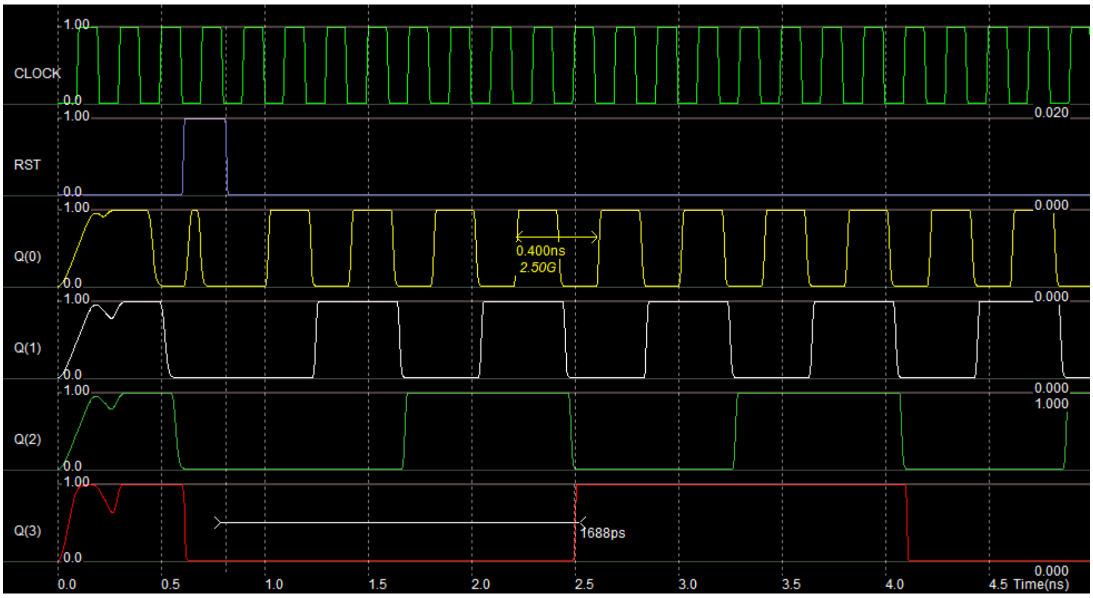


Figure 4.21. 16-Bit Counter simulation in Microwind35

In the different simulation scenarios there wasn't any major difference in the counting frequency and the variation in time delay is insignificant. The device stops working properly for voltages lower than 0.4V.

	Typical parameters (27℃)	Worst parameters (125℃)	Best parameters (-50℃)
Delay (s)	1.688n	1.780n	1.645n
Counting frequency (Hz)	2.5G	2.5G	2.5G

4.5. The 8-Bit Counter (Program Counter)

The schematic for the 8-Bit Counter is the same as the one of the 16-Bit Counter and the only difference is that it has only three D-Registers instead of four. The 8-Bit Counter is going to be used as a Program Counter for the whole device and its components. Its task is to signal the other components of the integrated circuit, but prior to this the Program Counter gets its instruction/commands from the Instructions Decoder.

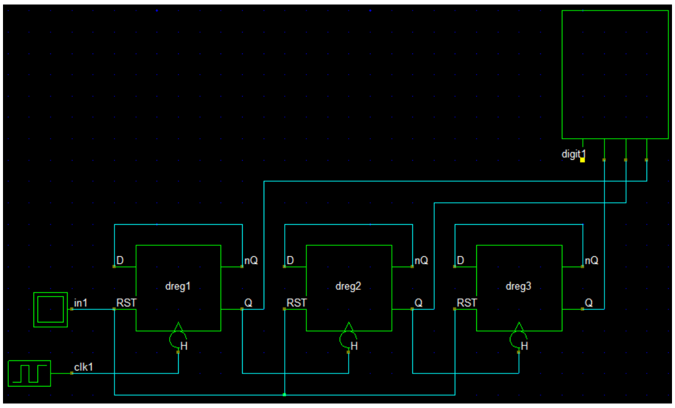


Figure 4.22. 8-Bit Counter in Dsch35

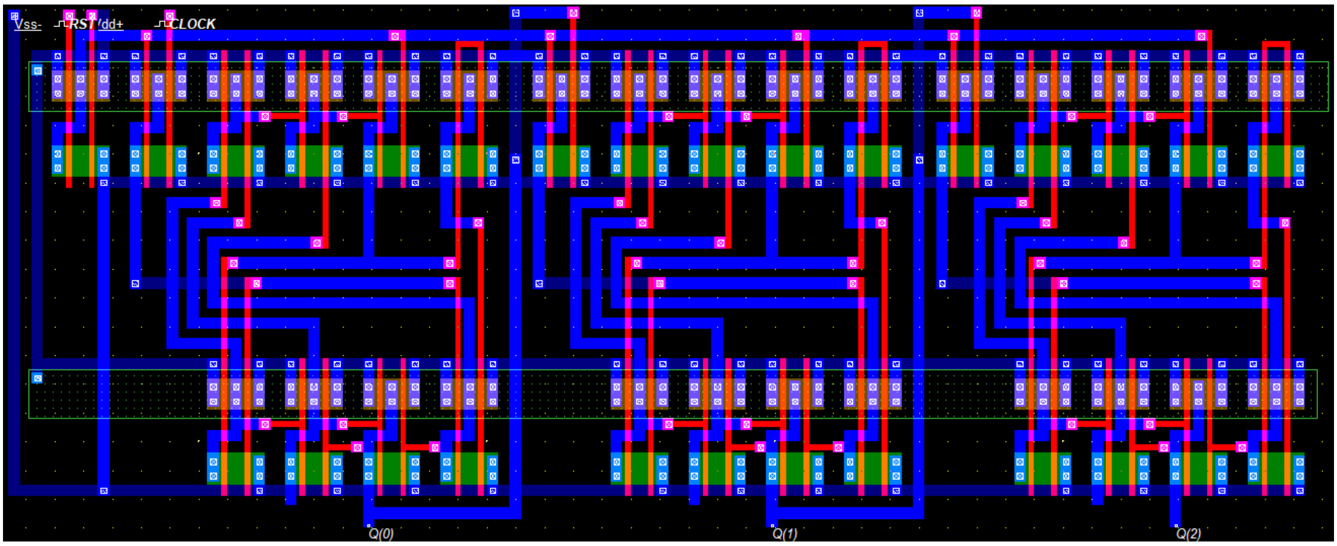
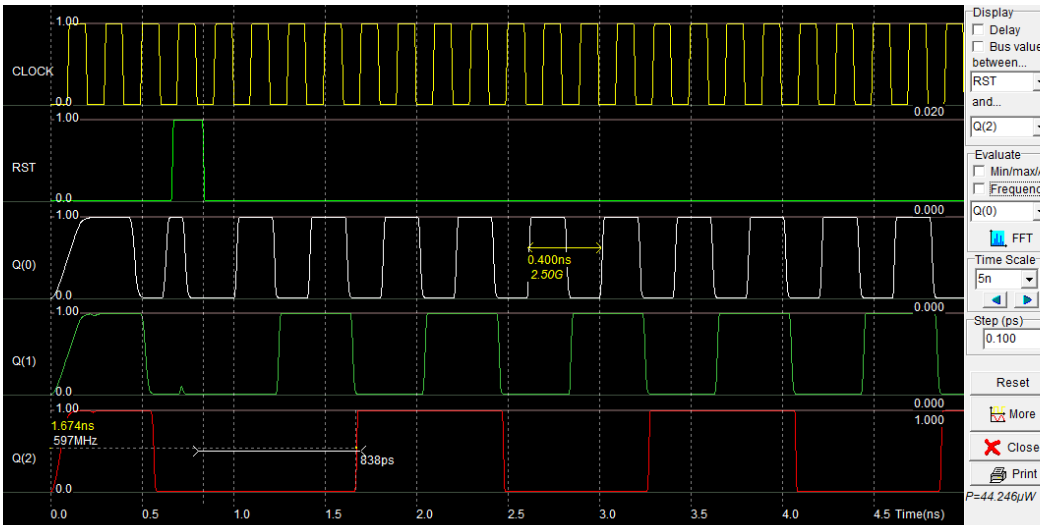


Figure 4.23. 8-Bit Counter in Microwind35

The results of the simulation of the Program Counter are the same as the ones of the 4-Bit Counter, but with only three outputs(Q0,Q1,Q2).



□ Delay
□ Bus value
between... RST
and... Q(2)
□ Min/max/
□ Frequenc
Q(0)
Time Scale 5n
Step (ps) 0.100
P=44.246µW

Figure 4.24. 8-Bit Counter simulation in Microwind35

Once more, the counting frequency is the same in each scenario. As expected, the component works faster in the best case scenario thanks to the low temperature, the increased carrier mobility and the higher voltage supply. The consumption power is roughly the same in all of the scenarios. Lastly, the component stops working properly for voltages lower than 0.4V.

	Typical parameters (27℃)	Worst parameters (125℃)	Best parameters (-50℃)
Start Time (s)	1.674n	1.740n	1.642n
Delay (s)	0.838n	0.905n	0.807n
Counting frequency (Hz)	2.5G	2.5G	2.5G
Power (W)	44.246u	46.450u	45.439u

4.6. The Instructions Decoder

The design of the Instructions Decoder was based upon the values of the following truth table. Its main feature is to send signals in order to enable the ALU, the inputs/outputs and RF circuits. In this design the Instructions Decoder can route only eight instructions from the memory. All the components of the integrated circuit are interconnected via internal bus.

C2	C1	C0		ENABLE ALU	ADD	SUB	COMP	STORE A	STORE B	OUT ENABLE	IN ENABLE	RF ENABLE	ENABLE A	ENABLE B
0	1	1	IN A	0	0	0	0	1	0	0	1	0	1	0
1	0	0	IN B	0	0	0	0	0	1	0	1	0	0	1
0	0	0	ADD	1	1	0	0	0	0	0	0	0	0	0
1	0	1	IN TO RF	0	0	0	0	0	0	0	1	1	0	0
0	0	1	SUB	0	1	0	0	0	0	0	0	0	0	0
1	1	0	OUT R	1	0	0	0	0	0	1	0	0	0	0
0	1	0	COMP	1	0	0	1	0	0	0	0	0	0	0
1	1	1	R OUT F	1	0	0	0	0	0	1	0	1	0	0

The schematic is quite simple and uses basic logic Gates, which were made before in this paper.

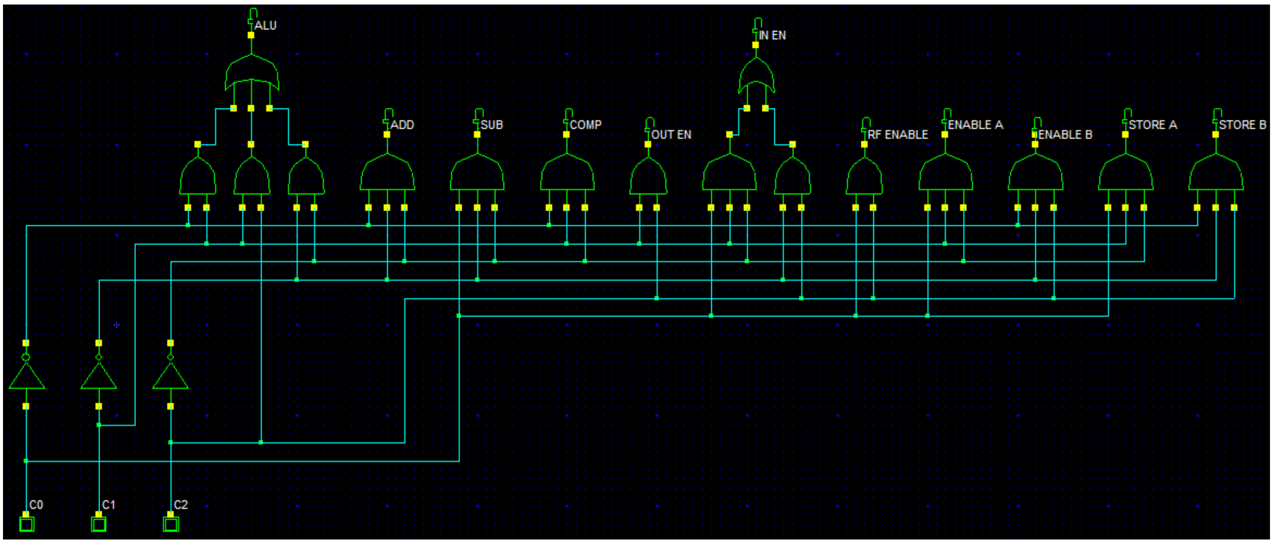


Figure 4.25. Instructions Decoder in Dsch35

The inputs ‘C0’, ‘C1’ and ‘C2’ represent the internal bus through which the memory sends instructions to the Instructions Decoder.

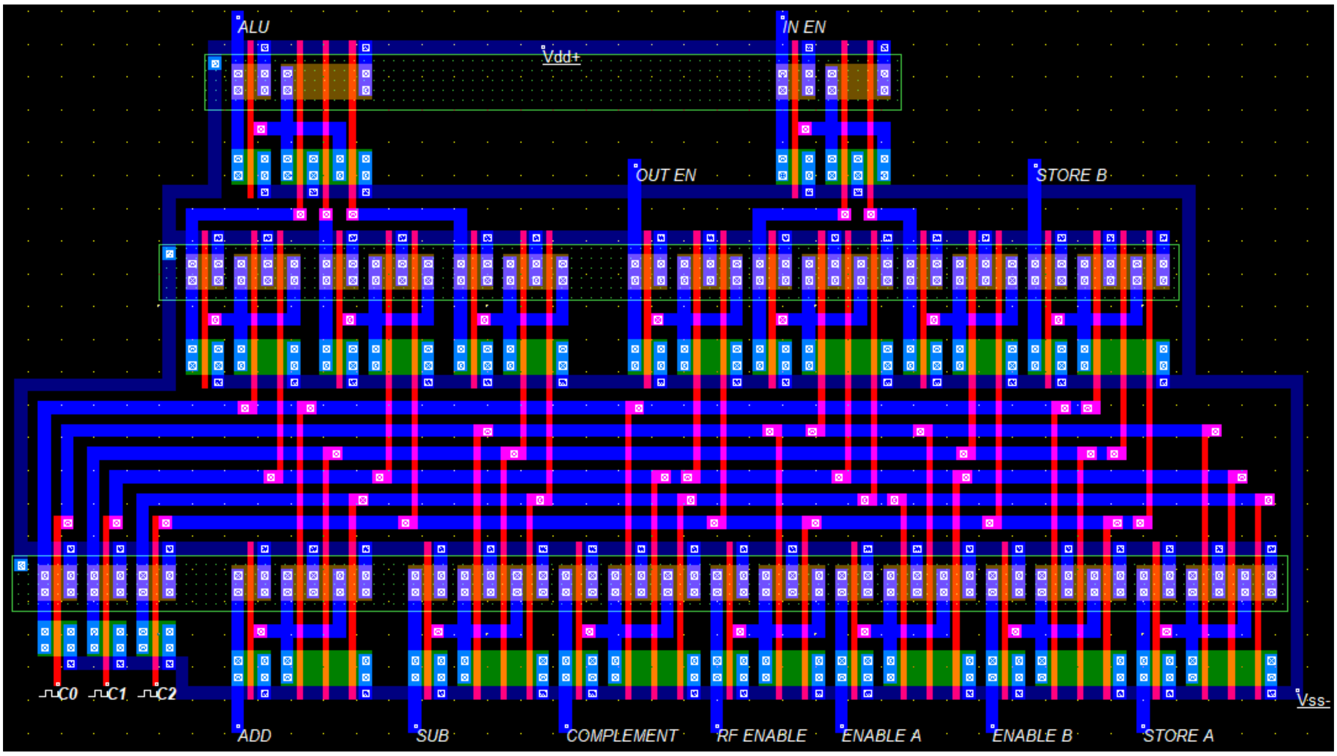


Figure 4.26. Instructions Decoder in Microwind35

Ultimately, by configuring the values of the internal bus, it is evident that the Instructions Decoder works properly.

