Capitolul 7.
Circuite de baza.

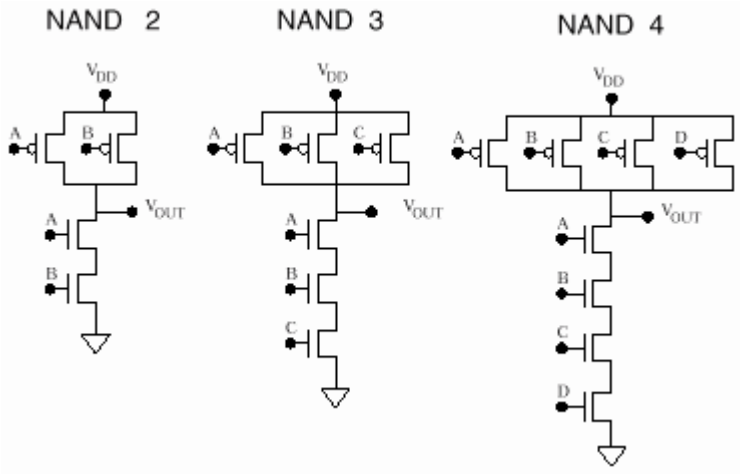
In acest capitol se vor prezenta circuitele standard CMOS sub forma de scheme si masti. Informatiile referitoare la mastile necesare in procesul de generare a diverselor straturi ale structurilor circuitelor CMOS numerice vor fi, intr-o mare masura, simplificate. Se va putea observa ca, in alcatuirea portilor de baza, se intalnesc microcelule elementare analoage limbajului de programare sau microcodului.

Portile standard.

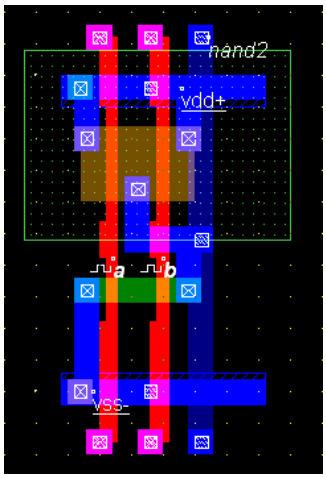
Cand se deseneaza un circuit complex, cea mai buna metoda consta intr-o abordare ierarhica. Spre exemplu, atunci cand se proiecteaza un microprocesor nu se va pleca de la nivelul tranzistoarelor, ci de la nivelul portilor de tipul NAND, NOR, si de la cel al bistabilelor. Pe aceasta baza se vor crea nivelurile superioare ale structurii: registrele, unitatile aritmetice-logice s.a. Aceasta abordare "de jos in sus" reprezinta un mijloc de a solutiona probleme cu caracter complex. In aceasta sectiune se vor prezenta cateva circuite standard si structuri de porti. Pe de alta parte, avand create aceste module sau megacelule, proiectele urmatoare se vor baza pe aceste structuri in cadrul unei abordari "de sus in jos".

Circuitul NAND.

Schema pentru poarta NAND a fost prezentata in primul capitol. Figura de mai jos prezinta schemele pentru portile NAND cu doua, trei si patru intrari. Tranzistoarele PMOS au sursele conectate la V_{DD} si drenele la V_{OUT}. Tranzistoarele NMOS sunt conectate in serie, cel de la nivelul cel mai de jos avand sursa conectata la masa, iar cel de la nivelul cel mai de sus drena conectata la V_{OUT}.



Desenul mastilor portii NAND cu doua intrari este aratat mai jos. Aceasta celula poseda cateva trasaturi notabile. Mai intai, traseele V_{DD} si GND formeaza cate o bara care se extinde transversal la nivelul superior si la nivelul inferior. Aceasta permite plasarea alaturata a unor asemenea celule. In acest mod se vor forma liniile de alimentare si de masa, care se extind de la stanga la dreapta, fara alte conexiuni externe. Ca rezultat, dimensiunea liniei si spatiile intre ele trebuie standardizate.



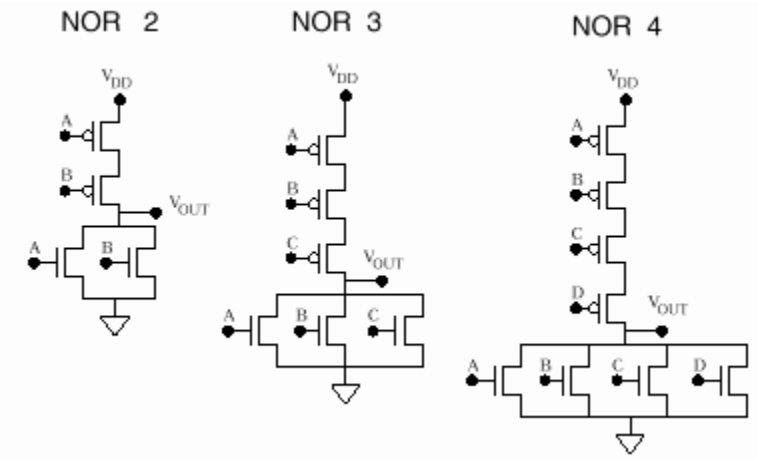
Acest tip de celula este folosit in programele de autorutare. Intrarile a si b sunt conectate la trasee din siliciu policristalin, care formeaza tranzistoarele PMOS si NMOS. Cele doua tranzistoare PMOS sunt adiacente la sursele lor, pentru a economisi spatiul si pentru a imbunatati performanta. O strategie similara de conectare va fi utilizata si pentru cele doua tranzistoare NMOS. Se poate observa ca dispozitivele PMOS se afla pe aceeasi insula N, care are, de asemenea, un contact la Vdd.

Circuitul NOR.

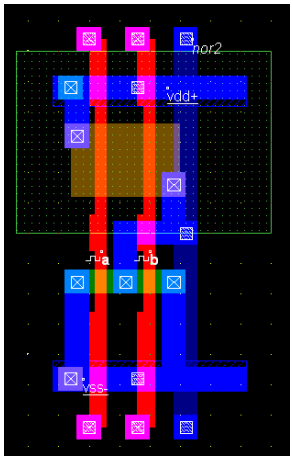
Schemele pentru portile NOR cu doua, trei si patru intrari sunt prezentate mai jos. Dupa cum se poate observa tranzistoarele PMOS sunt conectate in serie iar cele NMOS sunt conectate in paralel.

Desenul mastilor pentru poarta NOR cu doua intrari prezinta aceleasi caracteristici ca si cel pentru poarta NAND. Trebuie remarcat faptul ca tranzistoarele de tip P, fiind in serie, trebuie sa aibe dimensiuni mai mari decat tranzistoarele de tip N, pentru a asigura timpi de crestere si cadere egali. Explicatia acestei afirmatii s-a dat intr-un capitol anterior.

Un alt aspect se refera la faptul ca schema circuitului la nivelul tranzistoarelor reprezinta amplasarea relativa a acestora din urma. Desenele reale ale mastilor incearca, de regula, sa



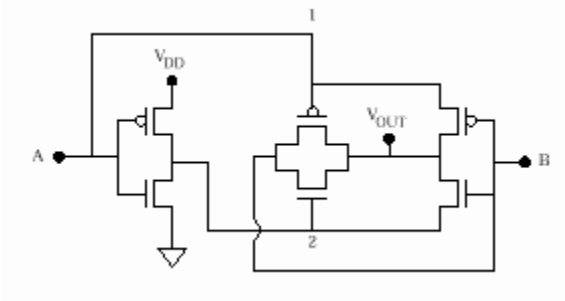
minimizeze aria ocupata. In acest scop se utilizeaza conexiuni si orientari ale tranzistoarelor cat mai convenabile.



SAU-Exclusiv (XOR).

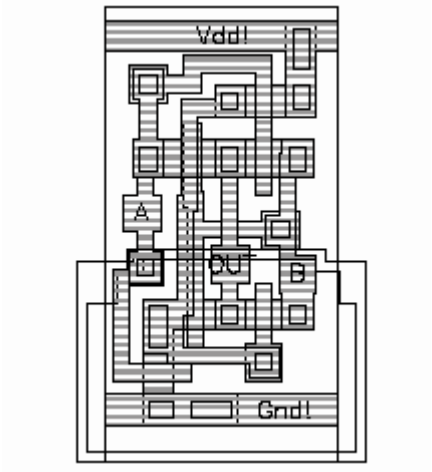
Poarta XOR are numeroase utilizari in proiectarea circuitelor numerice. Functia indeplinita de acest circuit nu se realizeaza direct, ca in cazul circuitului NAND. De aceea, vor fi prezentate doua versiuni ale circuitului XOR. Prima versiune se caracterizeaza printr-o arie ocupata mai mica, si prin folosirea portilor de transmisie. Intrucat puterea de comanda la iesire deriva din intrari, circuitul mai poarta numele de XOR pasiv. Cel de-a doua configuratie se bazeaza pe porti statice CMOS. Aceasta solutie asigura un timp de raspuns mai rapid, dar utilizeaza mai multe tranzistoare in proiectare.

In figura de mai jos se prezinta schema pentru poarta XOR pasiva. Se poate observa inversorul de pe intrarea A, structura care seamna cu un inversor pe intrarea B si poarta de transmisie in centrul circuitului. In continuare se va face o analiza a celor patru cazuri posibile, pentru semnalele logice de la intrare.



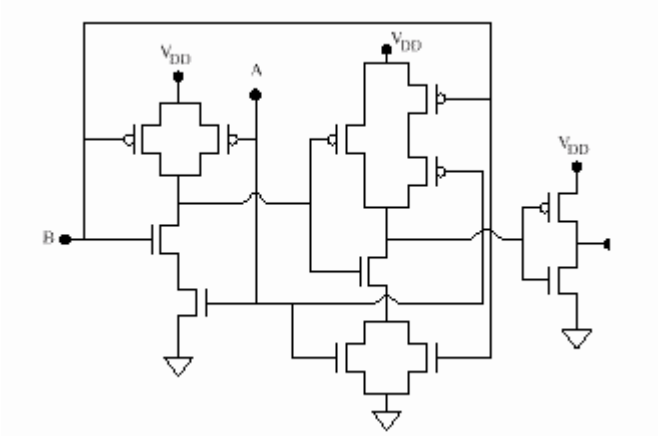
- A=0, B=0. Aceasta face ca punctul 1 sa aibe valoarea logica 0 si punctul 2 – valoarea 1, ceea ce va face ca poarta de transmisie sa fie in conductie. Prin aceasta se creaza o cale de la B la *VOUT*, prin poarta de transmisie. Intrucat B este la 0 sau la masa, *VOUT* =0
- A=0, B=1. Din nou punctele 1 si 2 au valorile 0 si respectiv 1. Poarta de transmisie conduce ceea ce face ca intrarea 1, de la B, sa se propage catre *VOUT*. Astfel, *VOUT* = 1.
- A=1, B=0. Punctele 1 si 2 au valorile 1 si respectiv 0. Poarta de transmisie este blocata, iar inversorul corespunzator intrarii B va fi operational. Deoarece B=0 si A=1, iesirea *VOUT* va avea valoarea 1.
- A=1, B=1. Acest caz este similar cu cel de mai sus, cu mentiunea ca inversorul corespunzator intrarii B va face ca *VOUT* = 0.

Aceasta poarta utilizeaza numai 6 tranzistoare, dupa cum se poate vedea din figura de mai jos.

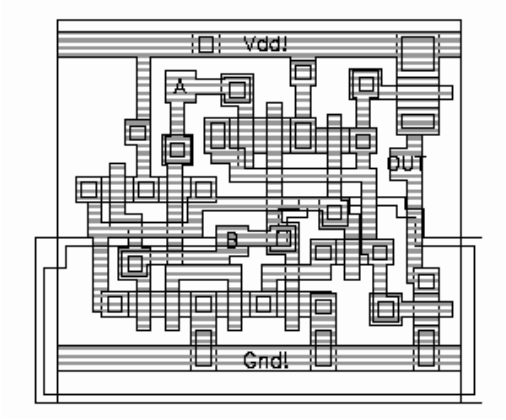


Cealalta implementarea, care utilizeaza 12 tranzistoare, are o performanta mai buna, dar este mai costisitoare, ca hardware. Prima sectiune a schemei poate fi recunoscuta ca o poarta NAND.

Sectiunea din mijloc a schemei implementeaza functia NOT((A∪B).F) unde F=NOT(A.B). Ultima sectiune reprezinta un inversor. Aceasta structura permite ultimului inversor sa realizeze o comanda mai puternica decat in cazul proiectului anterior, bazat pe poarta de transmisie. In figurile de mai jos se prezinta schema si mastile acestui circuit.In acest caz curentul de comanda de la iesire este furnizat de sursa de alimentare, in timp ce pentru prima solutie acesta era asigurat de intrarile A si B.



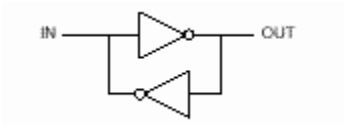
a) schema circuitului XOR activ



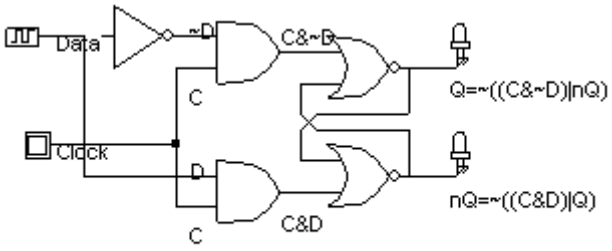
b) planul mastilor circuitului XOR activ

Elemente de memorare, registre de deplasare, bistabile.

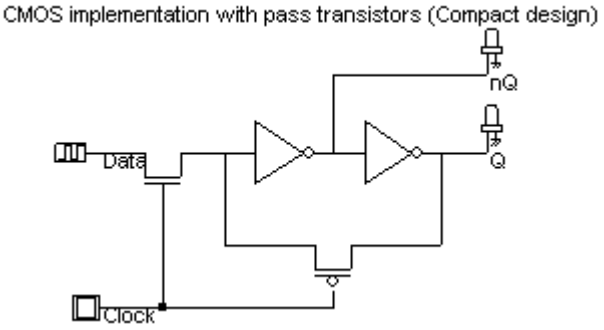
La implementarea logicii secventiale sunt necesare elemente de memorare pentru a stoca rezultatele. Aceste elemente stocheaza un bit si pot fi organizate pentru a forma structuri mai mari. Cel mai simplu element de memorare este latch-ul format din doua inversoare, ca in desenul de mai jos, unde iesirea OUT va avea o valoare constanta, fie 1, fie 0. Pentru a forta o noua informatie intrarea trebuie sa depaseasca reactia latch-ului. Semnalul de reactie trebuie sa fie mai slab, in ceea ce priveste comanda, decat semnalul de intrare.



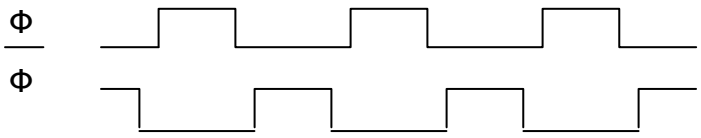
Nota 1. Exemplificari.



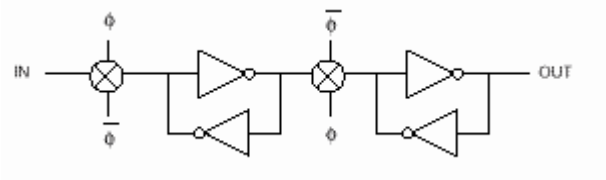
D-latch schematic diagram



In figura de mai jos se arata modul in care poate fi realizat un registru de deplasare, folosind latch-uri. Un element de memorare se poate realiza cu doua latch-uri si doua porti de transmisie. Portile de transmisie vor fi controlate cu impulsuri de ceas complementare Φ si $\overline{\Phi}$. Pentru efectuarea deplasarii cu un rang trebuie aplicat semnalul de ceas.

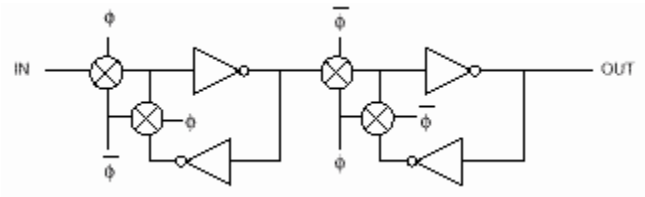


a) semnale de ceas bifazic, fara suprapunere.