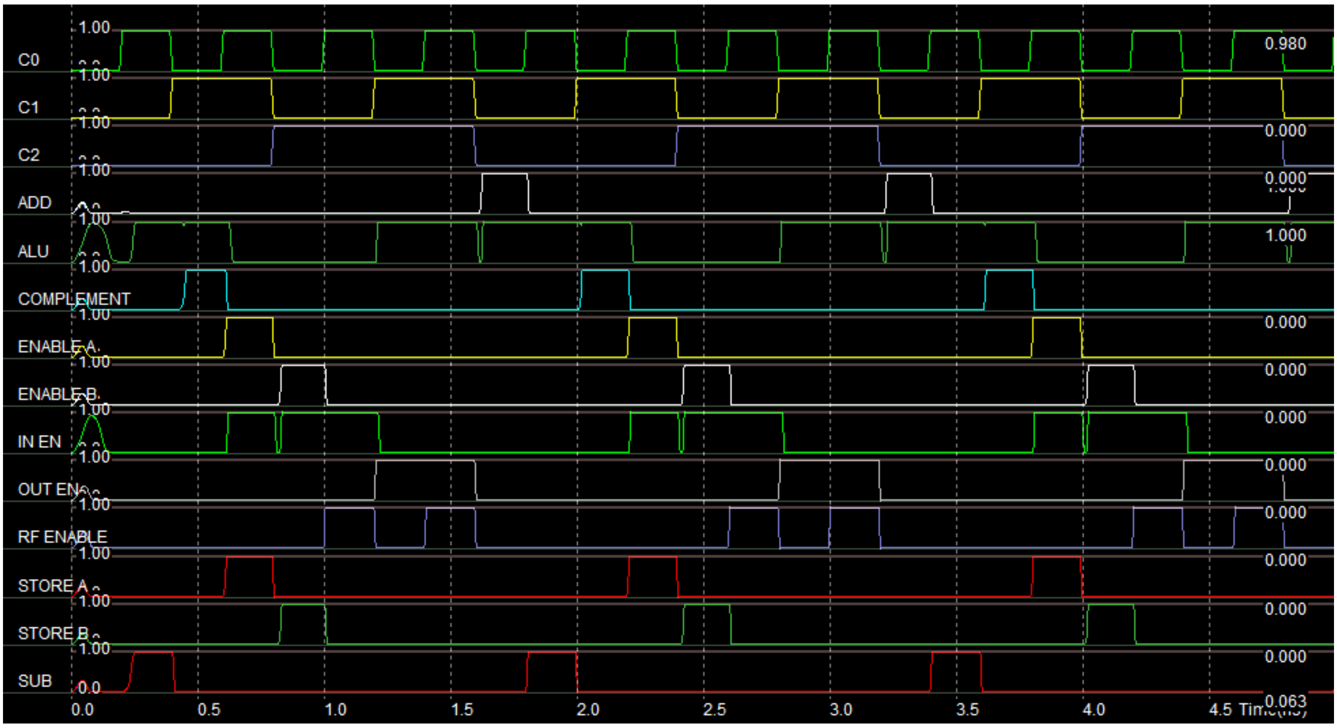


Figure 4.27. Instructions Decoder simulation in Microwind35

5. Static Random Access Memory (SRAM)

The objective of this chapter is to present the different cell designs of the SRAM and to identify their advantages and disadvantages. The showcased SRAM designs differ from each other in the number of transistors used for the individual cells. The SRAM is a volatile memory and needs a continuous supply voltage to maintain the stored information, otherwise the memory cell will lose its saved state.

The memory cell has two stable states(flip-flop) to store state information. It is a bistable multivibrator which can store one bit of information(high ‘1’ or low ‘0’). The memory is asynchronous which means that it is independent of clock frequency and data in and data out are controlled by address transition.

SRAM memory is mainly used for CPU cache, small on-chip memory, FIFOs or other small buffers by reason of being very fast for random access(faster than DRAM but also more expensive). It is also reliable and power consumption is low when idle. Conversely, the power consumption is high, compared to DRAM, when reading/writing data.

An SRAM cell has three states of operation.

- Standby: the cell is idle and holds its saved state ‘high’ or ‘low’.
- Reading: the data has been requested.
- Writing: the data is being updated.

The designs were made with 65nm(cmos65n.rul) technology.

5.1. Differential Voltage Sense Amplifier

Its primary function is the evaluation of the small analog voltage difference applied to its inputs and the conversion of it into a logic level output signal. Sense amplifiers are implemented in voltage and in current sensed SRAM, thanks to their simple design and robustness in the appearance of noise. They can be classified as static and dynamic circuits.

The design(Figure 5.1) used in this paper, is a typical circuit for a static voltage sense amplifier, but it contains a power down-switch to cut off static current. The bottom transistor is controlled by the logic input ‘SE’ to reduce the power consumption, due to static current, by activating and deactivating the sensing operation. The sensing operation must be nondestructive, as the SRAM cell does not feature data refresh after sensing.

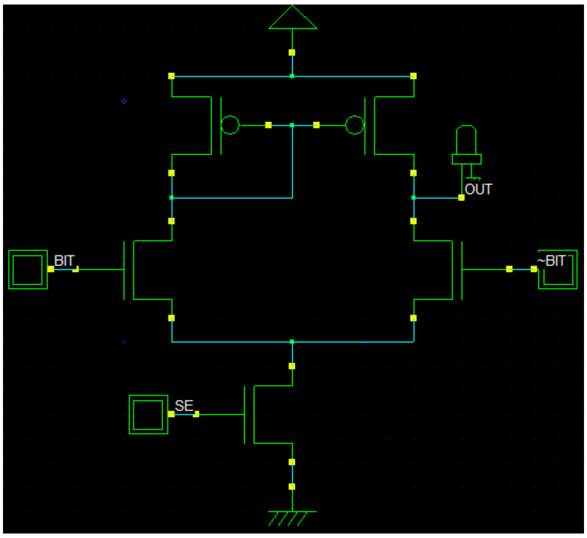
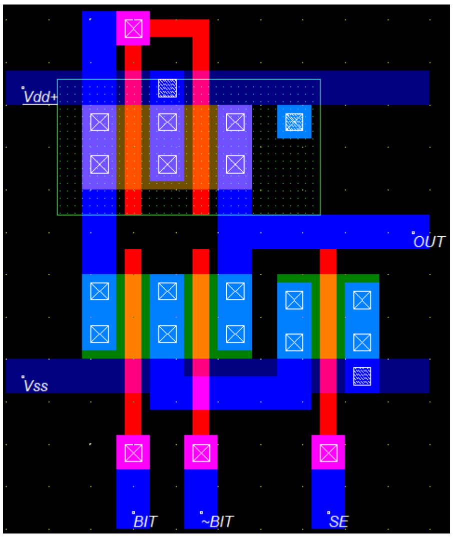
	

Figure 5.1. Sense Amplifier in Dsch35	**Figure 5.2.** Sense Amplifier in Microwind35

The sensing operation begins when there is high voltage in the ‘SE’ input. When the ‘BIT’ input is high(1) and the ‘~BIT’ input is low(0), the ‘OUT’ output will be high. When the ‘BIT’ input is low(0) and the ‘~BIT’ input is high(1), the ‘OUT’ output will be low. Only the difference between the two bitlines is amplified.

5.2. SRAM 6T cell

The circuit consists of two bitlines(BIT/~BIT) which hold the information(state) that needs to be saved in the cell. For the two inverters to work properly, the bitlines must always be in inverted state with each other. The 'WL' input is the signal that allows the writing operation and connects with the input transistors(T3, T6). The state of the bitlines is saved in the cell, when there is a high voltage in the 'WL' input. The outputs 'Q' and 'QB' of the cell connect to the inputs 'BIT' and '~BIT' of the sense amplifier respectively.

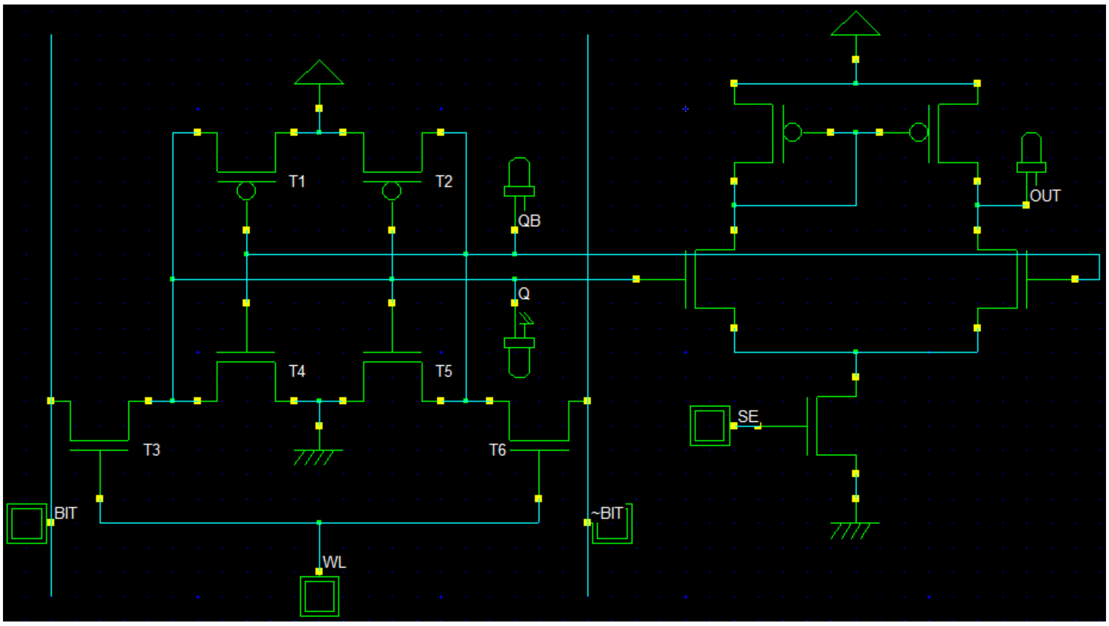


Figure 5.3. SRAM 6T cell in Dsch35

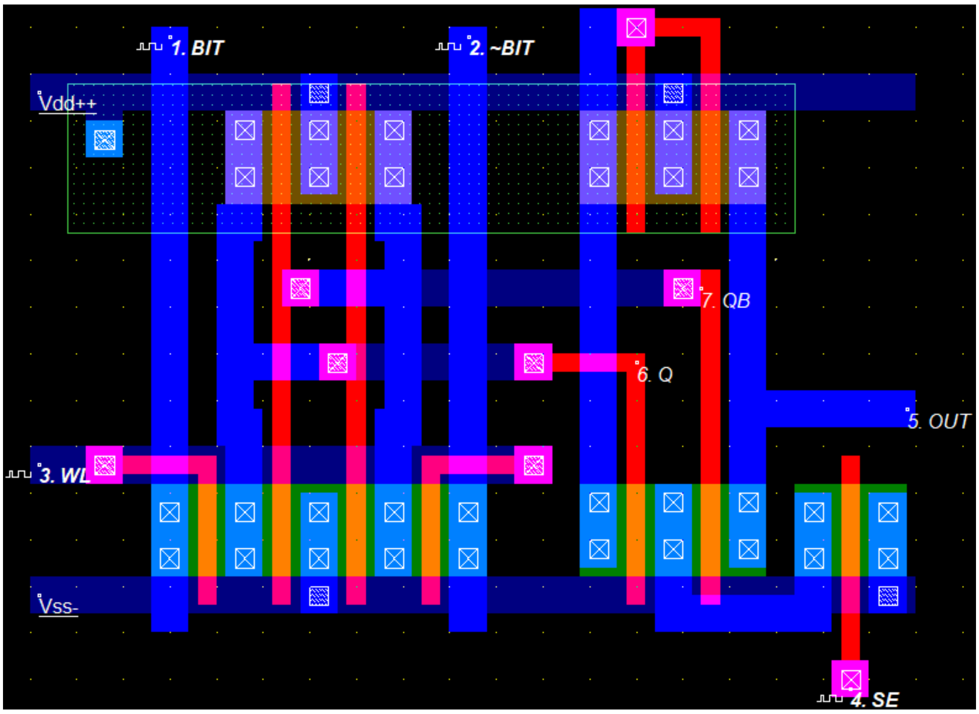
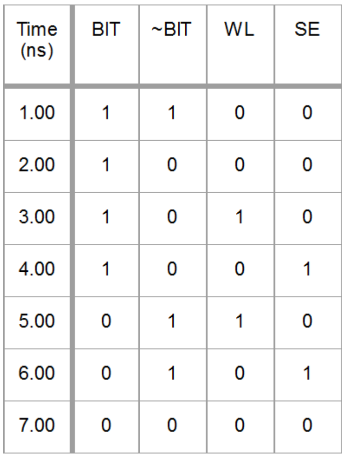


Figure 5.4. SRAM 6T cell in Microwind35

- **Writing high(1) into the cell:** the 'BIT' bitline needs to be high and the '~BIT' bitline low, as they must have inverted states. The input 'WL' needs to be high, for the state to be saved into the cell.
- **The cell is on standby:** regardless of the state of the bitlines, as long as the 'WL' input is low the cell holds its saved state.
- **Reading the state of the cell:** the 'SE' input of the sense amplifier needs to be high to allow the reading of the state of the cell. The sense amplifier compares the outputs(Q/QB) of the cell and converts the state of the cell into a logic level output signal(OUT).

A sequence(Table 5.1) of bits was used for the simulation(Figure 5.5) of the SRAM 6T cell. From the simulation it is noticeable that the output signal 'OUT' of the Sense Amplifier is high from the beginning. The reason for this, is that the state the cell "fell" in, at the time the circuit was powered on, was high. This is something to be expected in a bistable circuit, because the components are never identical and every lithography is unique. At the timestamp 3.0ns the 'WL' goes high to allow the writing of the bit 1(high) into the cell and at the timestamp 4.0ns the 'SE' goes high to allow the reading of the state of the cell at the output 'OUT'. At the timestamp 5.0ns the 'WL' goes high to allow the writing of the bit 0(low) into the cell and at the timestamp 6.0ns the 'SE' goes high to allow the reading of the state of the cell at the output 'OUT'.



Time (ns)	BIT	~BIT	WL	SE
1.00	1	1	0	0
2.00	1	0	0	0
3.00	1	0	1	0
4.00	1	0	0	1
5.00	0	1	1	0
6.00	0	1	0	1
7.00	0	0	0	0

Table 5.1. Bit sequence of SRAM 6T cell

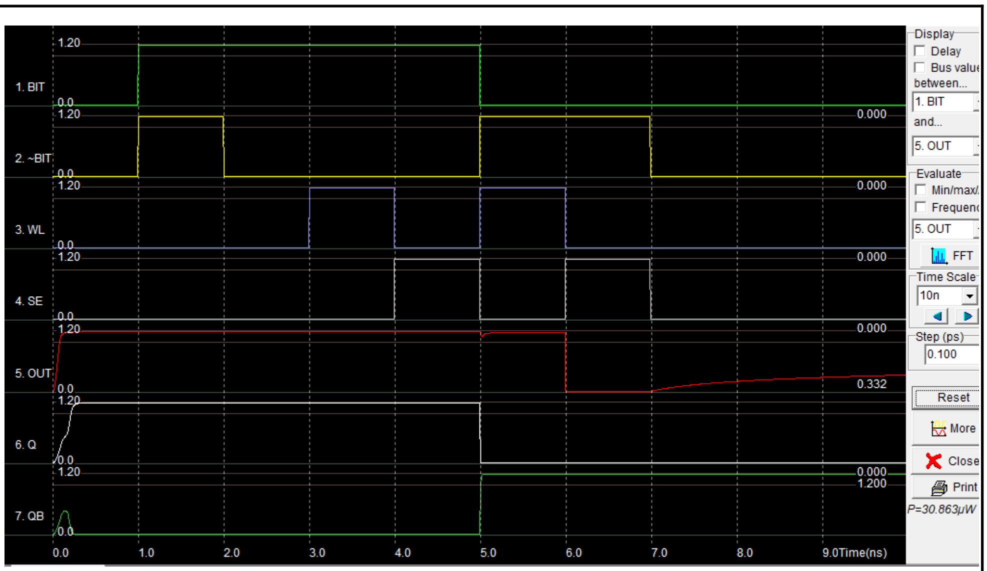


Figure 5.5. SRAM 6T cell simulation in Microwind35

5.3 SRAM 7T cell

In this design the input transistor T3 is connected to the Word Line(WL) input to control the writing operation and the transistor T7 is connected to the Read Line(R) to control the reading operation. The inverted 'QB' output of the cell is controlled by the new transistor T6. When '~BIT' goes high, the state of the cell goes low and when '~BIT' goes low, the state of the cell goes high.

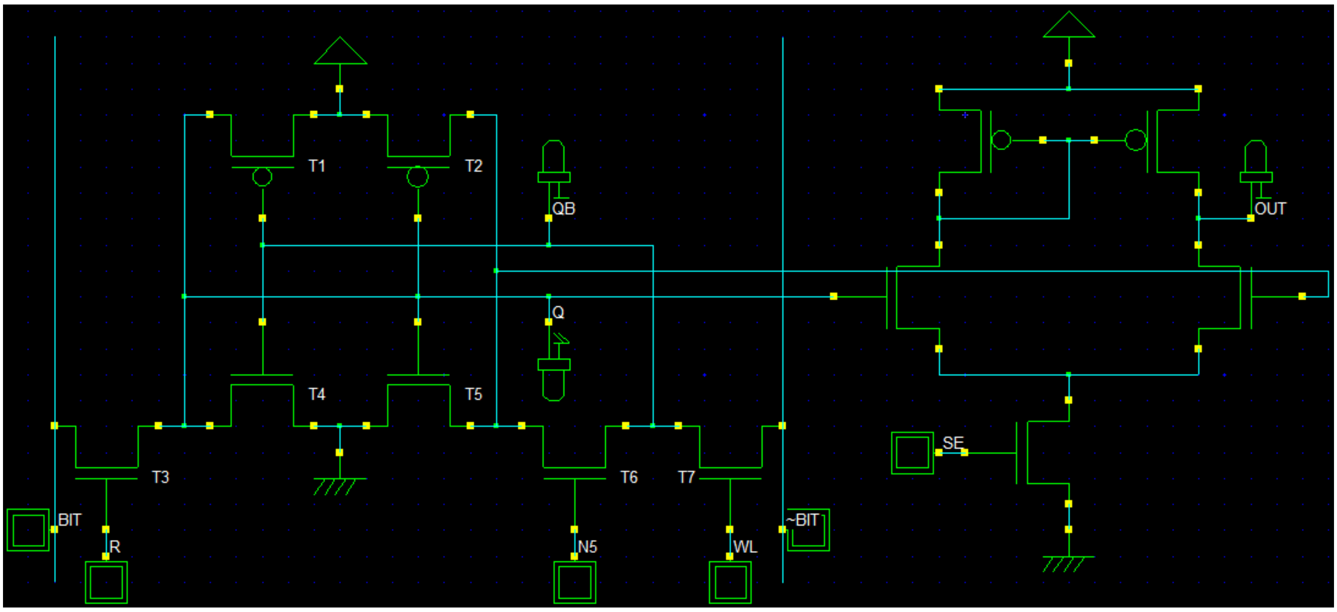


Figure 5.6. SRAM 7T cell in Dsch35

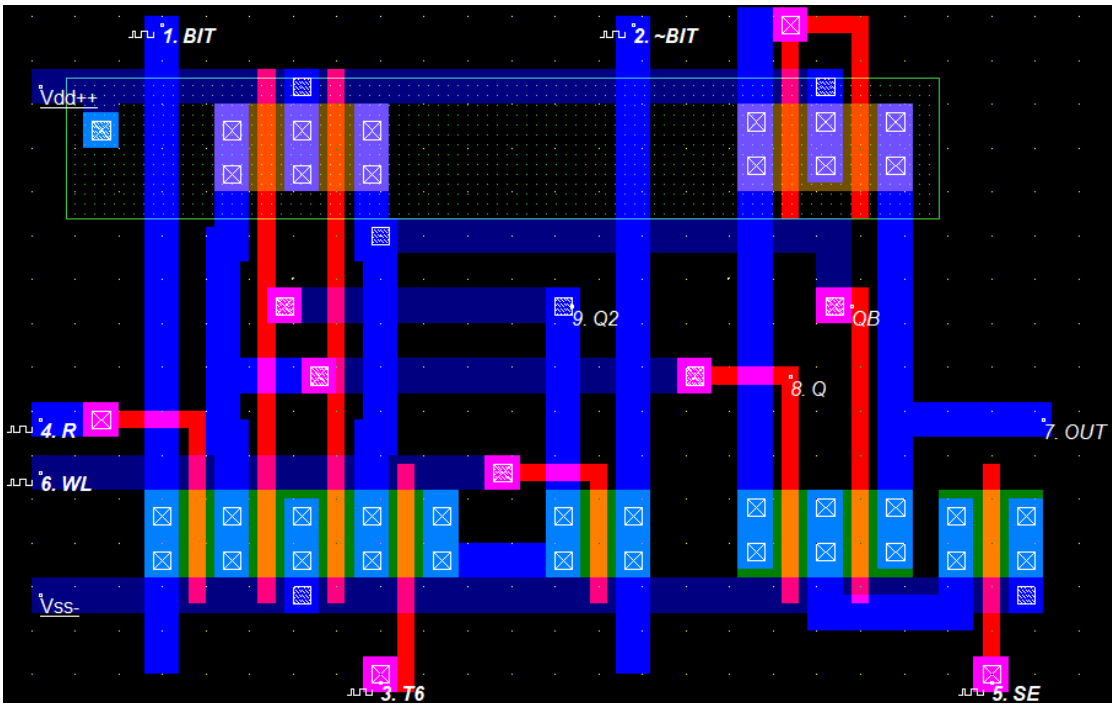


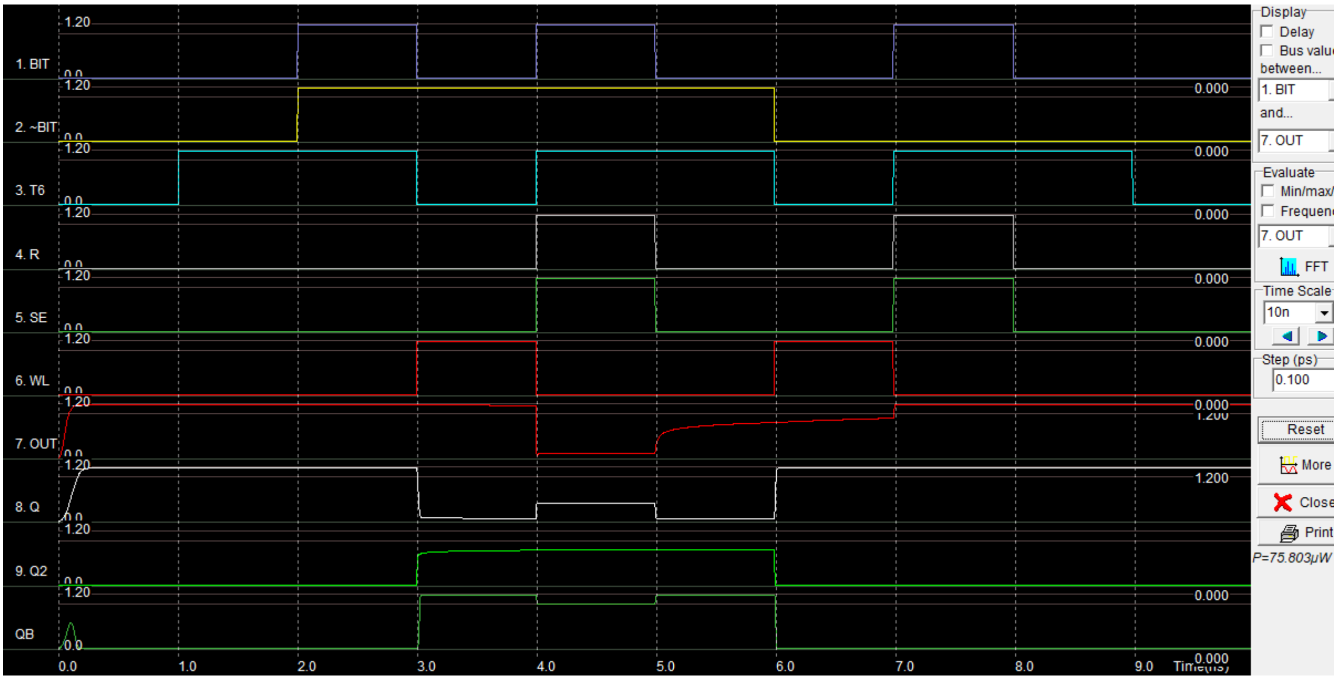
Figure 5.7. SRAM 7T cell in Microwind35

- **Writing into the cell:** First of all, by disabling the T6 the writing operation begins, as it is the transistor responsible for the feedback of the inverted ‘QB’ output. Then the ‘WL’ goes high and the T7 transfers the data from the ‘~BIT’ bitline which drives the T1 and T4 transistors to save the value ‘Q’ into the cell. Additionally, the ‘Q’ signal drives the T2 and T5 to save the value QB. The signal ‘QB’ has the same state as ‘Q2’, when the saved state of the cell is ‘0’, and slightly higher when the saved state is ‘1’. Lastly, the ‘WL’ goes low and the T6 gets enabled and the feedback between the two inverters gets connected so that the new state of the cell can be saved.
- **Requesting data from the cell:** The ‘BIT’ and the ‘R’ signals go high, as well as the T6. When ‘Q’ equals ‘0’ the reading route consists of the transistors T4 and T3. When ‘Q’ equals ‘1’ the reading route consists of the transistors T5, T6 and T7.

Time (ns)	BIT	~BIT	WL	T6	R	SE
1.00	0	0	0	1	0	0
2.00	1	1	0	1	0	0
3.00	0	1	1	0	0	0
4.00	1	1	1	1	1	1
5.00	0	1	0	1	0	0
6.00	0	0	1	0	0	0
7.00	1	0	1	1	1	1
8.00	0	0	0	1	0	0
9.00	0	0	0	0	0	0

Table 5.2. Bit sequence of SRAM 7T cell

A sequence(Table 5.2) of bits was used for the simulation(Figure 5.8) of the SRAM 7T cell. At the timestamp 3.0ns the ‘WL’ goes high to allow the writing of new data into the cell. Because the ‘~BIT’ is high, the new state of the cell is going to be low. Additionally the ‘T6’ goes low. At the timestamp 4.0ns the ‘T6’ goes high so that the state of the cell gets saved and also ‘R’ and ‘BIT’ go high to allow the reading of the new state of the cell at the output ‘OUT’ of the Sense Amplifier. The ‘SE’ input of the Sense Amplifier needs to be high as well.



□ Delay
□ Bus value between... 1. BIT and... 7. OUT

Evaluate
□ Min/max
□ Frequency
7. OUT

Time Scale 10n

Step (ps) 0.100

P=75.803µW

Figure 5.8. SRAM 7T cell simulation in Microwind35

5.4. SRAM 8T cell

The circuit's configuration creates a strong feedback signal in order to maintain the data into the cell and to also be reliable during the writing process.

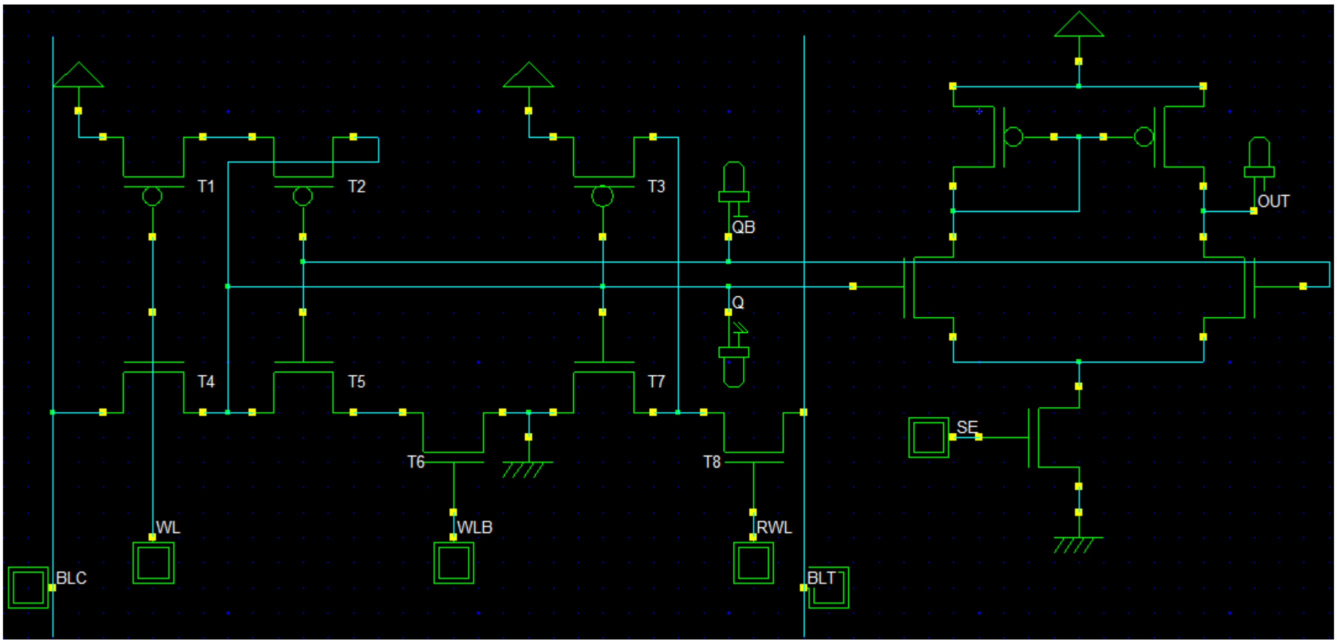


Figure 5.9. SRAM 8T cell in Dsch35

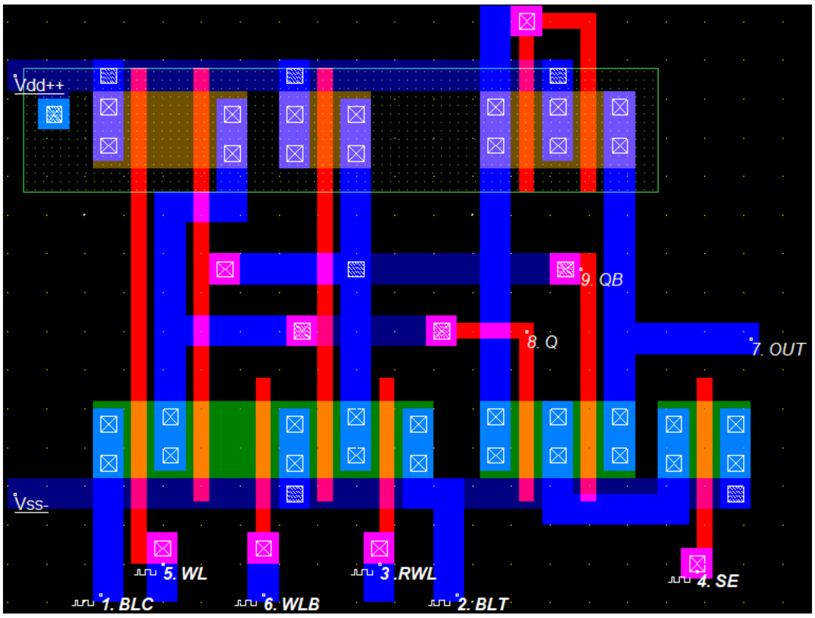


Figure 5.10. SRAM 8T cell in Microwind35

- **Writing into the cell:** the ‘T4’ is the transistor responsible for the writing operation. The writing process begins with the activation of the ‘T4’ transistor by the ‘WL’ input which goes high. The data are being transferred from the ‘BLC’ input to the cell. In order for the writing process to be successful, the transistors ‘T1’ and ‘T6’ need to be ‘closed’ so that the left inverter is ‘weaker’ than the right one.
- **The cell is on standby:** in order for the information to be maintained into the cell, the ‘RWL’ and ‘WL’ inputs need to be ‘closed’ while the ‘T1’ and ‘T6’ need to be ‘open’ in order to create a strong feedback. The ‘WLB’ also goes high.
- **Reading the state of the cell:** to allow the reading of the state of the cell, the ‘BLT’ and the read-word-line(RWL) inputs need to be high as well as the ‘SE’ input of the sense amplifier. The ‘WLB’ goes low.

Time (ns)	BLC	WL	RWL	BLT	WLB	SE
1.00	1	1	0	0	0	0
2.00	1	0	0	0	1	0
3.00	0	0	1	1	0	1
4.00	0	1	0	0	0	0
5.00	0	0	0	0	1	0
6.00	0	0	1	1	0	1
7.00	0	0	0	0	0	0

Table 5.3. Bit sequence of SRAM 8T cell

A sequence(Table 5.3) of bits was used for the simulation(Figure 5.11) of the SRAM 8T cell. At the timestamp 1.0ns the ‘WL’ goes high to allow the writing of new data into the cell. Because the ‘BLC’ is high, the new state of the cell is going to be high. At the timestamp 2.0ns the ‘WLB’ and every other input signal is low as the cell is on standby mode. At the timestamp 3.0ns the ‘WLB’ goes low while the ‘RWL’ and ‘BLT’ go high to allow the reading of the new state of the cell at the output ‘OUT’ of the Sense Amplifier. The ‘SE’ input of the Sense Amplifier needs to be high as well.

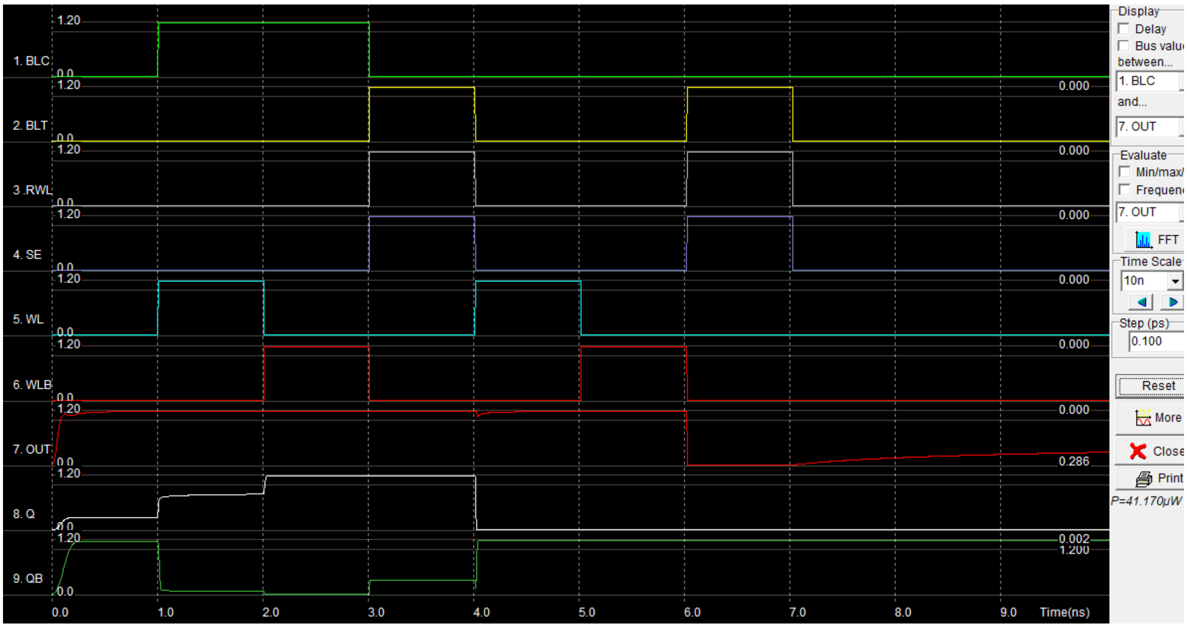


Figure 5.11. SRAM 8T cell simulation in Microwind35

5.5. SRAM 9T cell

In this design, the transistor T1 is connected as a diode. Its main feature is to reduce the dynamic voltage, which compromises the stability of the cell. This dynamic voltage scaling increases the probability of data inversion in the cell, because of the working bitlines during the reading operation.