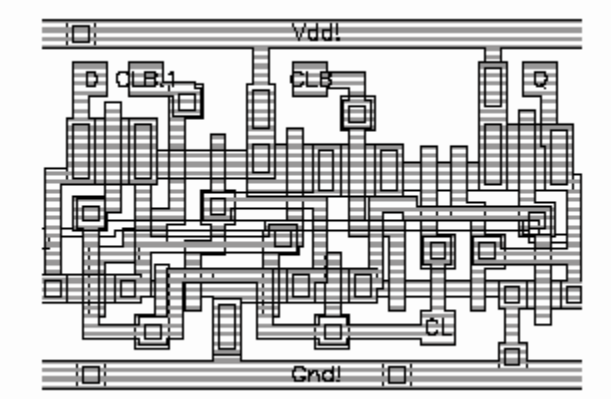
b) element de memorare cu latch-uri si porti de transmisie.

In figura urmatoare se prezinta schema unui bistabil de tip D realizat cu doua latch-uri si doua porti de transmisie controlate cu semnale de ceas bifazic.

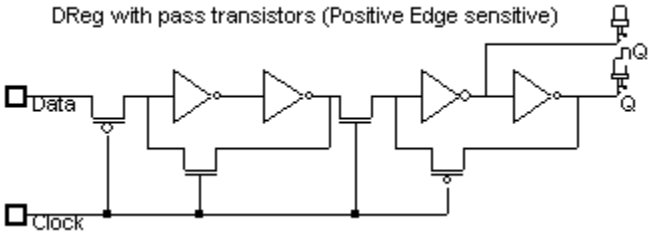


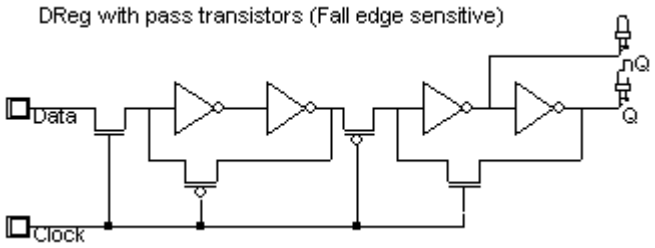
Cand semnalul de ceas este pe nivel coborat , poarta de transfer este blocata, calea de reactie e ramane activa, iar latch-ul pastreaza valoarea memorata. Cand semnalul de ceas este pe nivel ridicat, calea de reactie este blocata, iar calea de intrare este activa, astfel ca, o noua valoare a semnalului poate fi stocata. Pentru a realiza un registru de deplasare se pot conecta in cascada astfel de celule.

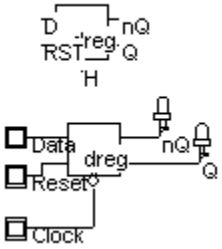
In figura de mai jos se prezinta planul mastilor pentru un asemenea bistabil, de tip D.

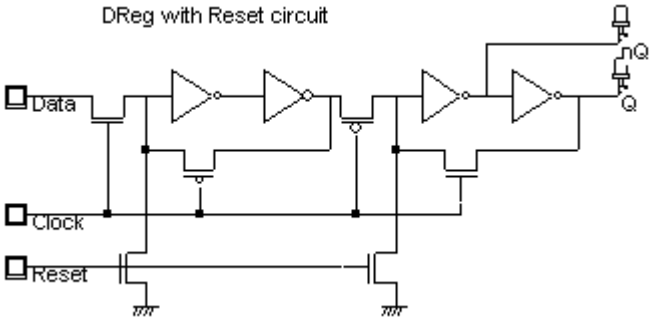


Nota 2. Implementari cu tranzistoare de trecere.





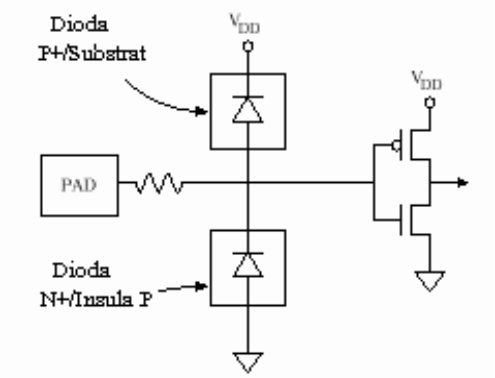




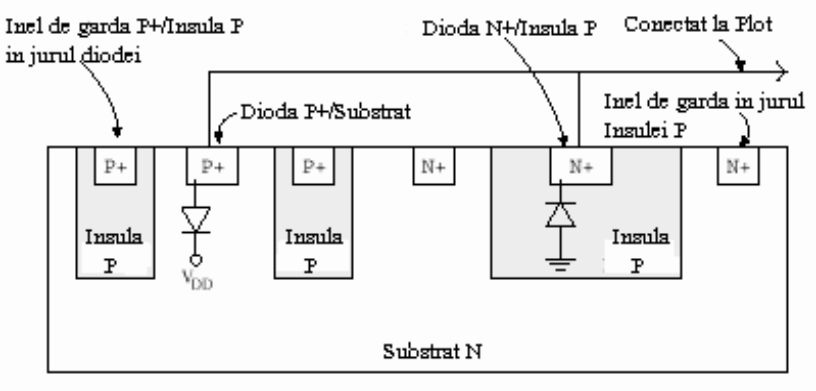
Ploturi de Intrare/Iesire.

Intrare.

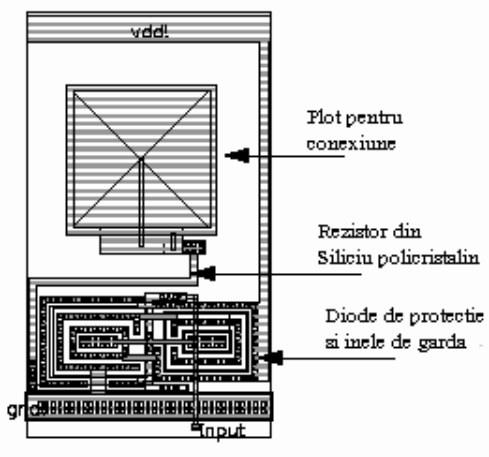
Atunci cand se realizeaza un plot de intrare trebuie sa se aibe in vedere protectia circuitului. Variatiile de tensiune pot conduce la defectarea circuitului. Pentru a preveni asemenea situatii trebuie utilizata o schema de protectie ca in figura de mai jos.



Rezistorul este realizat din siliciu policristalin si are o valoare a rezistentei de 500 – 3000 Ω. Daca tensiunea de intrare depaseste V_{DD}, dioda superioara va conduce la limitarea tensiunii de intrare a portii la valoarea V_{DD}. In mod asemanator, daca tensiunea de intrare scade sub Vss, dioda inferioara va conduce limitand inferior tensiunea de intrare a portii la Vss. In cazul unui proces cu insula P, dioda superioara este creata prin plasarea unei regiuni P^+ pe un substrat de tip N . Dioda inferioara este realizata prin formarea unei regiuni N^+ pe o insula P. Pentru a realiza protectia fata de curentii din substrat se creaza un inel de protectie in jurul diodelor. In figura de mai jos se prezinta o sectiune transversala prin structura de protectie.

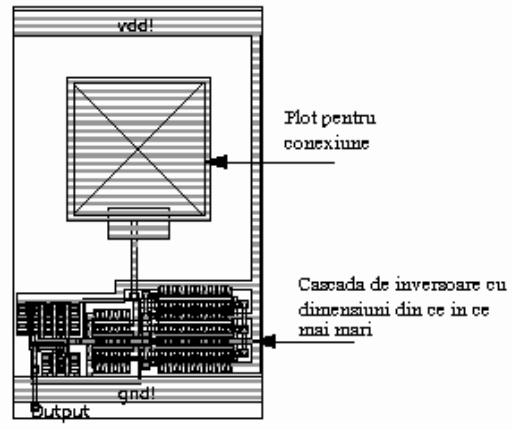


In figura urmatoare se prezinta planul mastilor unui plot de intrare. Dioda polarizata la VDD se afla in stanga jos, iar dioda polarizata de Vss este plasata in dreapta sus.



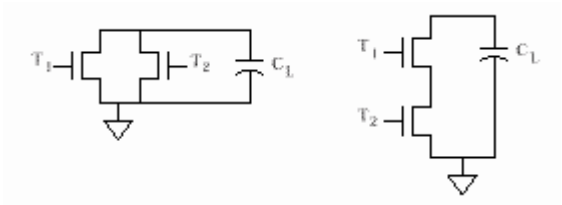
Iesire.

Un plot de iesire trebuie sa aibe o capacitate de comanda a unei sarcini externe astfel incat timpii de crestere si de cadere sa nu fie excesiv de mari. Configuratia uzuala se bazeaza pe mai multe inversoare, cu dimensiuni din ce in ce mai mari, conectate in cascada. Raportul optim pentru dimensiuni este de circa 2,7. Planul mastilor pentru un plot de iesire este dat mai jos. In stanga jos se afla unul dintre cele doua inversoare ale acestui plot de iesire. In dreapta jos este plasat un alt inversor cu dimensiuni mai mari decat cel din stanga.