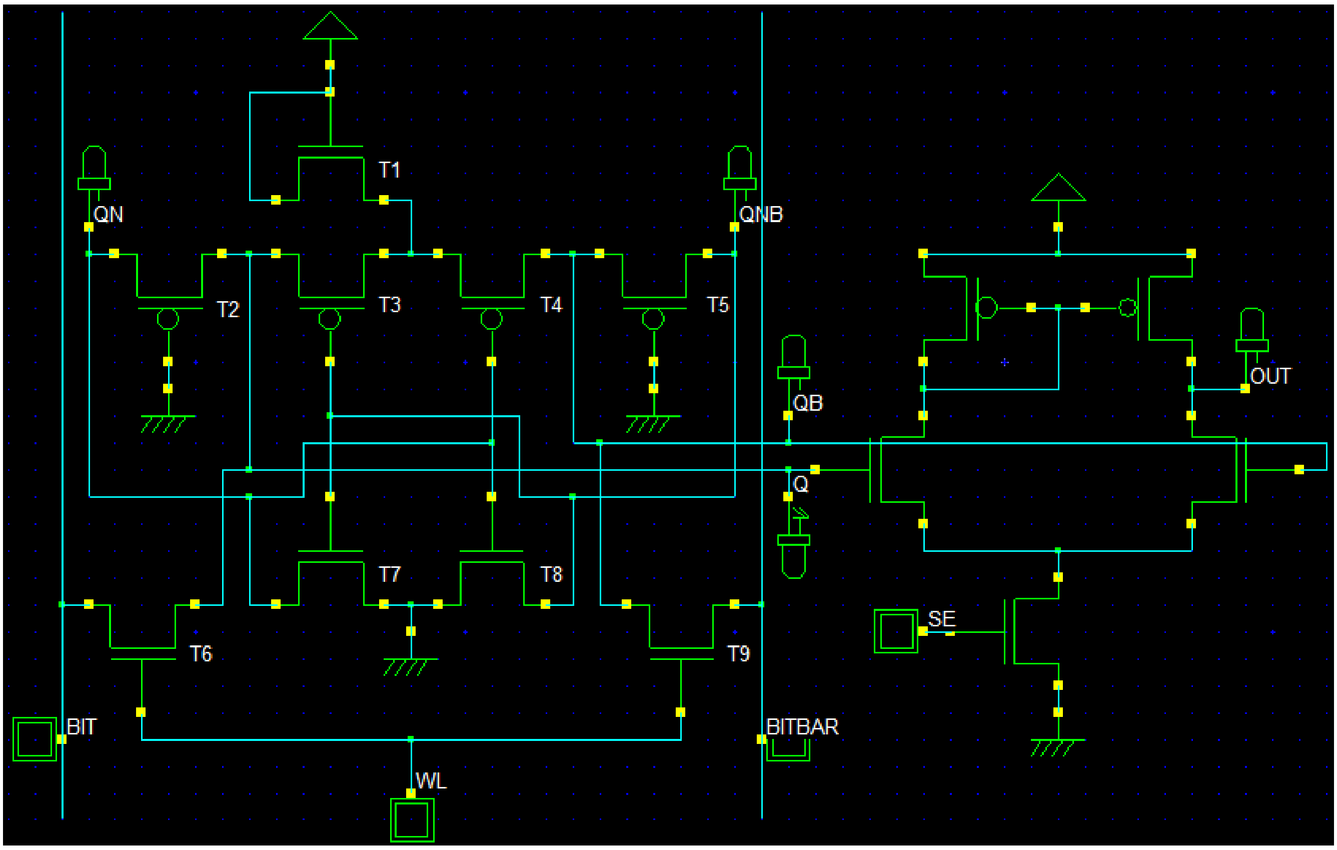


Figure 5.12. SRAM 9T cell in Dsch35

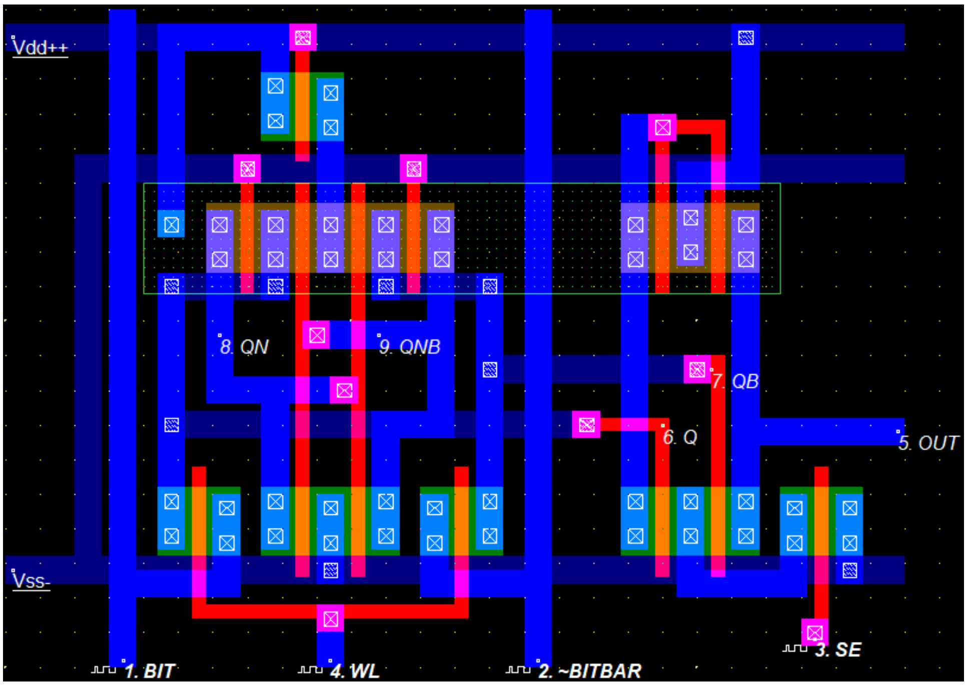


Figure 5.13. SRAM 9T cell in Microwind35

This reduction in stability is compensated by the transistors ‘T2’ and ‘T5’ which are always turned on and placed between the driver and input transistors. This arrangement increases the stability of the cell during the reading operation.

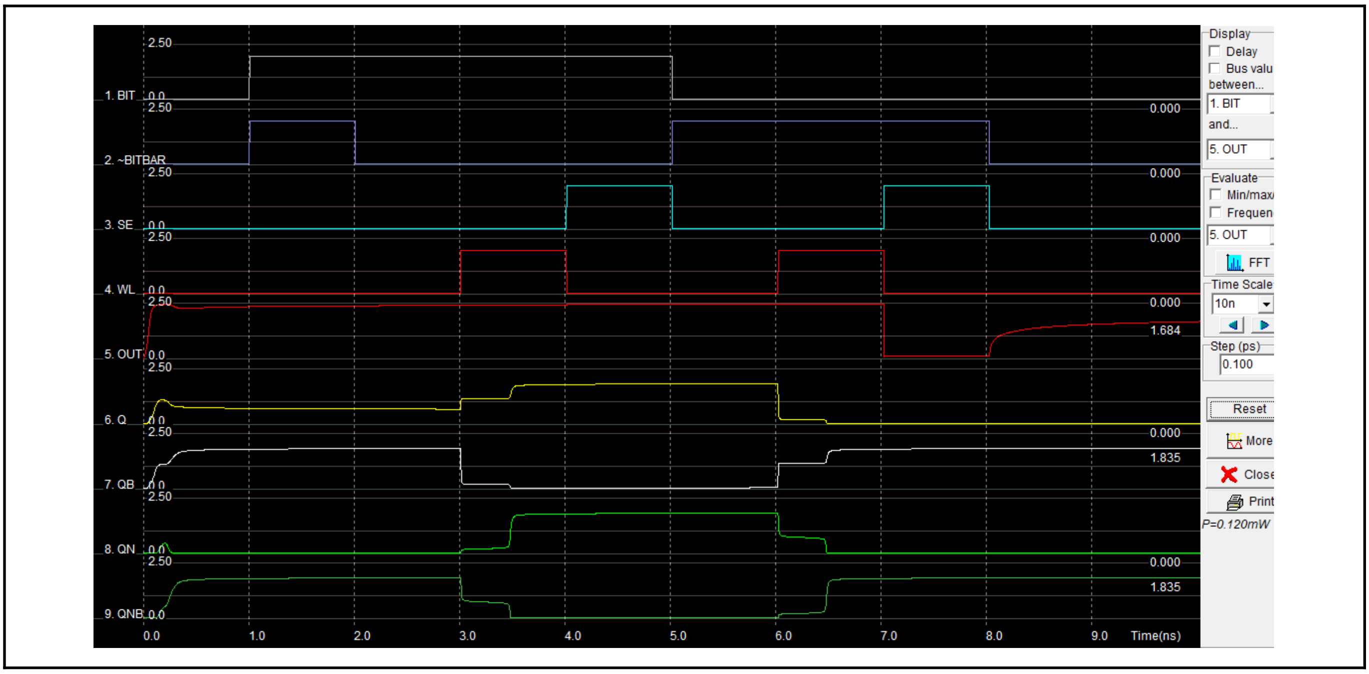
For instance, when reading ‘1’, the ‘QN’ is ‘1’ and the ‘QNB’ is ‘0’. When the ‘WL’ goes high and the voltage gets divided between ‘T6’, ‘T9’, ‘T5’ and ‘T8’, they decrease the voltage of the ‘QNB’. As a consequence this reduced voltage is not capable of reverting the state of the cell. The saved states in ‘Q’ and ‘QB’ never reach the Vdd voltage, but this is not a problem since a differential sense amplifier is used for the reading of data.

- **Writing into the cell:** the procedure is the same as in SRAM 6T cell.
- **The cell is on standby:** regardless of the state of the bitlines, as long as the ‘WL’ and ‘SE’ inputs are low the cell holds its saved state. As in every circuit of static RAM, the cell needs a continuous supply voltage to maintain the stored information.
- **Reading the cell state:** firstly, the ‘WL’ input needs to be low and the ‘SE’ input high. The sense amplifier compares the outputs(Q/QB) of the cell and converts the state of the cell into a logic level output signal(OUT).

A sequence(Table 5.4) of bits was used for the simulation(Figure 5.11) of the SRAM 9T cell. At the timestamp 6.0ns the ‘WL’ goes high to allow the writing of the bit 0(low) into the cell and at the timestamp 7.0ns the ‘SE’ goes high to allow the reading of the state of the cell at the output ‘OUT’.

Time (ns)	BIT	~BITBAR	WL	SE
1.00	1	1	0	0
2.00	1	0	0	0
3.00	1	0	1	0
4.00	1	0	0	1
5.00	0	1	0	0
6.00	0	1	1	0
7.00	0	1	0	1
8.00	0	0	0	0

Table 5.4. Bit sequence of SRAM 9T cell



Time Scale 10n

Step (ps) 0.100

Figure 5.14. SRAM 9T cell simulation in Microwind35

5.6. SRAM 10T cell

The cell function of the 10T cell is similar to the 6T, the main difference is that the 10T cell has read and write buffers on either side, in order to improve the reading and writing functions.

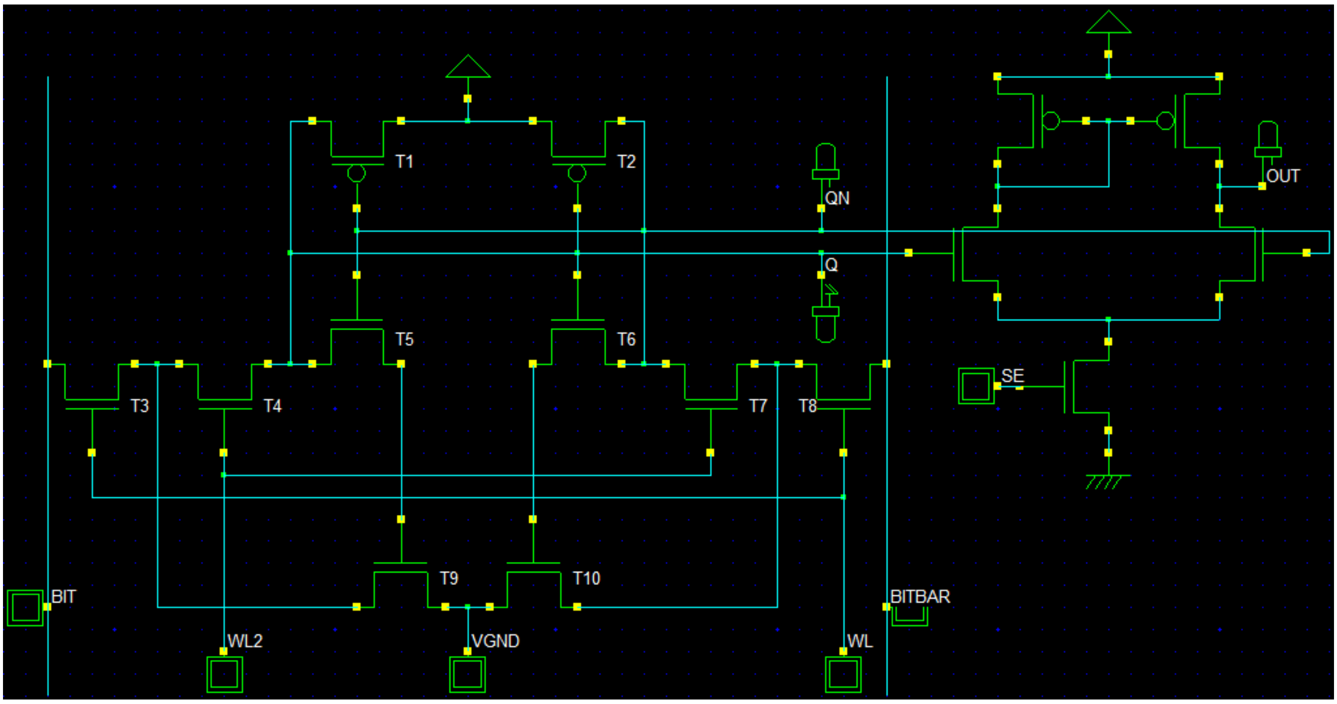


Figure 5.15. SRAM 10T cell in Dsch35

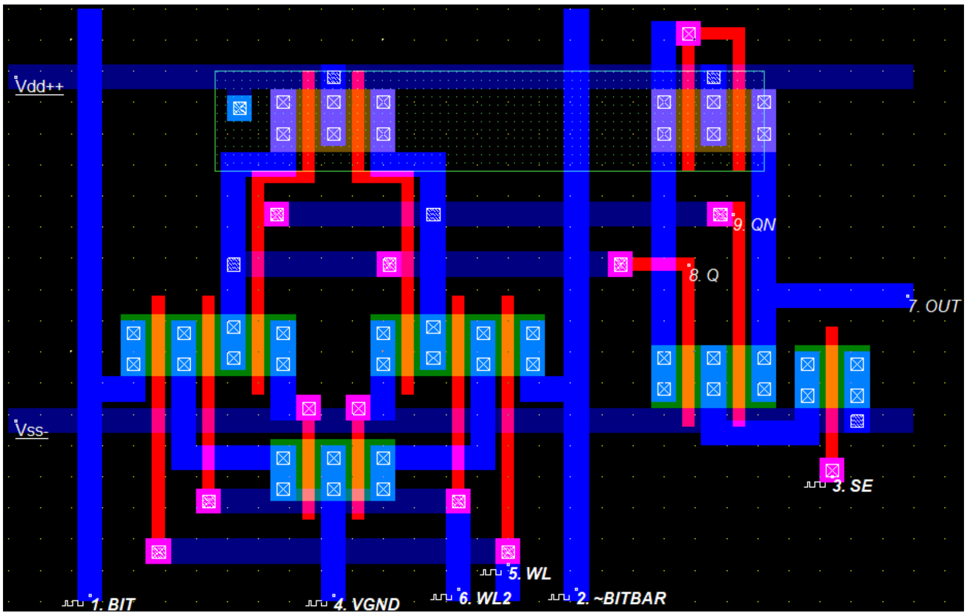


Figure 5.16. SRAM 10T cell in Microwind35

- **Writing into the cell:** the inputs ‘WL’ and ‘WL2’ are both capable of transferring information from the bitlines to the cell. The ‘VGND’ signal has the same voltage as the Vdd to compensate for any voltage threshold problem that may arise from the two sides of the input transistors.
- **The cell is on standby:** the two wordlines(WL, WL2) need to be low and the ‘VGND’ input needs to have the same voltage as the Vdd.
- **Reading the cell state:** the worldline ‘WL2’ needs to be low, whereas the ‘WL’ must be high due to the fact that the bitlines would be disconnected from the cell’s state junctions otherwise and that may lead to noise during the reading operation. The ‘VGND’ input gets connected to earth to create a route for discharge.

Time (ns)	BIT	~BITBAR	VGND	WL	WL2	SE
1.00	1	1	1	0	0	0
2.00	1	0	1	1	1	0
3.00	0	0	1	0	0	0
4.00	0	0	0	1	0	1
5.00	0	1	1	1	1	0
6.00	0	0	1	0	0	0
7.00	0	0	0	1	0	1
8.00	0	0	0	0	0	0

Table 5.5. Bit sequence of SRAM 10T cell

A sequence(Table 5.5) of bits was used for the simulation(Figure 5.17) of the SRAM 10T cell. At the timestamp 5.0ns the ‘WL’ and ‘WL2’ go high to allow the writing of ‘0’ into the cell. At the timestamp 6.0ns the ‘WL’ and ‘WL2’ go low and the cell goes on standby mode. At the timestamp 7.0ns the ‘WL’ and ‘SE’ go high whereas the ‘WL2’ stays low to allow the reading of the new state of the cell at the output ‘OUT’ of the Sense Amplifier.

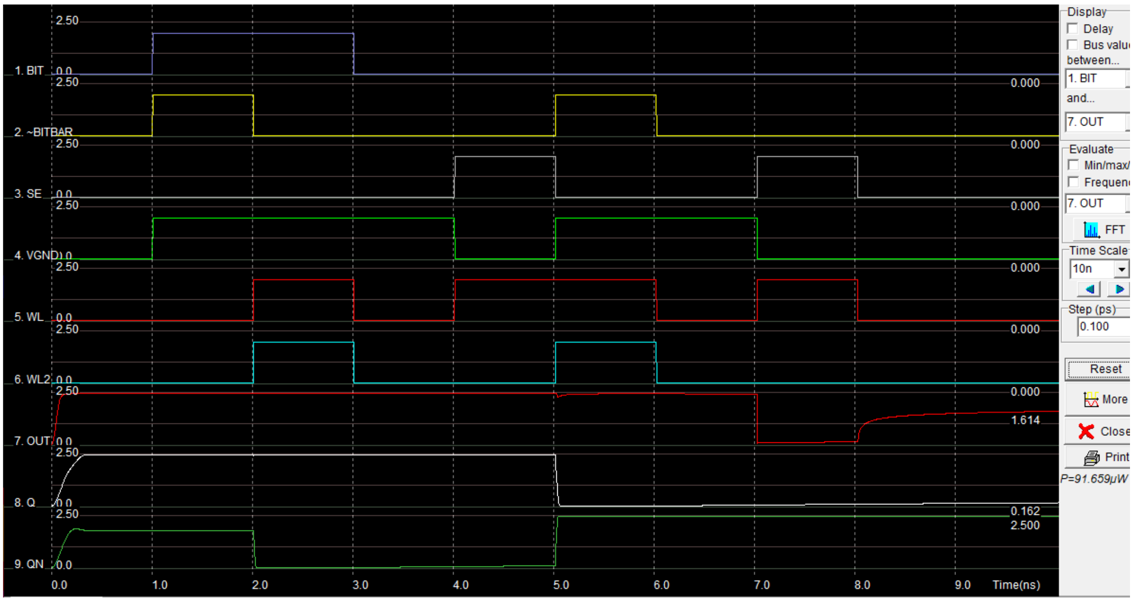


Figure 5.17. SRAM 10T cell simulation in Microwind35

5.7. The design of the 1Byte SRAM

The 1Byte SRAM consists of three components; a Decoder, eight SRAM cells and a Multiplexer. The 6T cell was used for the design of the 1Byte SRAM, because it has a quite simple design. Eight SRAM cells are needed to save an 1Byte number, because it consists of 8bits and every cell is capable of saving the state of one bit.

5.7.1. The design of the Decoder

The decoder is simple in design and its main feature is to select which SRAM cell is going to be written. The inputs ‘A’, ‘B’, and ‘C’ are used for the selection of a particular cell and the ‘PR’ input is used as a flag that allows the writing of the cells only when is needed.

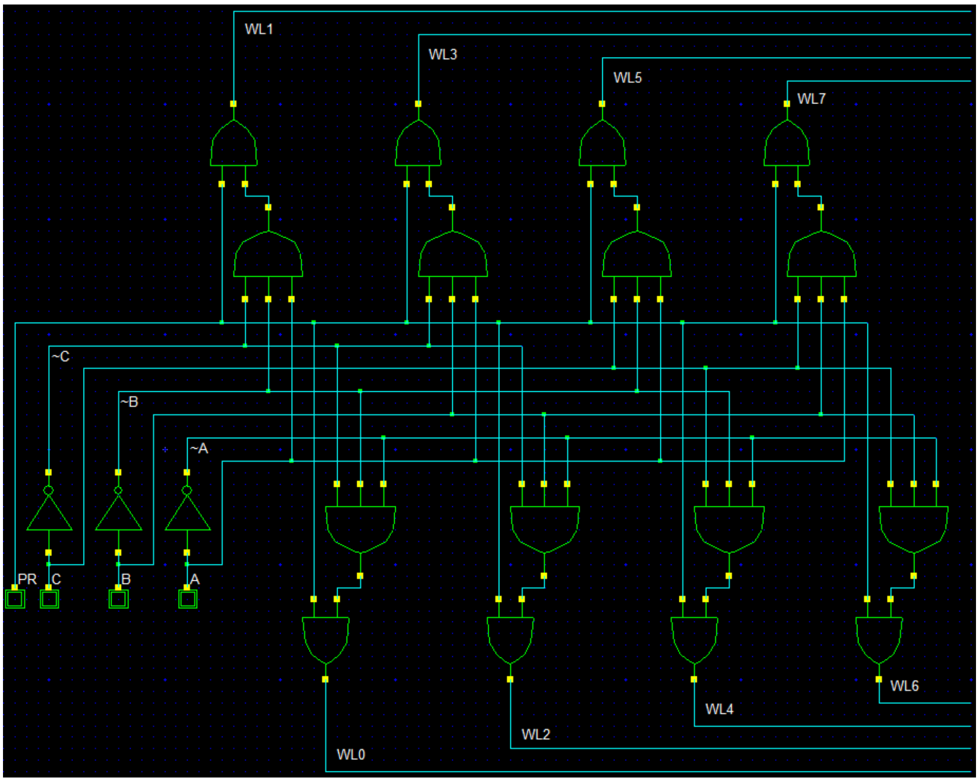


Figure 5.18. SRAM’s Decoder in Dsch35

The ‘WL#’ outputs go to the ‘WL’ inputs of the SRAM cells.

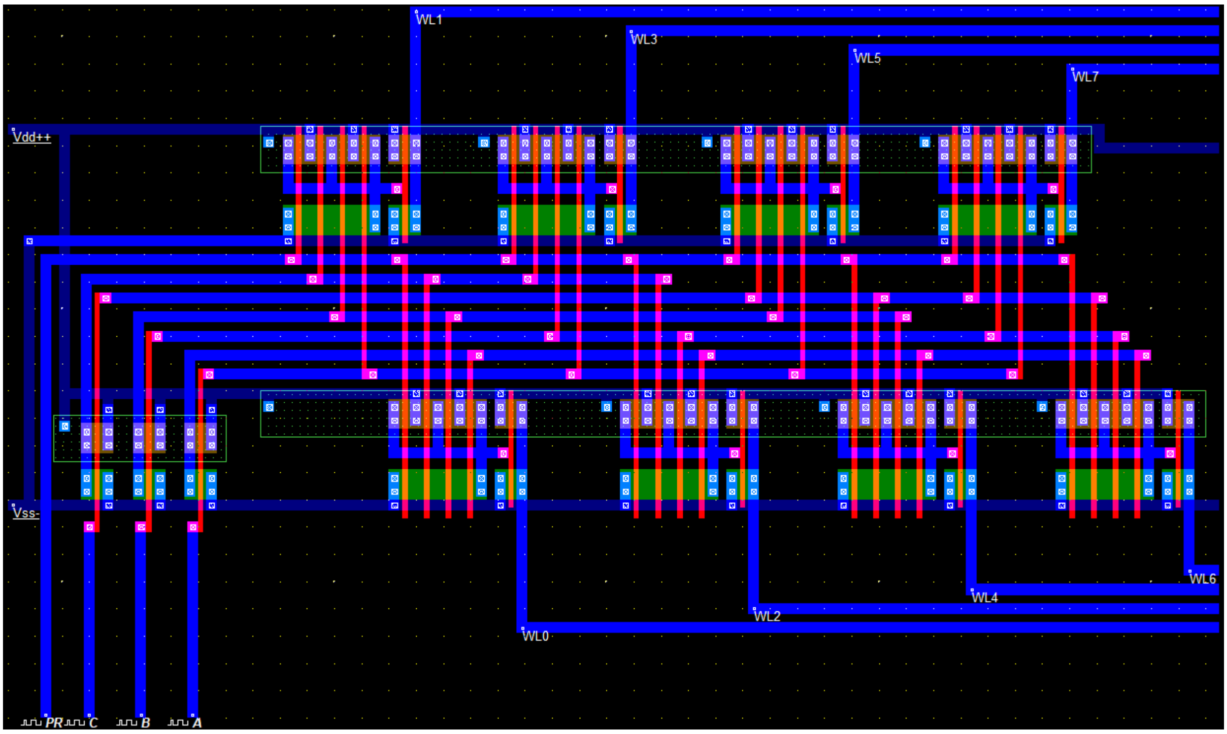


Figure 5.19. SRAM’s Decoder in Microwidn35

5.7.2. The design of the SRAM cells

As mentioned before, the 6T cell was used for the design of the 1Byte SRAM. The input ‘SE’ is used to allow the reading of the cell only when is needed. The ‘BL’ input serves as the non-inverted biltine and the ‘BLB’ as the inverted bitline.

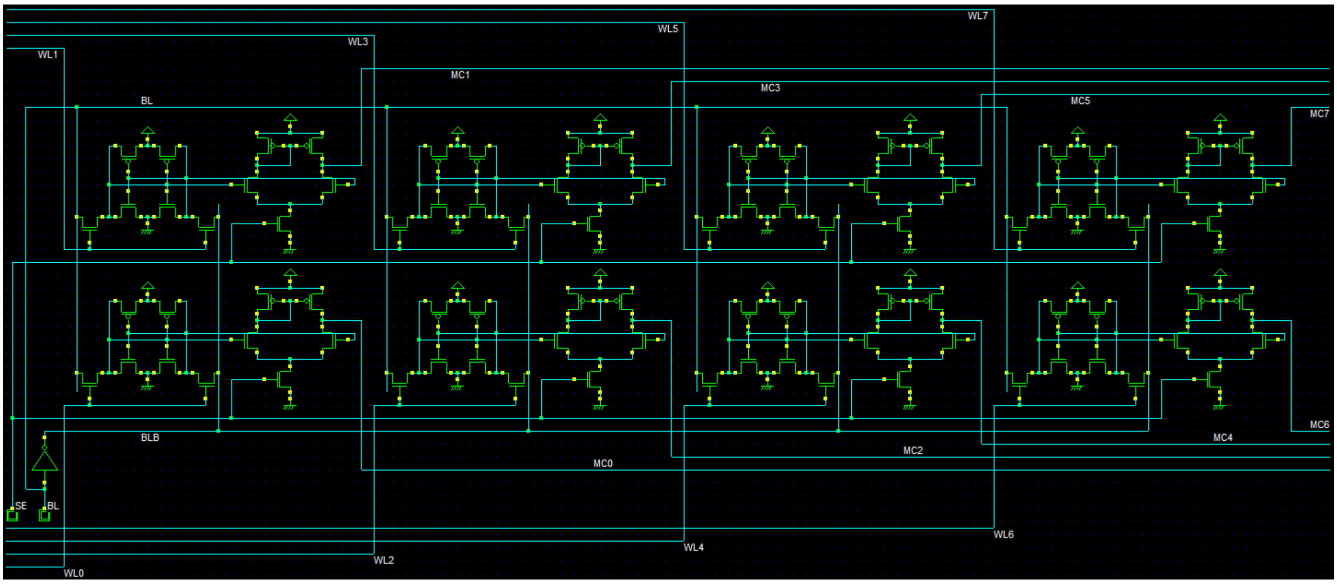


Figure 5.20. SRAM’s cells in Dsch35

The ‘MC#’ outputs go through the multiplexer which allows the reading of a particular cell on the output.

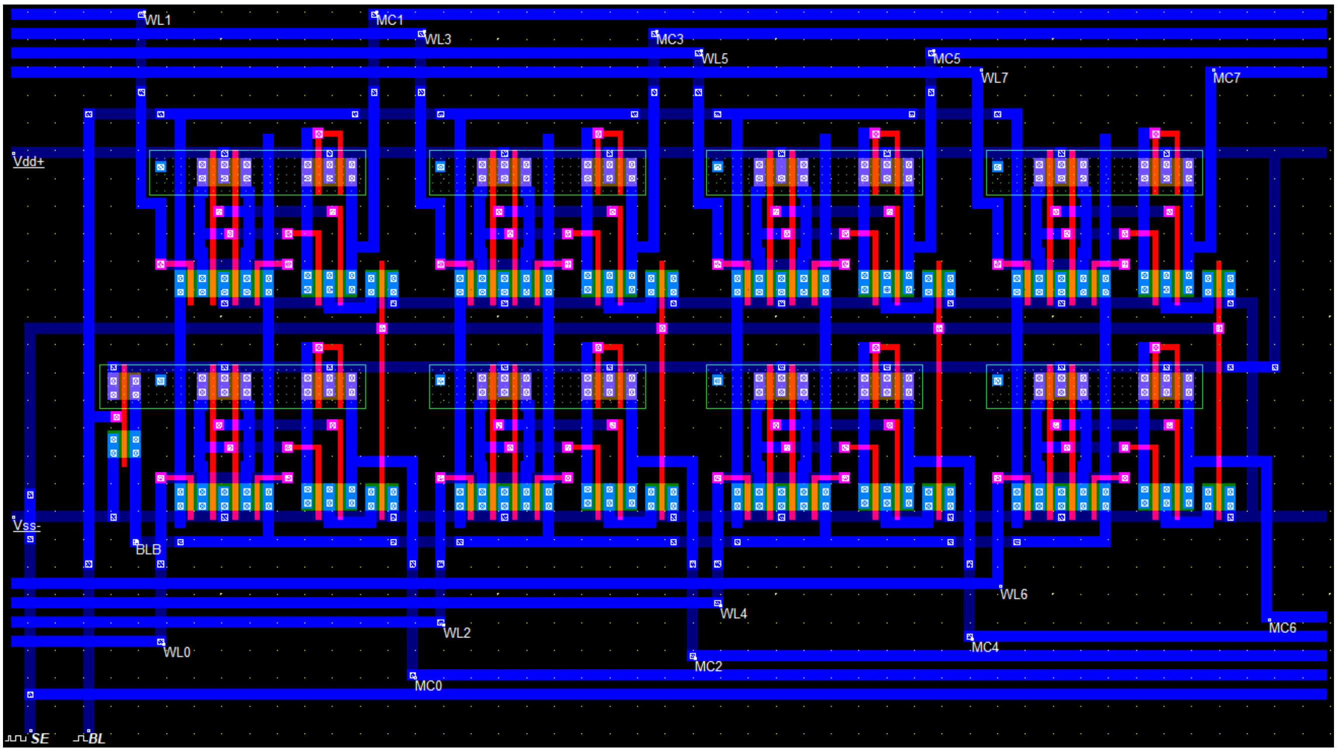


Figure 5.21. SRAM’s cells in Microwind35

5.7.3. The design of the Multiplexer

The design for the Multiplexer resembles that of the Decoder. The Multiplexer takes the eight outputs from the cells and gives only one output. It does this with the use of an eight input OR Gate on the output. With the exception of the OR GATE the Multiplexer behaves similarly to the Decoder.