Performanta.

Proiectarea portilor logice trebuie sa aibe in vedere si performanta. In exemplele de porti NAND si NOR, tranzistoarele erau conectate in serie si in paralel. In figura de mai jos se prezinta doua tranzistoare NMOS conectate in paralel si in serie, prin care trebuie sa se descarce sarcina capacitiva C_L.

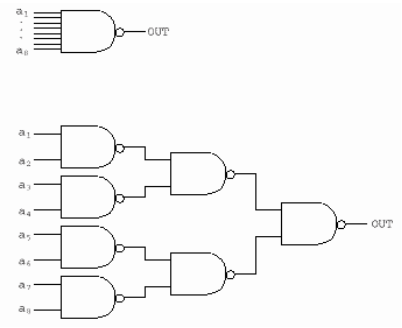


Cand ambele tranzistoare sunt deschise ele vor opera in regiunea liniara si se vor comporta ca niste rezistoare. In cazul tranzistoarelor in serie descarcarea va fi mai lenta deoarece rezistentele celor doua tranzistoare se aduna. In cazul legarii in paralel rezistenta va fi mai mica decat cea a unui tranzistor. Astfel, timpul necesar descarcarii unui capacitor va fi dat de

$$T_{fSerie} \sim m.T_f \text{ si } T_{fParalel} \sim T_f/m$$

Unde *m* este numarul de tranzistoare. Trebuie aratat ca relatiile se pastreaza daca tranzistoarele comuta simultan.

In figura urmatoare se prezinta doua modalitati pentru realizarea unei porti NAND cu 8 intrari. Prima consta in construirea unei porti mari cu 8 tranzistoare PMOS si 8 tranzistoare NMOS. Cea de-a doua modalitate se bazeaza pe utilizarea unor porti NAND mai mici, care sunt conectate in cascada. La prima vedere s-ar parea ca poarta unica cu 8 intrari va avea timpi mai mici de crestere si cadere. In realitate lucrurile stau exact invers, deoarece poarta unica presupune conectarea tranzistoarelor NMOS in serie, ceea ce va duce la cresterea timpului de cadere. Cu toate ca cele de mai sus sunt adevarate, se impune efectuarea unei simulari pentru a trage concluziile corecte.



ANEXE.

Latches & Memories.

This chapter details the structure and behavior of latches and memory circuits. The RS Latch, the D Latch and the edge-sensitive register are presented. Then , the concepts of ROM, static RAM and dynamic RAM memories are introduced, together with simulations.

RS Latch

The RS Latch, also called Set-Reset Flip Flop (SR FF), transforms a pulse into a continuous state. The RS latch can be made up of two interconnected NAND gates. In that case, the Reset and Set inputs are active low. The memory state corresponds to Reset=Set=1. The combination Reset=Set=0 should not be used, as Q=nQ=1. Furthermore, the simultaneous change from Reset=Set=0 to Reset=Set=1 provokes what is called the metastable state, that corresponds to a parasitic ring effect that may jeopardize the behavior of the whole circuit.

RS Latch (NAND)

R	S	Q	nQ
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Q	nQ

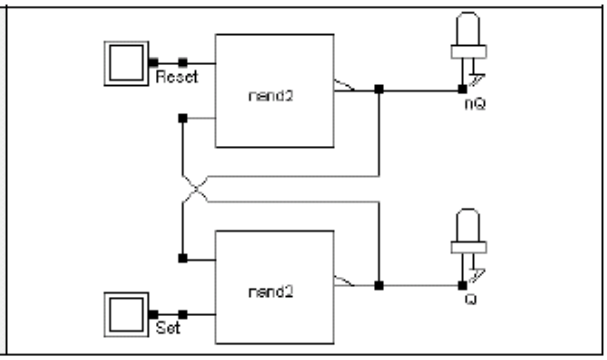


Fig. 7-1. The truth table and schematic diagram of a RS latch made (RSNor.SCH)

Verilog description:

```
module RSNor( Reset,Set,Q,nQ);
input Reset,Set;
output Q,nQ;
nor nor1(Q,nQ,Reset);
nor nor2(nQ,Set,Q);
endmodule
```

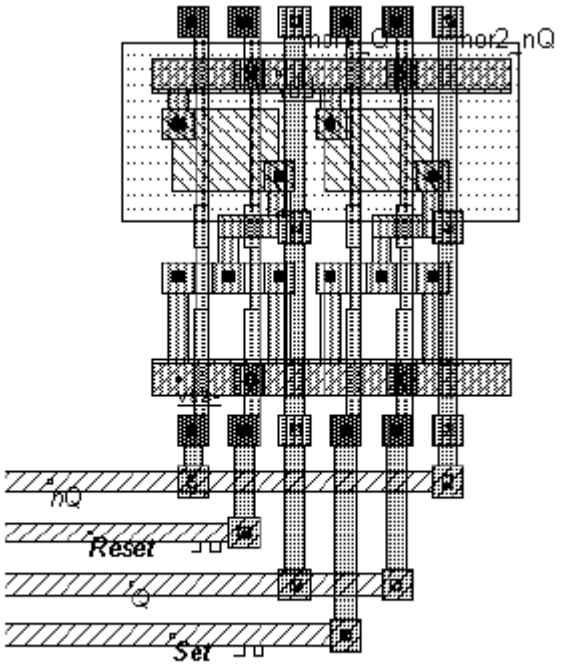


Fig. 7-2. Layout of the RS latch made (RSNor.MSK)

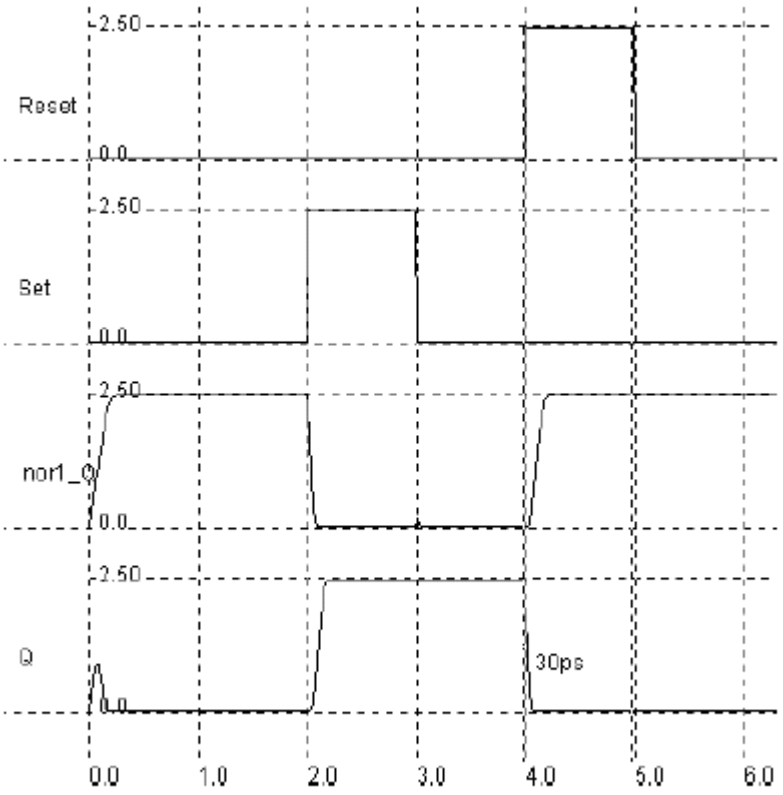
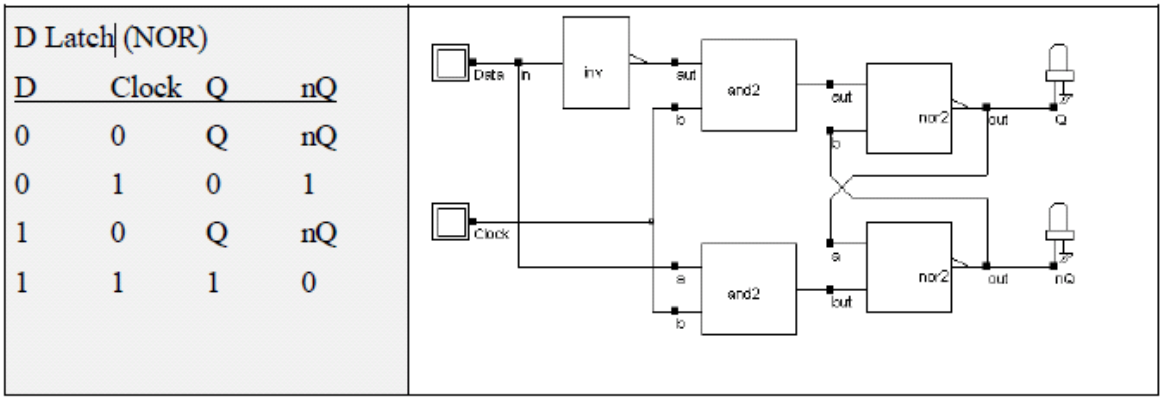


Fig. 7-3. Simulation of the RSNOR latch (RSNor.MSK)

7.2 D Latch

The truth table and schematic diagram of the static D latch, also called Static D-Flip-Flop, are shown in Figure 7-4. The data input D is transferred to the output if the clock input is at level 1. When the clock returns to level 0, the latch keeps its last value.