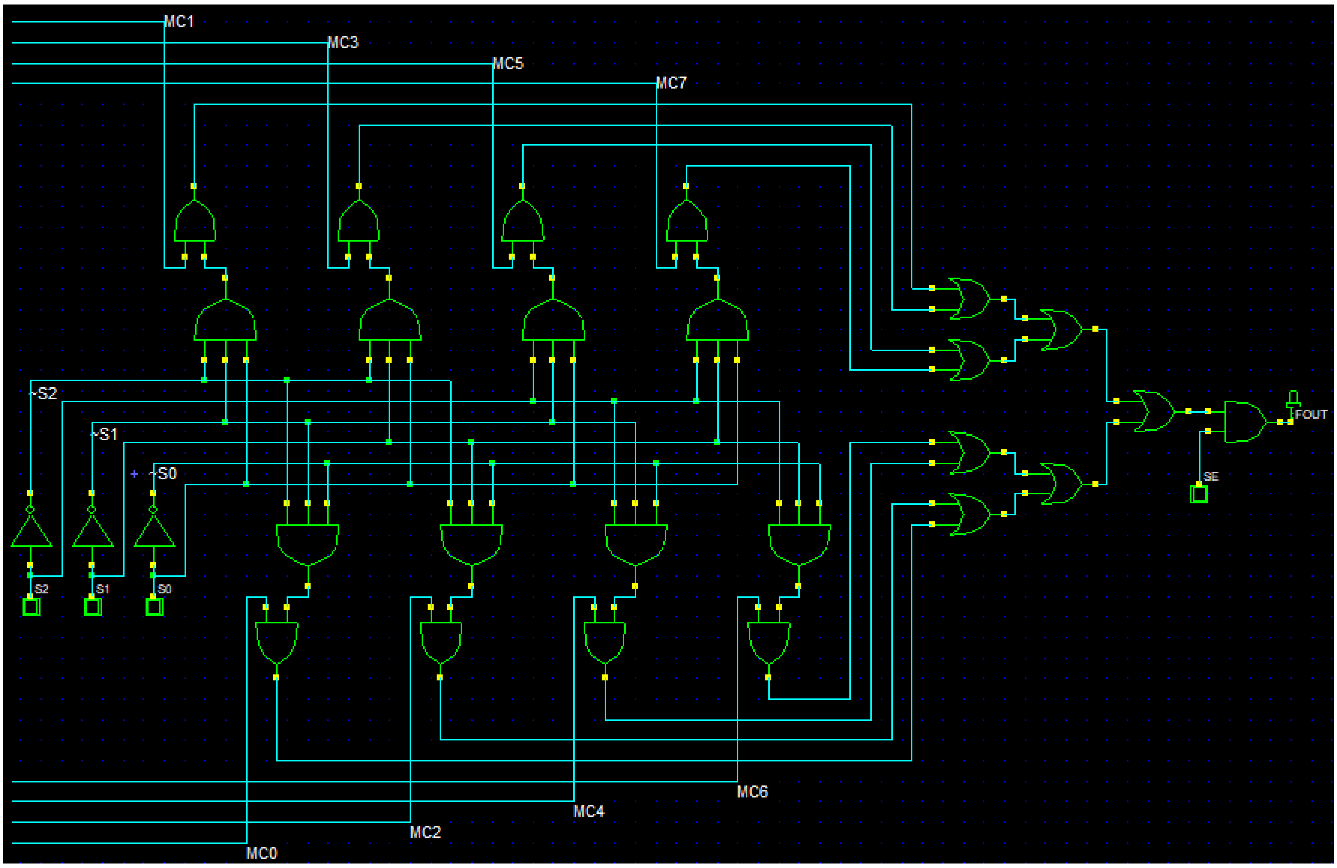


Figure 5.22. SRAM's Multiplexer in Dsch35

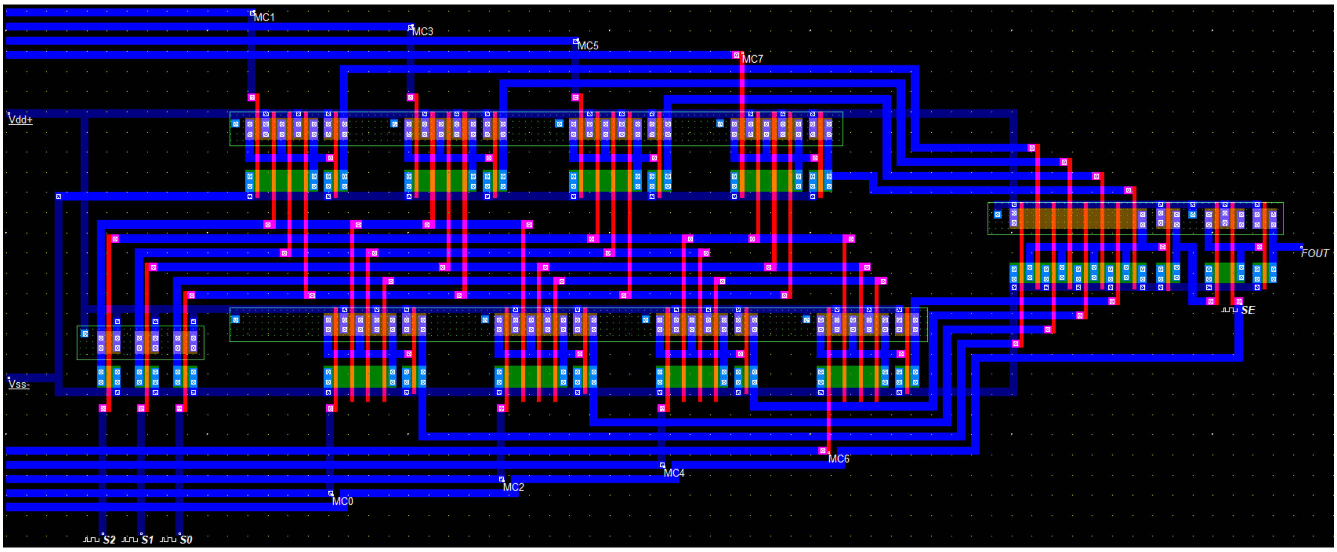


Figure 5.23. SRAM's Multiplexer in Microwind35

5.8. The simulation of the 1Byte SRAM

The figure below(Figure 5.24) shows the design of the 1Byte SRAM with all the components; the Decoder(left stage), the SRAM cells(center stage) and the Multiplexer(right stage).

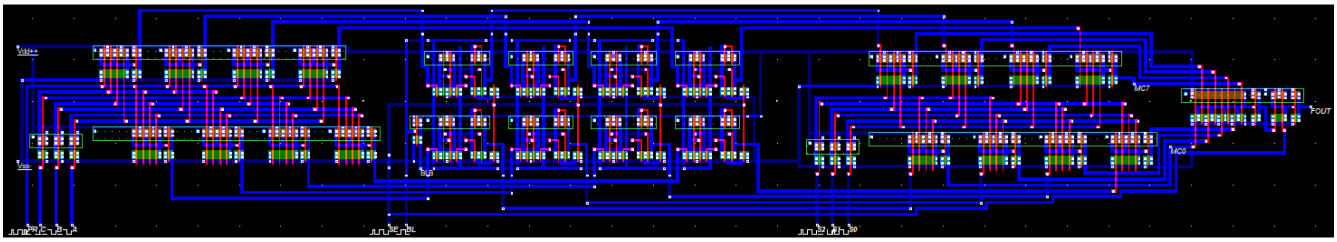


Figure 5.24. 1Byte SRAM in Microwind35

The designs of the decoder and multiplexer are very simple and suitable for this example with eight SRAM cells, but in real life applications, the SRAM cells are thousands or even more. For these kinds of SRAM designs, it might be best to have a different decoder for the cells in parallel and for the cells in series and maybe even a different component to drive these decoders and cells. The same applies for the multiplexer.

In the simulation(Figure 5.25) it is noticeable that at the timestamp 1.0ns the value ‘0’(BL=0) is written into the cell MC0 with address ‘000’ and at the timestamp 2.0ns the input ‘SE’ goes high to allow the reading of the state of the cell MC0. At the timestamp 3.0ns the value ‘0’(BL=0) is written into the cell MC7 with address ‘111’ and at the timestamp 4.0ns the input ‘SE’ goes high to allow the reading of the state of the cell MC7. At the timestamp 5.0ns the value ‘1’(BL=1) is written into the cell MC7 with address ‘111’ and at the timestamp 6.0ns the input ‘SE’ goes high to allow the reading of the state of the cell MC7. At the timestamp 7.0ns the input ‘SE’ goes high to allow the reading of the state of the cell MC0 and at the timestamp 8.0ns stays high to allow the reading of the state of the cell MC7.

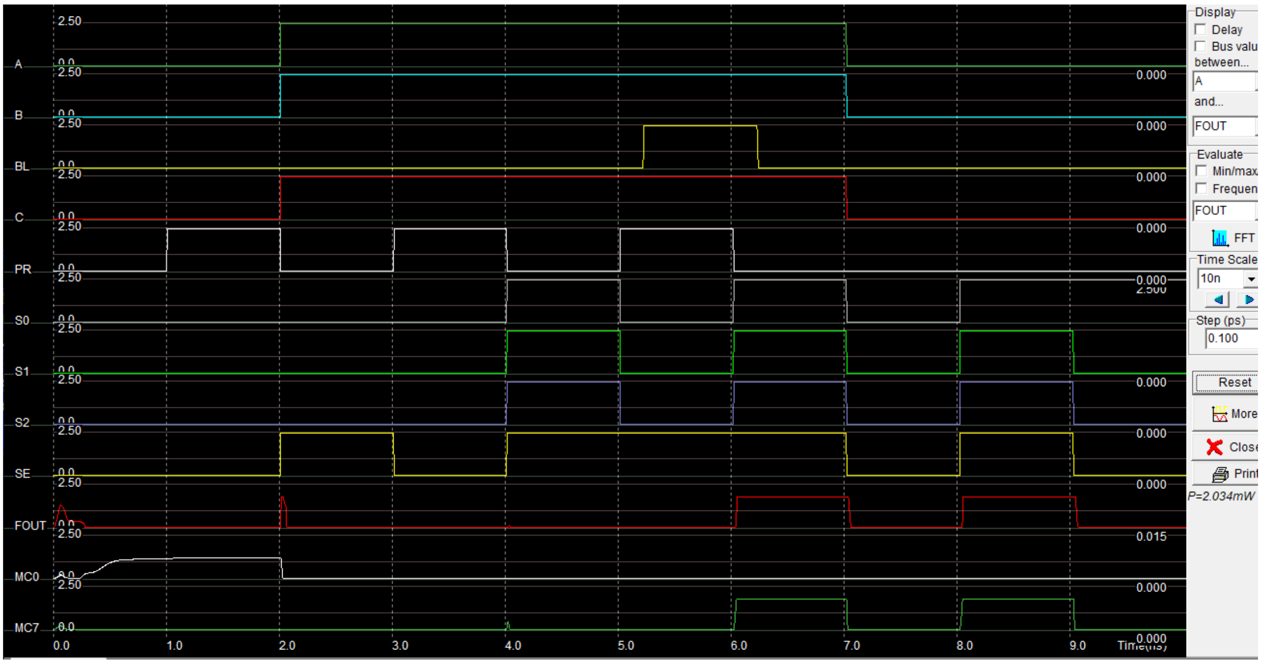


Figure 5.25. 1Byte SRAM simulation in Microwind35

6. Analog Components in VLSI designs

There are some basic analog components that are very important in CMOS designs. This chapter has some of those essential designs that are used for signal switching, amplification and regulation.

6.1. Analog Switch

By connecting a p-MOS in parallel with an n-MOS allows signals to pass in either direction with equal ease. The n-channel device carries signal current that is dependent on the ratio of input voltage to positive supply voltage, while the p-channel device carries signal current that is dependent on the ratio of input voltage to negative supply voltage or ground in single-supply designs(Figure 6.1). Because the switch has no preferred direction for current flow, it has no preferred input or output. The p-MOS and n-MOS can also be switched on and off by inverting the inputs of their gates.

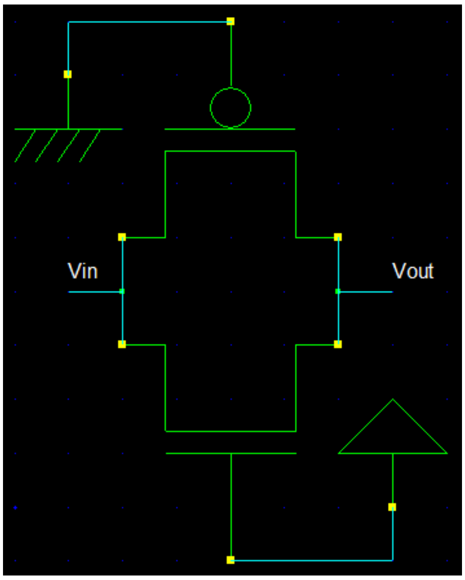


Figure 6.1. Analog Switch in Dsch35

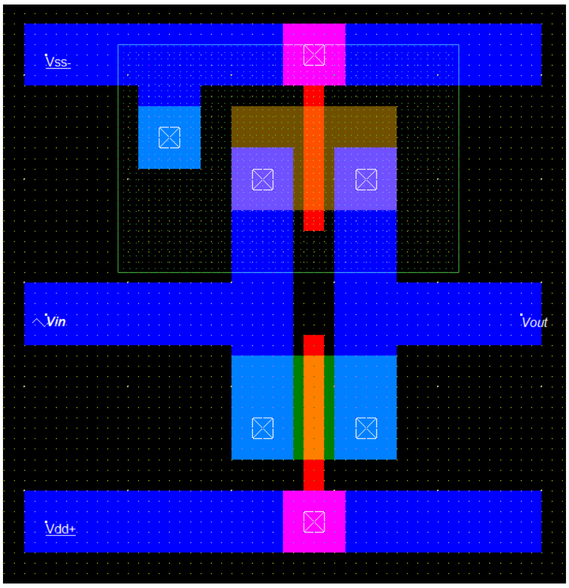


Figure 6.2. Analog Switch in Microwind35

As expected, in the simulation(Figure 6.3) the output is the same as the input signal.

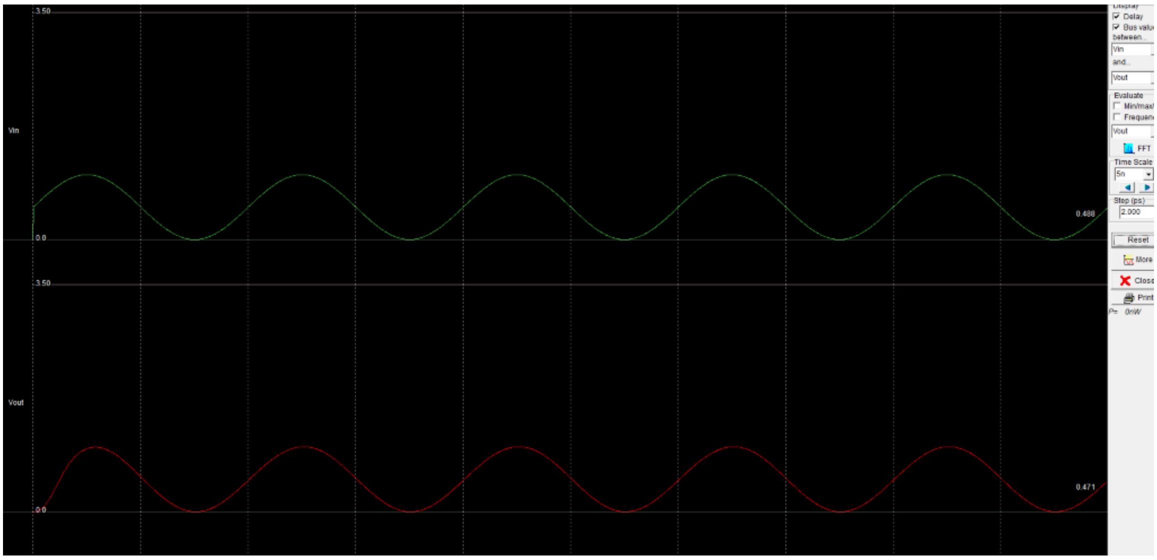


Figure 6.3. Analog Switch simulation in Microwind35

6.2. Single Stage Amplifier

Source followers exhibit high input impedance and moderate output impedance, but at the cost of nonlinearity and voltage headroom limitation. Even when biased by an ideal current source, there is input-output nonlinearity due to nonlinear dependence of Vth on the source potential.