D Latch.



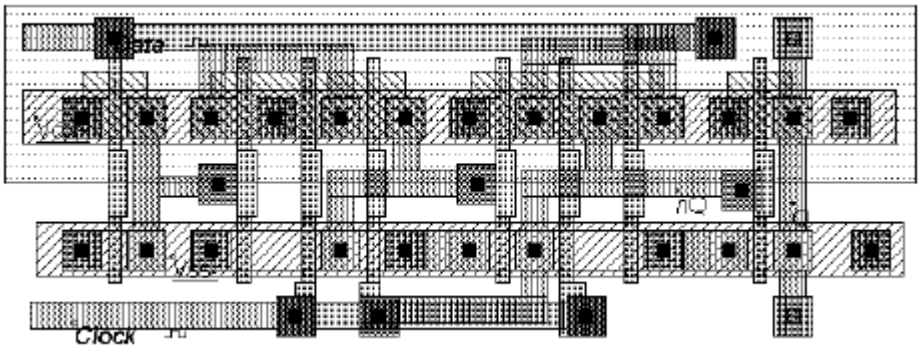
D Latch (NOR)

D	Clock	Q	nQ
0	0	Q	nQ
0	1	0	1
1	0	Q	nQ
1	1	1	0

Fig. 7-4. The truth table and schematic diagram of a D Latch (File DLATCH.SCH).

MANUAL DESIGN. Note that the NOR2-AND combination can be implemented in a complex-gate style. You may find useful to invoke the one line compiler to create successively one inverter **nd=~d**, and two complex gates which include the AND/NOR cells using the syntax **Q=~(nQ|(nd&h))** and **nQ=/(Q|(d&h)**. Build the interconnections and run the Design Rule Checker.

Assign a clock to CLK and a clock to DATA. An example of such an implementation can be found in the file "DLatchLevel.MSK". Its layout an corresponding simulation are illustrated in figure 7-5.



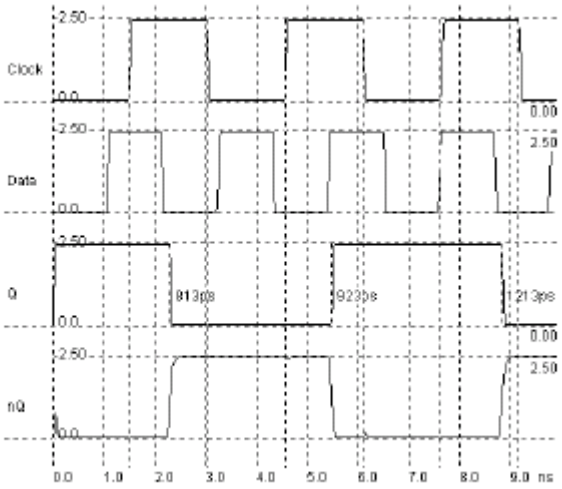


Fig. 7-5 Implementation and simulation of the D Latch (File DLatchLevel.MSK)

7.3 Edge Trigged Latch

The most common example of an edge-trigged flip flop is the JK latch. Anyhow, the JK is rarely used, a more simple version that features the same function with one single input D is preferred. This simple type of edge-trigged latch is one of the most widely used cells in microelectronics circuit design. The cell structure comprises two master-slave basic memory stages.

The most compact implementation of the edge-trigged latch is reported below. The schematic diagram is based on inverters and pass-transistors. On the left side, the two chained inverter are in memory state when the pMOS pass transistor P1 is on, that is when CLK=0. The twochained inverters on the right side act in an opposite way. The reset function is obtained by a direct ground connection of the master and slave memories, using nMOS devices.

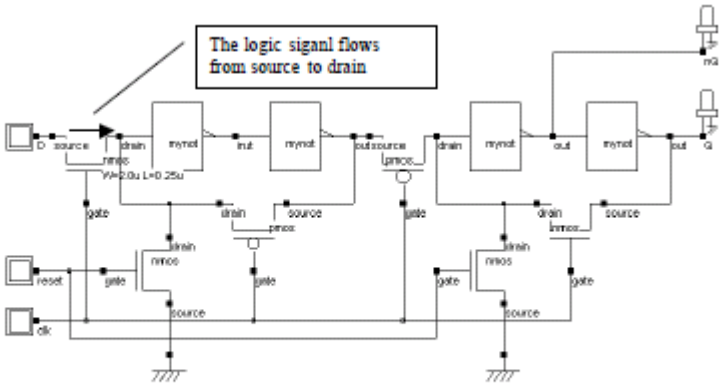


Figure 7-6 : The edge-trigged latch and its logic simulation (Dreg.MSK)

Notice that the logic model of the MOS device is not working the same way as for the real MOS switch. In the case of the logic implementation, the logic signal flows only from the source to the drain. This is not the case of the real switch where the signal can flow both ways.
Use the Verilog compiler to generate the edge-trigged latch, using the following text (dreg.txt), or by creating a schematic diagram including the "D" register symbol, in the symbol palette of DSCH2. As can be seen, the register is built up from one single call to the primitive "dreg". For simulation:
- RESET is active on a level 1. RESET is activated twice, at the beginning and later, using a piece-wise linear description included in the pulse property.
- CLK is a clock with 10ns at 0 and 10ns at 1.
- D is the data chosen here not synchronized with CLK, in order to observe various behaviors of the register.
To compile the DREG file, use the command "Compile"è "Compile Verilog Text". The corresponding layout is reported below. The piece-wise-linear data is transferred to the text "rst" automatically.

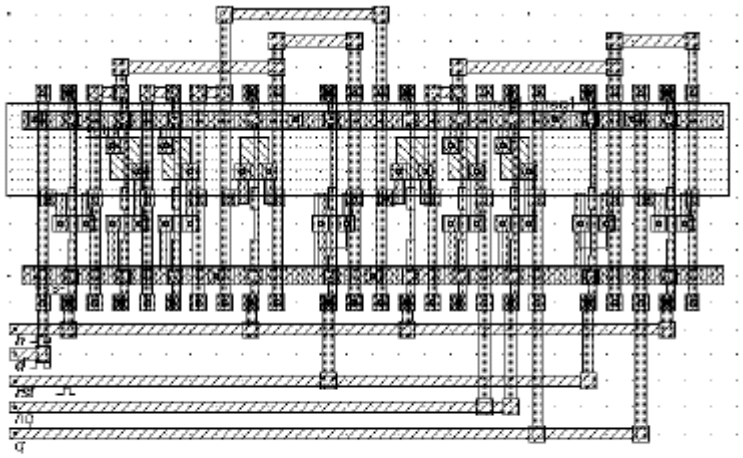


Fig. 7-7: Compiled version of the Edge-trigged D Flip Flop

The simulation of the edge-trigged latch is reported in figure 7-8. The signals Q and nQ always act in opposite. When RESET is asserted, the output Q is 0, nQ is 1. When RESET is not active, Q takes the value of D at a fall edge of the clock. For all other cases, Q and nQ remain in memory state. The latch is thus sensitive to the fall edge of the clock.

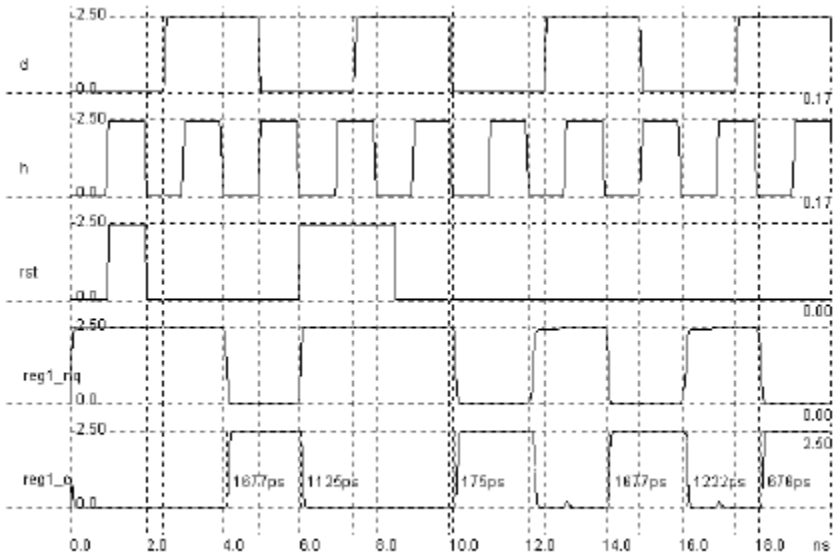


Fig. 7-8 Simulation of the DREG cell (DREG.MSK)

7.5 RAM Memory

The schematic diagram of the static memory cell used in High Capacity Static RAMs is given in figure 7-11. The circuit consists of 2 cross-coupled inverters and two nMOS pass transistors. The cell has been designed to be duplicated in X and Y in order to create a large array of cells. Usual sizes for Megabit SRAM memories are 256 x 256 cells or higher. An arrangement of 4x4 RAM cells is also shown in figure 6-14. The selection line Sel concerns all the cells of one row. The lines Data and nData concern all the cells of one column.

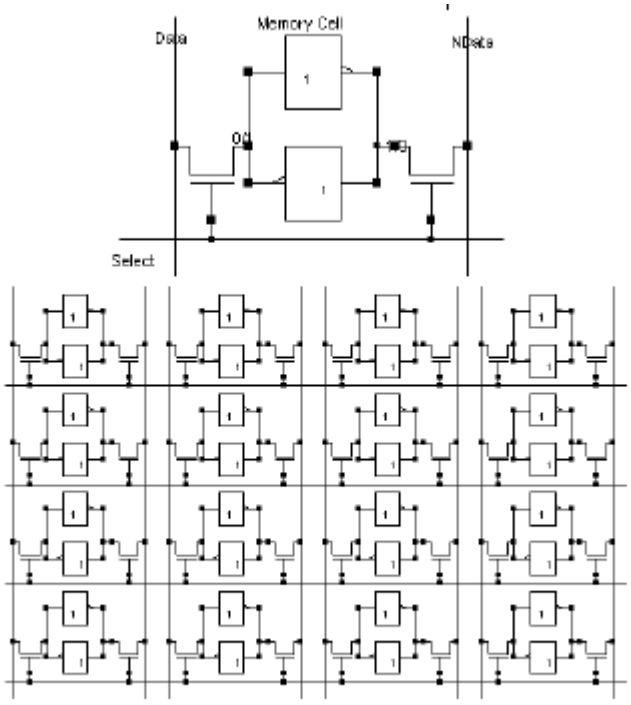


Fig. 7-11 The schematic diagram of the static RAM cell (RAM1.SCH).

The RAM layout is given in Figure 7-12. Click on File à Open à RAM.MSK to read it. The Data and nData signals are made with metal2 and cross the cell from top to bottom. The supply lines are horizontal, made with metal3. This allows easy matrix-style duplication of the RAM cell. The cross-section shows the nMOS devices and the connection to VSS using metal3, situated on the middle of the cell. The Data and nData lines, in metal2 are on both sides.

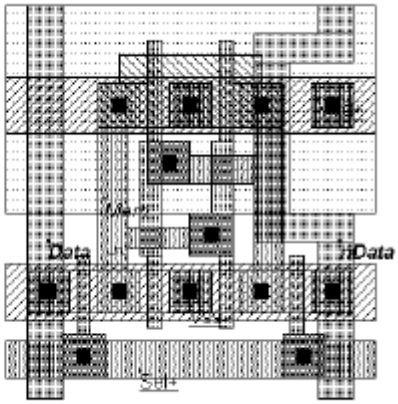


Fig 7-12. The layout of the static RAM cell (RAM1.MSK).

WRITE CYCLE. Values 1 or 0 must be placed on **Data**, and the data inverted value on **nData**. Then the line **Sel** goes to 1. The two-inverter latch takes the Data value. When the line **Sel** returns to 0, the RAM is in a memory state. See figure 7-13 for the analog simulation of the WRITE cycle.

READ CYCLE. In order to read the cell, the line **Sel** must be asserted. The RAM value propagates to **Data**, and its inverted value propagates to **nData**.

SIMULATION. The simulation parameters correspond to the write cycle in the RAM. The simulation steps describe din figure 6-16 are as follows:

1. **Mem** reaches 1, after an unstable period (unpredicatable value).
2. **Data** gets to value 0 and **nData** to value 1.
3. **Sel** is asserted. The memory cell **Mem** goes down to 0.
4. **Data** gets to a value of 1 and **nData** gets to a value of 0.
5. **Sel** is still asserted. The memory cell fights against Data=1 and surrenders (Mem=1).
6. **Sel** is inactive. The RAM is in a memory state.

7.6 RAM Array

You can duplicate the RAM cell into a 4x4 bit array using the command **Edit -> Duplicate XY**. Select the whole RAM cell and a new window appears. Enter the value « 4 » for X and « 4 » for Y into the menu. Click on « **Generate** ». The result is shown below.

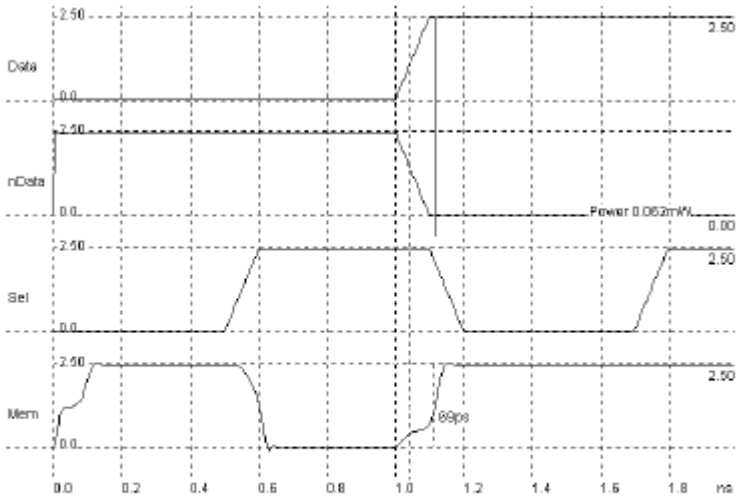


Fig. 7-13. Write cycle for the static RAM cell (RAM1.MSK).

7.6 RAM Array

You can duplicate the RAM cell into a 4x4 bit array using the command **Edit -> Duplicate XY**. Select the whole RAM cell and a new window appears. Enter the value « 4 » for X and « 4 » for Y into the menu. Click on « **Generate** ». The result is shown below.

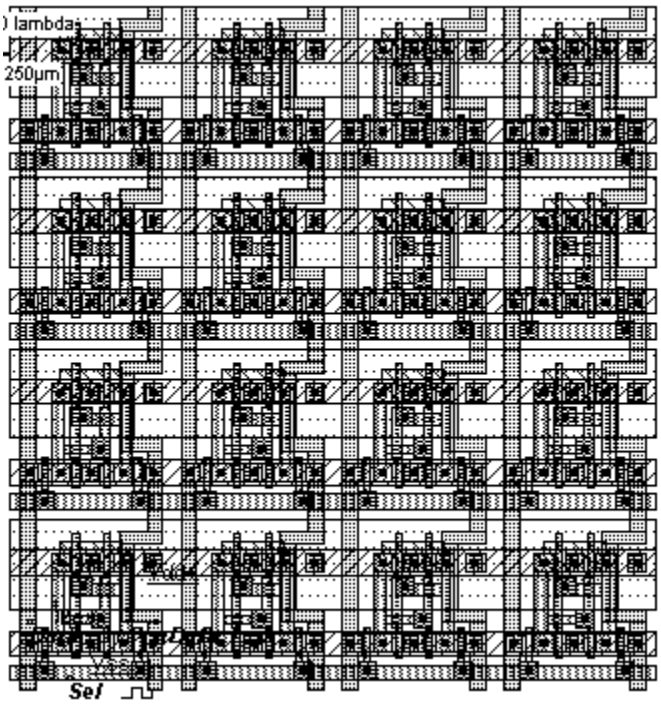


Fig. 7-14 Duplicating the RAM Cell in X and Y

7.7 RAM Line decoder

The line decoder is based on the following schematic diagram. One line is asserted while all the other lines are at zero. In this circuit one line was picked out from a choice of four lines. Using AND gates would be an easy solution, but in order to save the inverter, we choose NOR gates