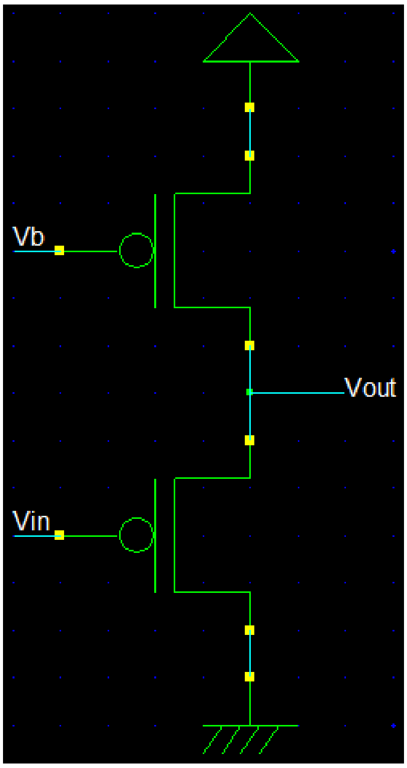


Figure 6.4. Single Stage Amplifier in Dsch35

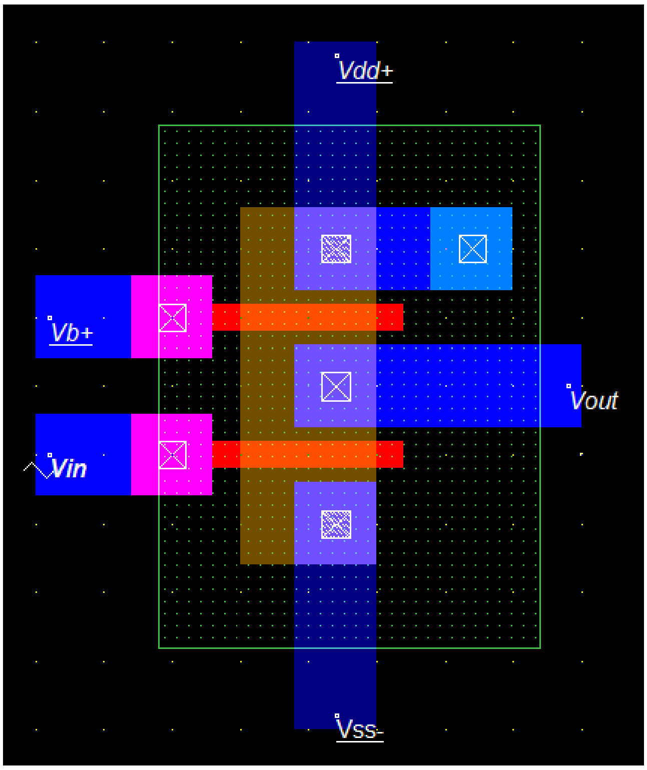
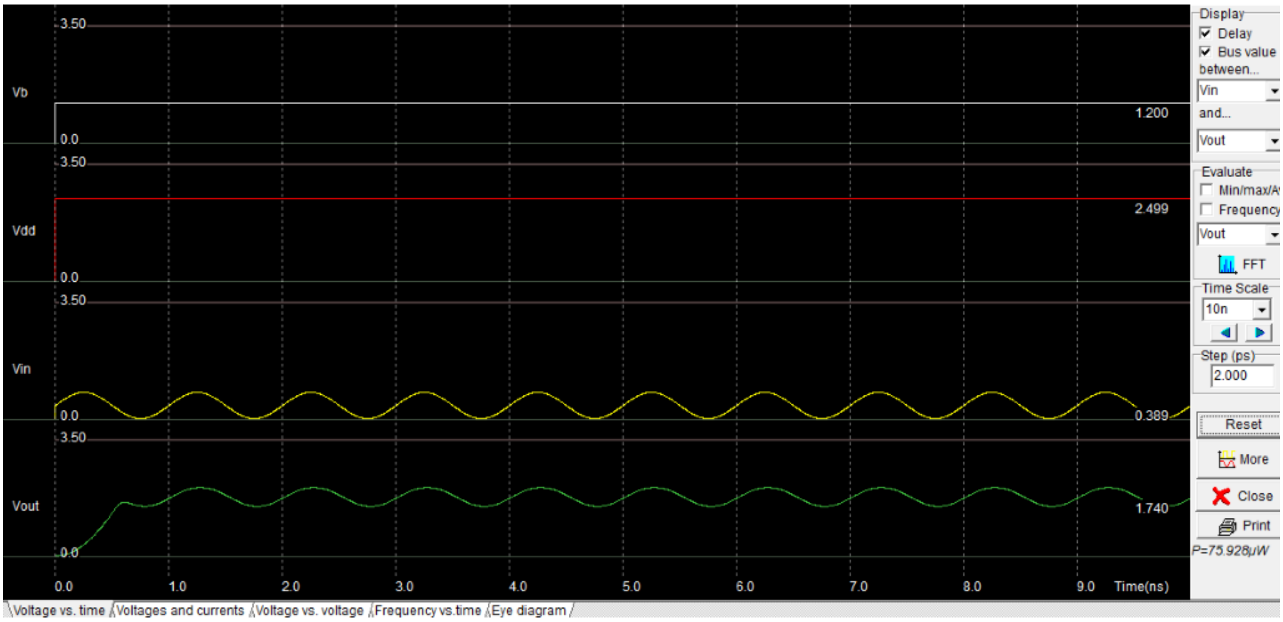


Figure 6.5. Single Stage Amplifier in Microwind35

As expected, in the simulation(Figure 6.6) the output is the same as the input signal but it's been amplified.



Display
☑ Delay
☑ Bus value between...
Vin
and...
Vout

Evaluate
□ Min/max/A
□ Frequency
Vout

Time Scale
10n

Step (ps)
2.000

P=75.928µW

Figure 6.6. Single Stage Amplifier simulation in Microwind35

6.3. Differential Amplifier

The differential amplifier is one of the most versatile circuits in analog circuit design. It is also very compatible with integrated-circuit technology and serves as the input stage to most op amps. The voltages Vin+, Vin-, Vout1 and Vout2 are called single-ended voltages as they are defined with respect to ground. The feature of the differential amplifier is to amplify only the difference between two different potentials regardless of the common-mode gain.

A characteristic that is affecting the performance of the differential amplifier is the offset voltage. In CMOS differential amplifiers, the most serious issue is the offset voltage. In the schematics below, the transistors T3 and T4 make up the differential pair, this configuration is often called a source-coupled pair. The large signal analysis begins by assuming that T3 and T4 are perfectly matched. The large signal characteristics can be developed by assuming that T3 and T4 are always in saturation, this condition is reasonable in most cases and simulates the behavior even when this assumption is not valid.

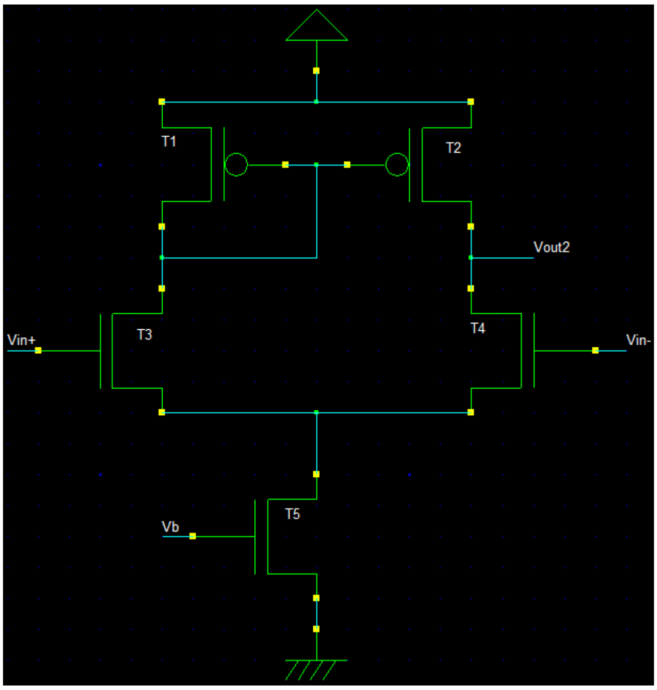


Figure 6.7. Differential Amplifier in Dsch35

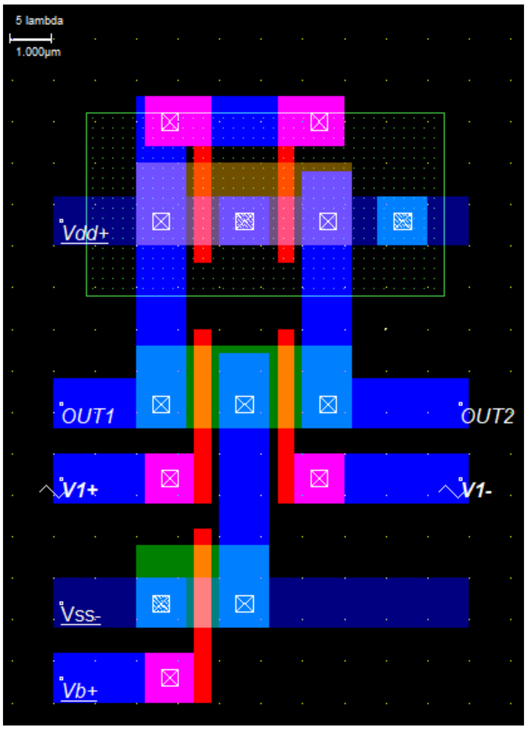


Figure 6.8. Differential Amplifier in Microwind35

In the simulation(Figure 6.9) it is noticeable that the 'OUT2' output follows(and amplifies) the difference of the two input 'V1+' and 'V1-'. The output signal can then be used to determine the state '1' when it is high and '0' when it is low.

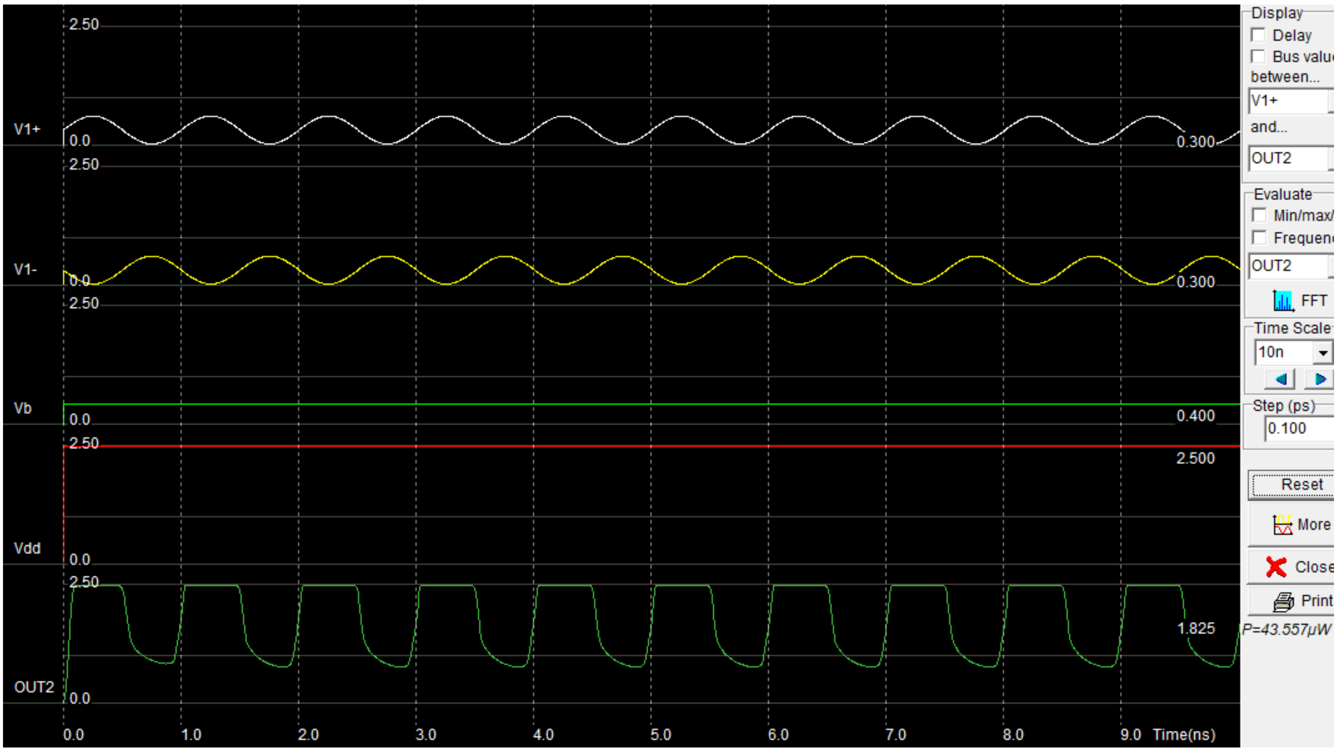


Figure 6.9. Differential Amplifier simulation in Microwind35

The simulation below shows the Gain(dB) and Phase(deg) changes in a wide range of frequencies. The simulation was obtained with the help of the Tina program.

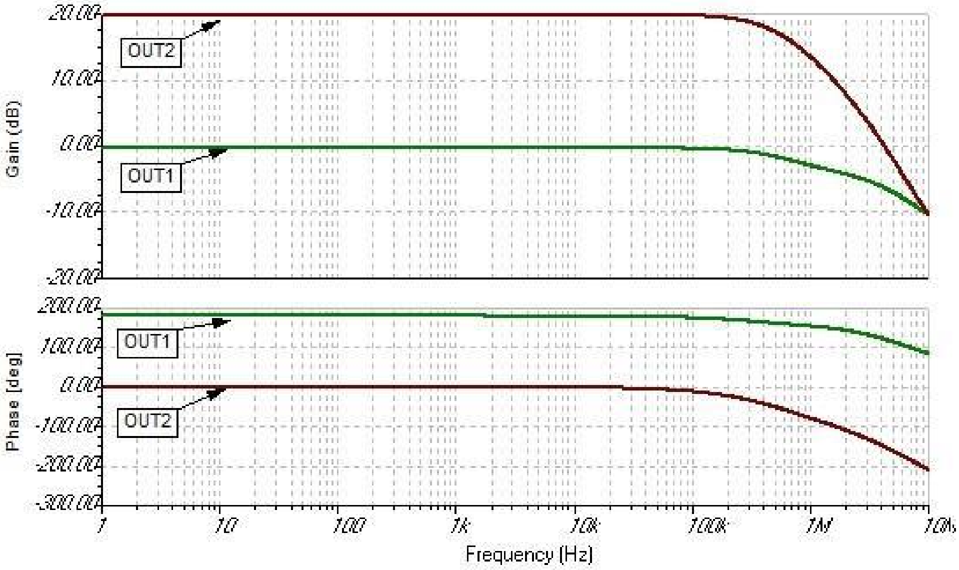


Figure 6.10. Differential Amplifier simulation in Tina9.35

6.4. Wide Range Amplifier

The same basic principles as in the Differential Amplifier(chapter 6.3) apply to the Wide Range Amplifier as well. The main difference is that this component can amplify a wide range of frequencies, with only small gain losses, throughout the band gap.

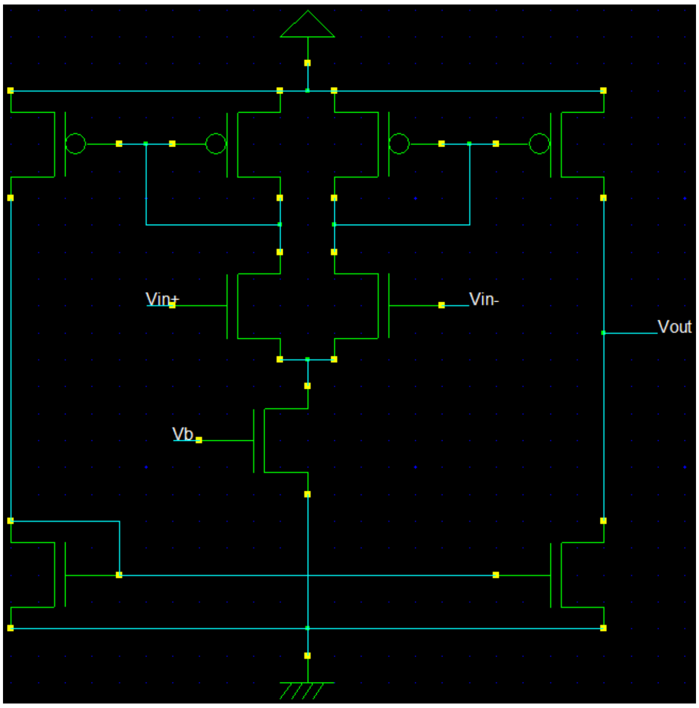


Figure 6.11. Wide Range Amplifier in Dsch35

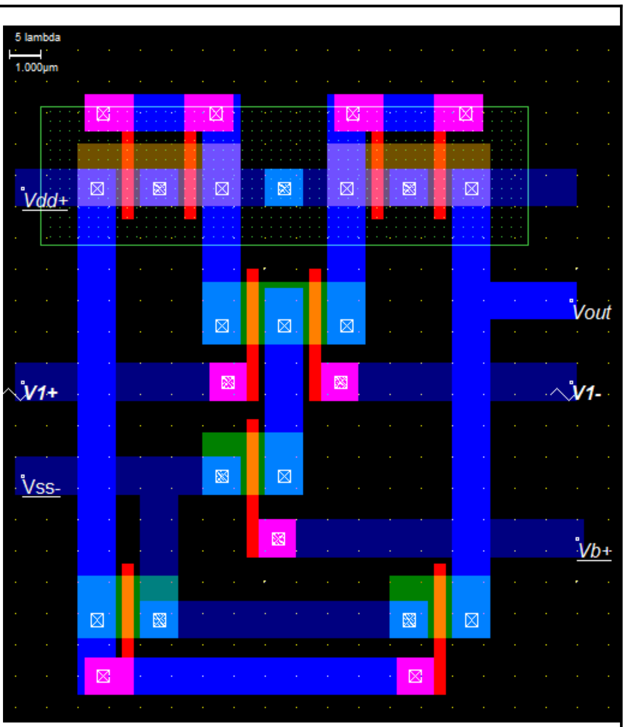


Figure 6.12. Wide Range Amplifier in Microwind35

In the simulation(Figure 6.13), as well as in simulation(Figure 6.9), it is noticeable that the 'Vout' output follows(amplifies) the difference of the two input 'V1+' and 'V1-'. The output signal can then be used to determine the state '1' when it is high and '0' when it is low.

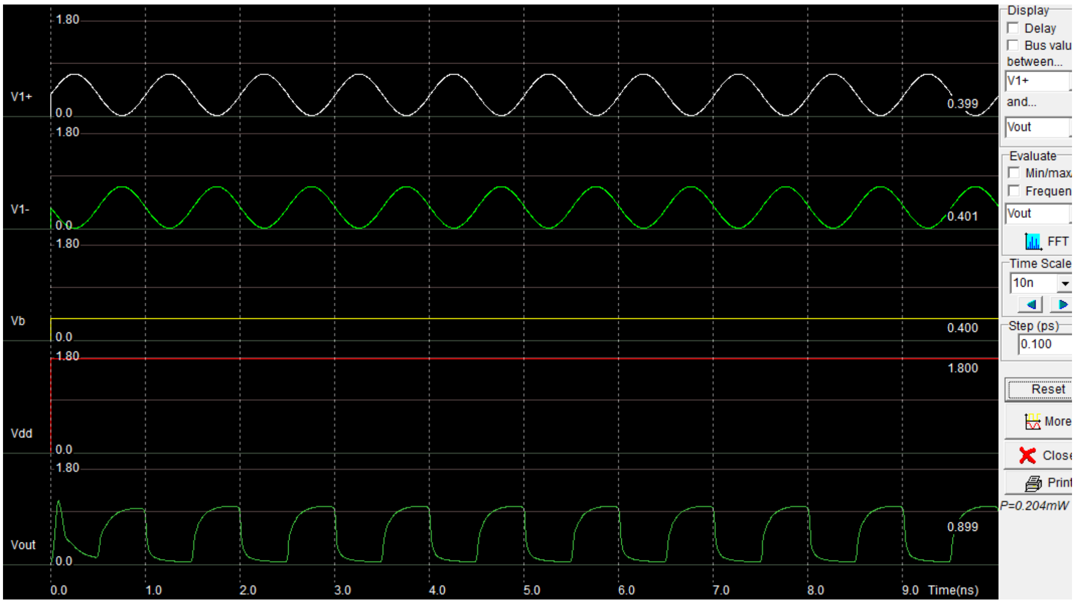


Figure 6.13. Wide Range Amplifier simulation in Microwind35

The simulation below shows the Gain(dB) and Phase(deg) changes in a wide range of frequencies. The simulation was obtained with the help of the Tina program.

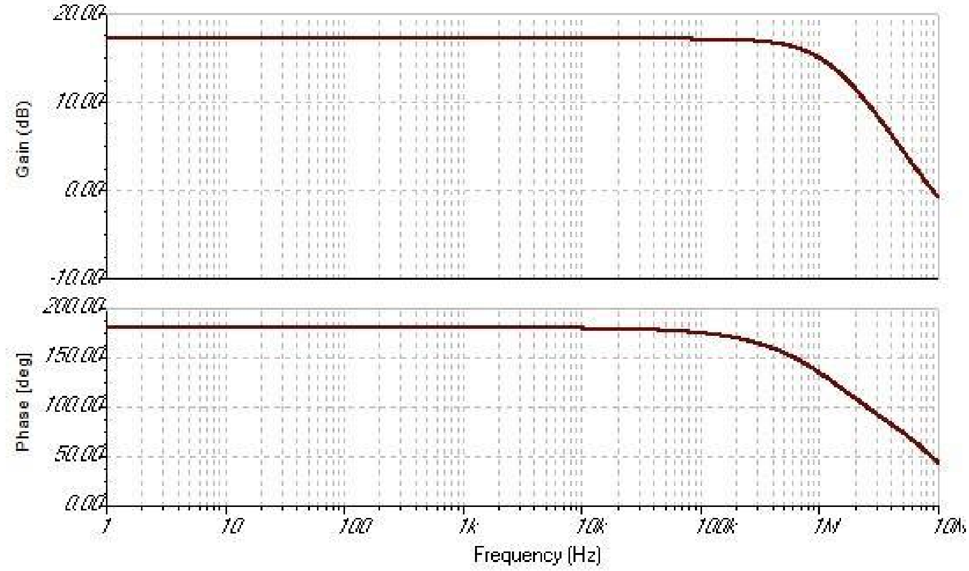


Figure 6.14. Wide Range Amplifier simulation in Tina9.35

6.5. Voltage Regulator

In the voltage shunt feedback circuit, a fraction of the output voltage is applied in parallel with the input voltage through the feedback network. This design is also known as shunt-driven shunt-fed feedback, a parallel-parallel prototype. As the feedback circuit is connected in shunt with the output and the input as well, both the output impedance and the input impedance are decreased.

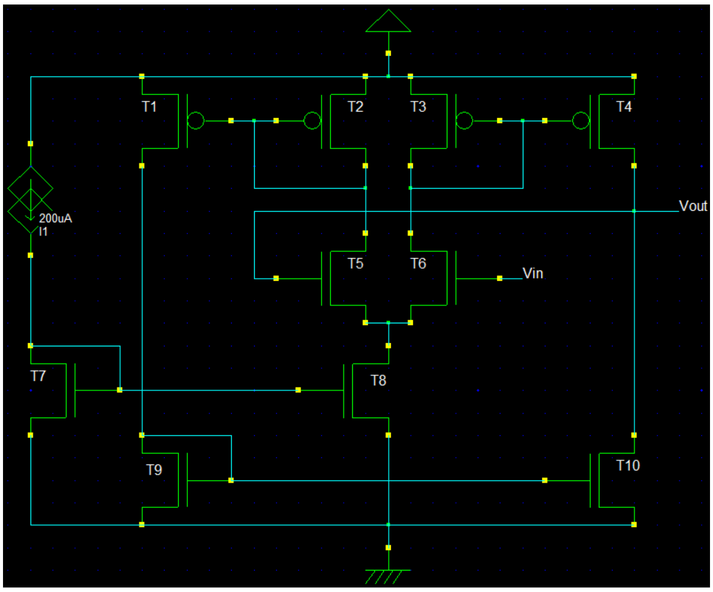


Figure 6.15. Voltage Regulator in Dsch35

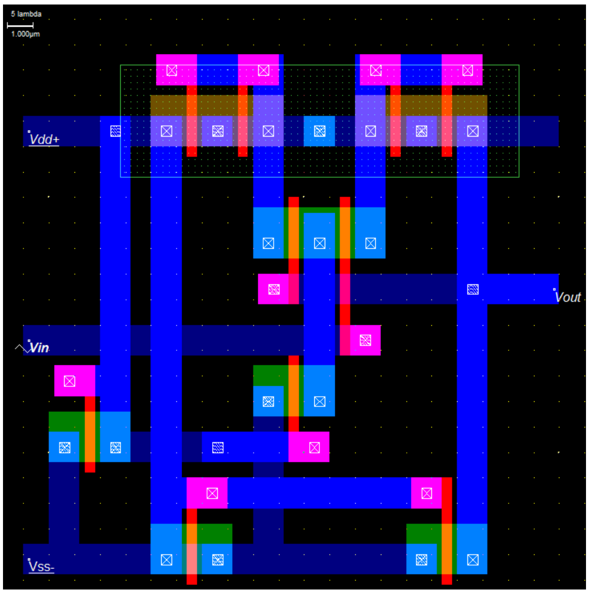
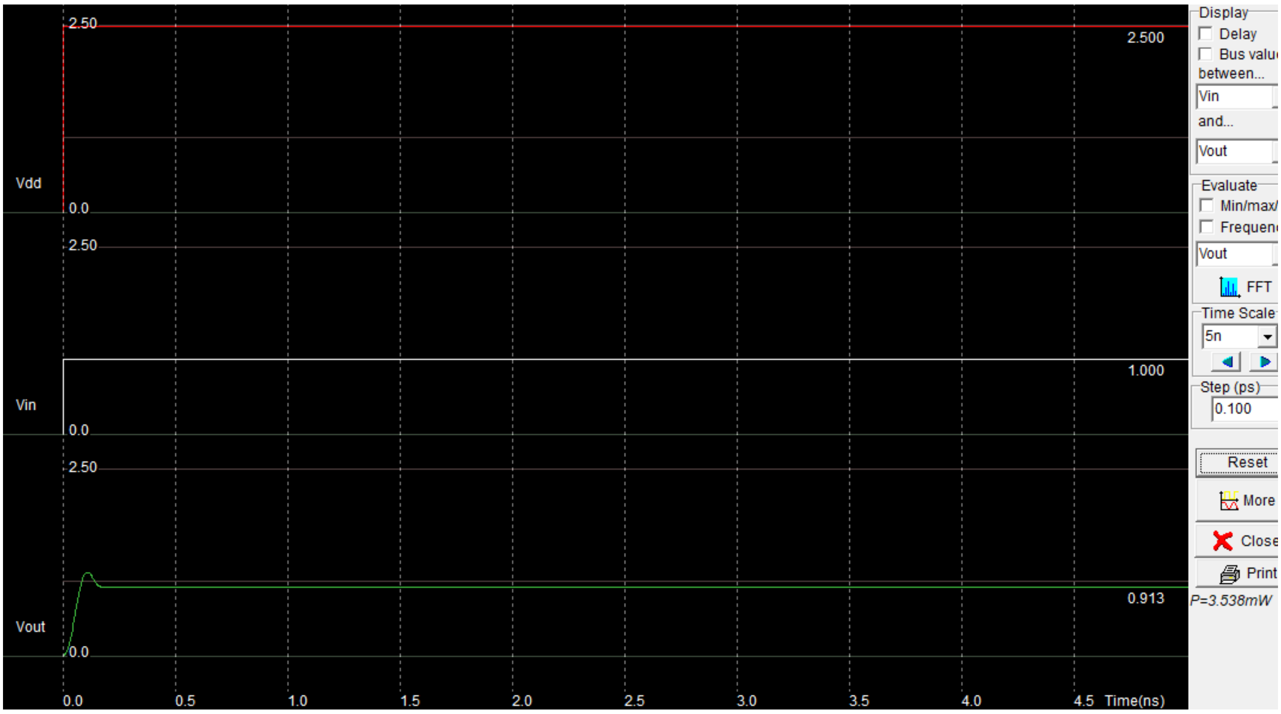


Figure 6.16. Voltage Regulator in Microwind35

The voltage at the output can change depending on the current drawn by the circuit(Figure 6.15) through the T7. The 'Vout' output follows the 'Vin' input as shown in the simulations(Figures 6.17 & 6.18) below.



between... Vin

and... Vout

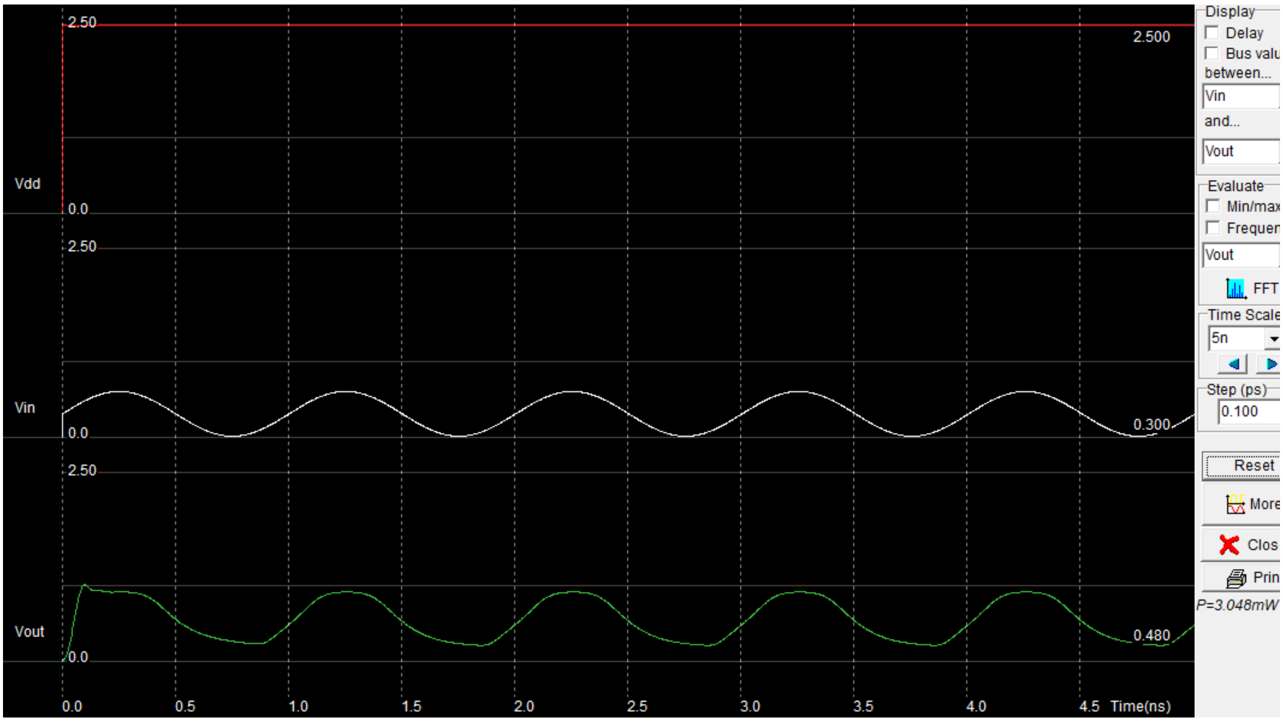
Evaluate: Vout

Time Scale: 5n

Step (ps): 0.100

P=3.538mW

Figure 6.17. Voltage Regulator first simulation in Microwind35



between... Vin

and... Vout

Evaluate: Vout

Time Scale: 5n

Step (ps): 0.100

P=3.048mW

Figure 6.18. Voltage Regulator second simulation in Microwind35

Conclusions

As this paper suggests, there are components that are easy to design and essential in an integrated circuit. There are a lot of design choices that an engineer must be able to make for every application in which the component must work. Every design has its own advantages and disadvantages, as well as its tradeoffs. Those range from the cost of production to the reliability of the component. One thing that it is certain, is that the programs which were used in this paper, are valuable tools that can help every engineer learn the basics in VLSI design and get familiarised with more advanced design choices.

Any errors or inaccuracies are of course my sole responsibility.

References

The references are numbered and ordered according to where they are first mentioned in the paper.

Bibliography - Papers - Websites

[1] L. Ockel. “A chip with microscopic transistors and circuits.” unsplash.com. https://unsplash.com/photos/qOx9KsvpqcM?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink (accessed: Jan. 26, 2021).

[2] E. Sicard. “MICROWIND.” microwind.net. https://www.microwind.net/products (accessed Jan.26, 2021).

[3] Microwind Lite. (3.5). Inst. INSA Toulouse. Accessed: Jan. 26, 2021. [Online]. Available: https://www.microwind.net/downloads

[4] Y. Calvignac, “PHYSICS AND MODELING OF SEMICONDUCTOR DEVICES 4-stage binary counter,” Aug. 2007.

[5] C. Chaduc, A. Deyts, “Random Signal Generator,” Inst. INSA Toulouse, France, Jan. 8, 2006. Accessed: Jan. 26, 2021. [Online]. Available: http://srv-wwwperso.insa-toulouse.fr/~sicard/microwind/students/students_2016/CHADUC_DEYTS.pdf

[6] S. Lara, “Program Counter and Instructions Decoder for a chip design” Univ. SFQ, Ecuador, 2017. Accessed: Jan. 26, 2021. [Online]. Available: http://srv-wwwperso.insa-toulouse.fr/~sicard/microwind/students/students_2017/Report-USFQ-Lara-Instructions-2017.pdf

[7] Wikipedia. “Static random-access memory.” wikipedia.org. https://en.wikipedia.org/wiki/Static_random-access_memory (accessed: Jan. 26, 2021).

[8] B. Wicht, “Voltage Sense Amplifiers,” in *Current Sense Amplifiers. Springer Series in Advanced Microelectronics*, vol 12. Springer, Berlin, Heidelberg, 2003, pp. 15-16. [Online]. Available: https://doi.org/10.1007/978-3-662-06442-9_3. Accessed: Jan. 26, 2021.

[9] A. S. Pavlov, “Design and Test of Embedded SRAMs,” Ph.D. dissertation, Dept. Elect. Eng., Univ. Waterloo, Ontario, Canada, 2005. [Online]. Available: https://ece.uwaterloo.ca/~cdr/pubs/Andrei_PhD_thesis.pdf

[10] McGraw-Hill Education “Chapter 3: Single-stage Amplifiers,” 2017. Accessed: Jan. 26, 2021. [Online]. Available: https://vdocuments.mx/chapter-3-single-stage-amplifiers-copyright-2017-mcgraw-hill-education-all.html

[11] Κ. Θ. Ζήκος, “Design and analysis of static random access memory circuits in TINA DesignSoft software” Dept. Elect. Eng., Univ. West Attica, Greece, Jan. 2019. Accessed: Jan. 26, 2021. [Online]. Available: http://okeanis.lib2.uniwa.gr/xmlui/bitstream/handle/123456789/5158/eee_06515.pdf?sequence=1&isAllowed=y

[12] S. L. Tripathi, T. S. Kumar, “Implementation of CMOS SRAM Cells in 7, 8, 10 and 12-Transistor Topologies and their Performance Comparison,” IJEAT, vol 8, no 2S2, pp. 227-229, Jan. 2019, [Online]. Available: https://www.researchgate.net/publication/334362460_Implementation_of_CMOS_SRAM_Cells_in_7_8_10_and_12-Transistor_Topologies_and_their_Performance_Comparison (accessed: Jan. 26, 2021).

[13] P. E. Allen, D. R. Holberg, “CMOS Amplifiers” in *CMOS Analog Circuit Design*, 2nd ed, pp. 167-201, 2002, [Online]. Available: https://www.scribd.com/document/266455556/Allen-holberg-CMOS-Analog-Circuit-Design Accessed: Jan. 26, 2021.

[14] U. Munir, D. Canny, “Selecting the Right CMOS Analog Switch,” APP 5299, Jan. 29, 2013. Accessed: Jan. 26, 2021. [Online]. Available: https://pdfserv.maximintegrated.com/en/an/AN5299.pdf