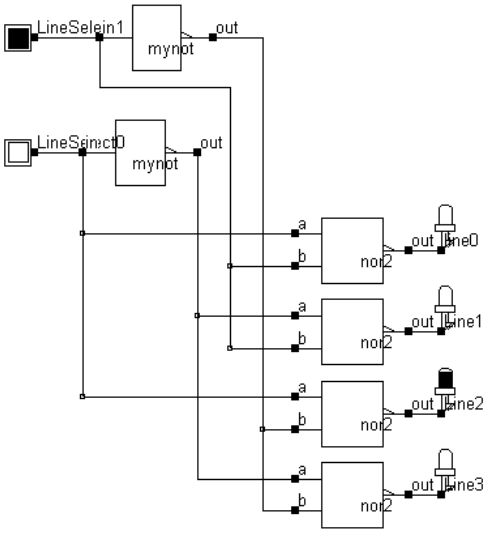


Fig. 7-15. A line selection circui

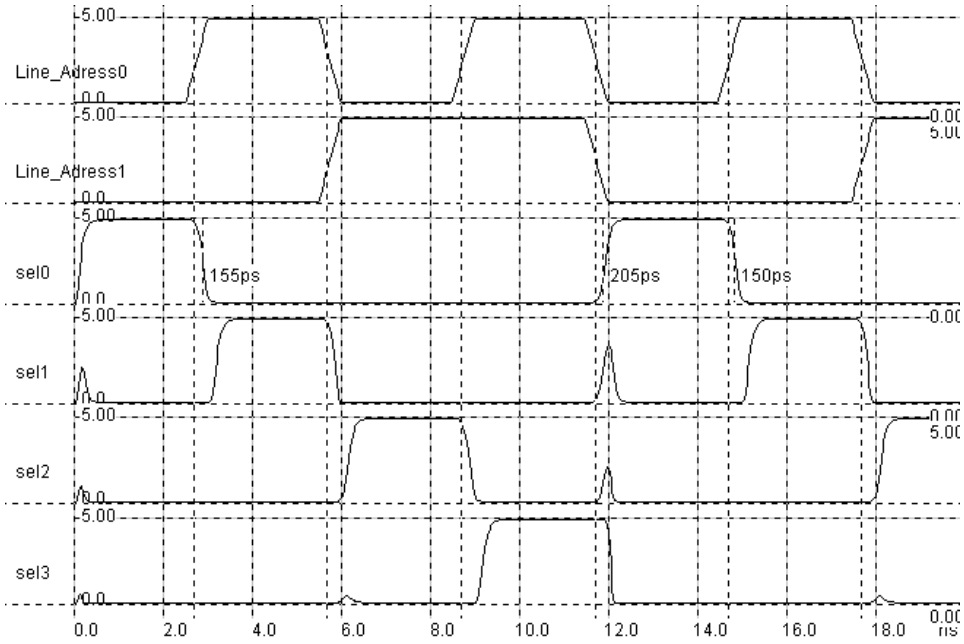


Fig. 7-16 A line selection layout and its corresponding simulation (RamLineSelect.MSK)

The NOR gate height should be adjusted to that of the RAM cell height. When making the final assembly between blocks, the command **Edit -> Move Area** is very important. This command helps to move a selected block with a lambda step.

7.8 RAM Column Selection

The column selection circuit is based on the same principles as those of the line decoder. The major modification is that the data flows both ways, that is firstly from the cell to the read circuit (Read cycle) and secondly from the write circuit to the cell (Write cycle). Fig. 7-17 proposes an architecture for this.

The n-channel MOS device is used as a switch controlled by the column selection. When the nchannel

MOS is on and **Write** is asserted, the data issued from DataIn is amplified by the

buffer, flows from the bottom to the top and reaches the memory. If **Write** is off, the 3-state inverter is in high impedance, which allows one to read the information.

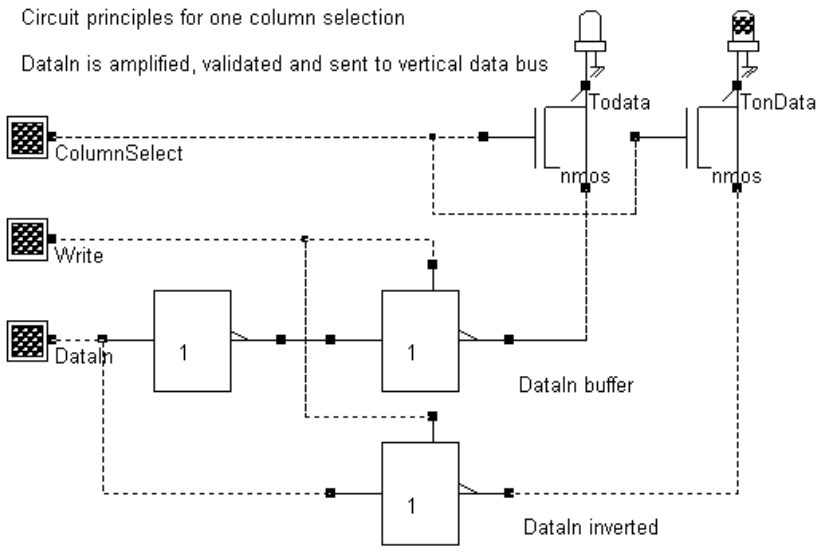


Fig. 7-17. Row selection and Read/Write circuit (RamColumn.SCH)

7.9 Dynamic RAM Memory

The dynamic RAM memory uses a single MOS device with a parasitic junction capacitance as a storage element. In figure 7-18, a set of 4x4 dynamic RAM cells are reported. The gates are connected horizontally while the drains are connected vertically. Furthermore, the sources are connected to a polarization node.

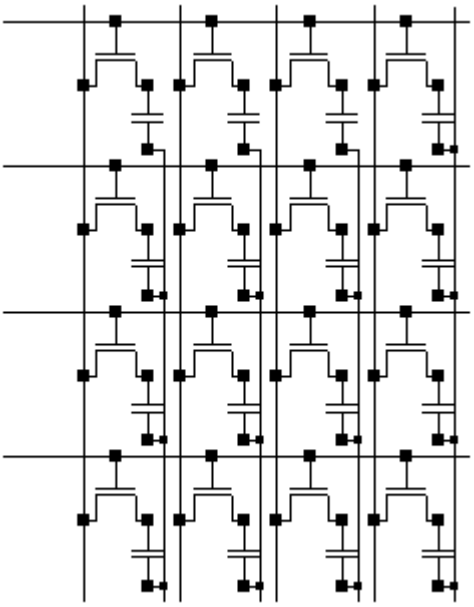


Fig. 7-18. An array of 4x4 dynamic cells (Dram4x4.SCH)

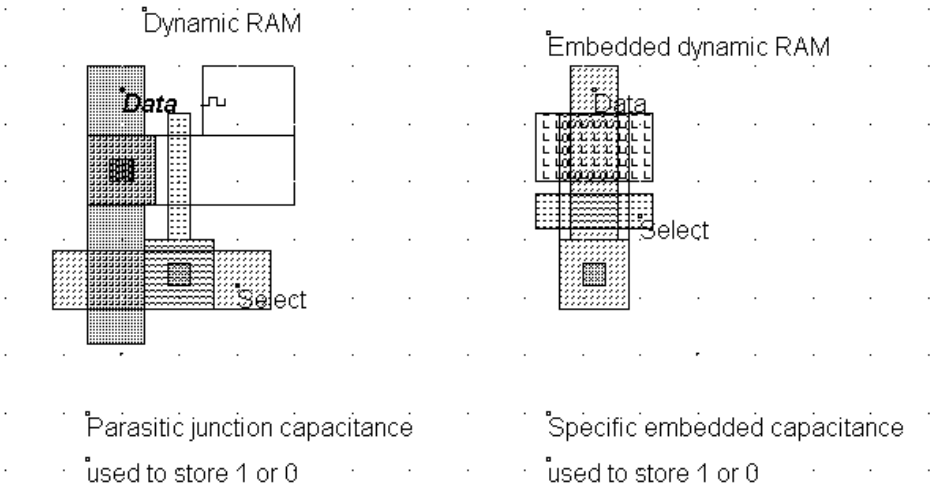


Fig. 7-19. The dynamic ram cell may be a single MOS device with enlarged source region or an embedded capacitance.

In figure 7-19, two layout implementation are proposed. In the left part, the source area of the MOS is significantly enlarged to increase artificially the parasitic junction capacitance. A typical target capacitance value Ccell for large DRAM arrays is 3fF. In this layout, Ccell is 0.2fF. In the layout of figure 7-19 right, a specific option layer is added, with a property of embedded capacitance with a very high value.

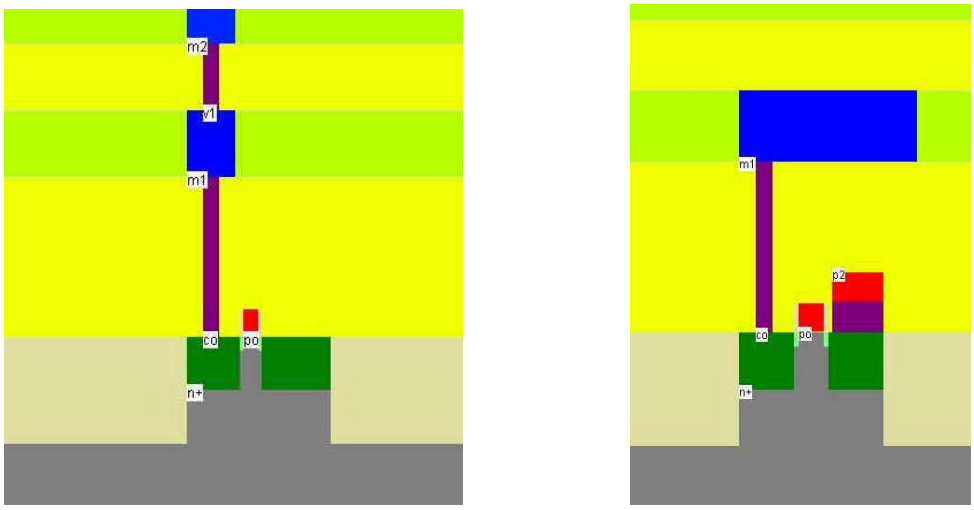


Fig. 7-20. Compared to the regular MOS (left), the embedded DRAM uses poly2 and a second specific layer (violet) to create an efficient capacitor.

Thanks to a specific option layer available in embedded RAM process, C_{cell} is increased to 3fF, while reducing the cell area (Figure 7-21 right). Three 8x8 DRAM arrays have been created by duplicating (Command Edit Duplicate XY in Microwind) the basic DRAM cell. See the impact of embedded capacitor on the reduction of silicon area. Also notice that joining two cells at the common contact saves silicon area. The fixed voltage of the embedded capacitor is driven by the 2_{nd} layer of polysilicon.

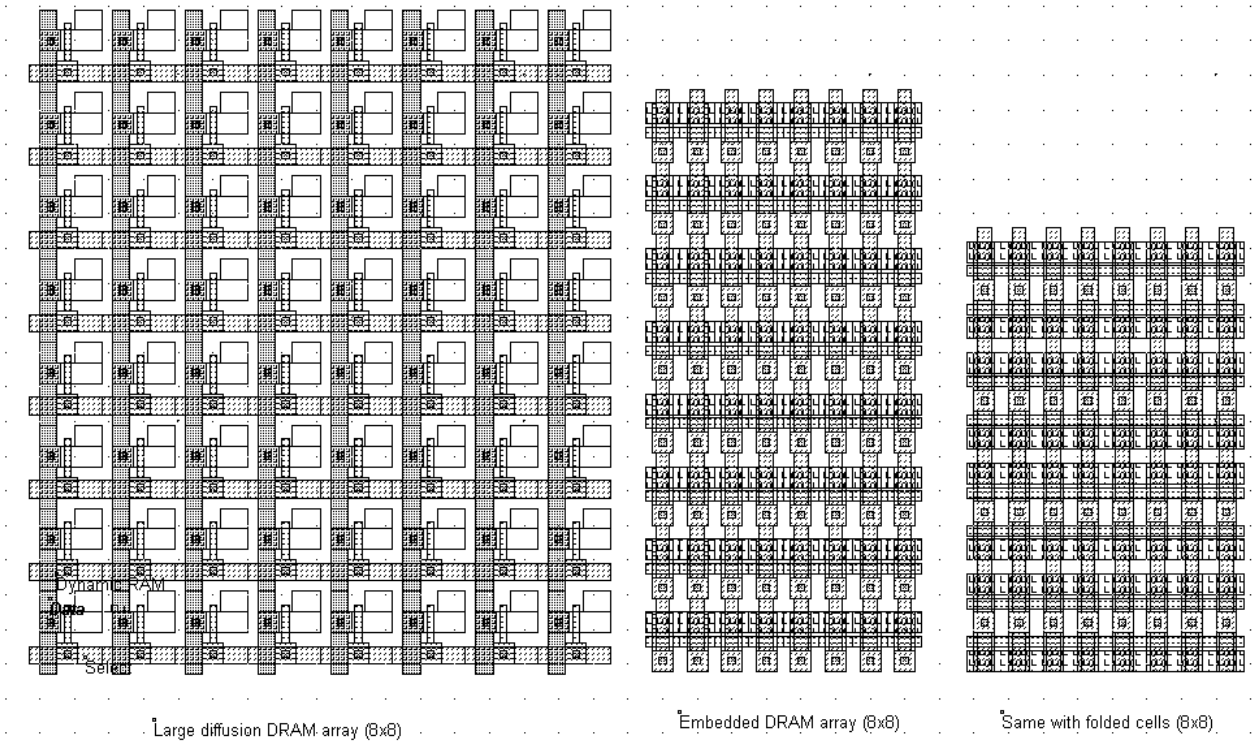


Fig. 7-21. Impact of embedded capacitor on silicon area on a 8x8 array

7.10 EEPROM

The EEPROM memory includes 2 poly gates, with the bottom polysilicon floating, isolated by oxide (Figure 7-22) . The programming of a double-poly transistor involves the transfer of electrons from the source to the floating gate through the thin oxide. The erasure involves the transfer of electrons from the floating gate back to the source.

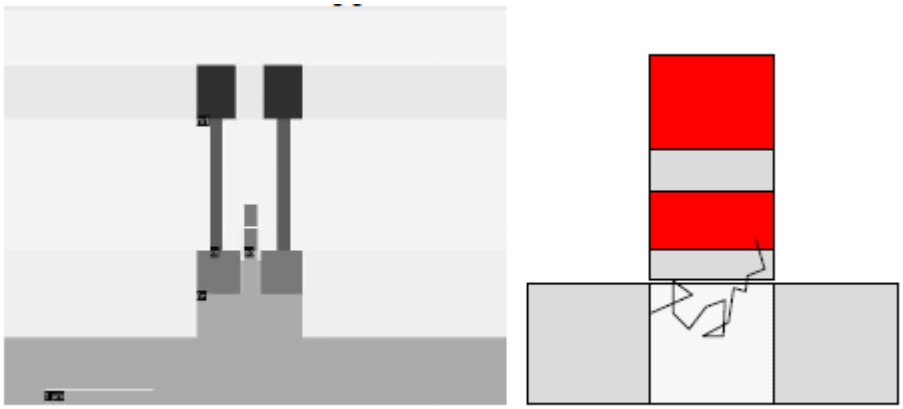


Fig. 7-22: The double-poly MOS device used as a non-volative memory

The programming operation is performed using a very high gate voltage on poly2, usually around 10V. The mechanism for electron transfer from the grounded source to the floating polysilicon gate is called tunneling. With a sufficiently positive voltage on the poly2 gate, the voltage difference between poly and source is high enough to enable electrons to pass through the thin oxide. The electron transfer mechanism is called hot electron injection.

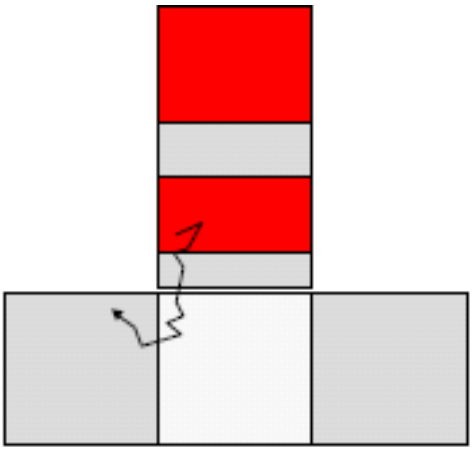
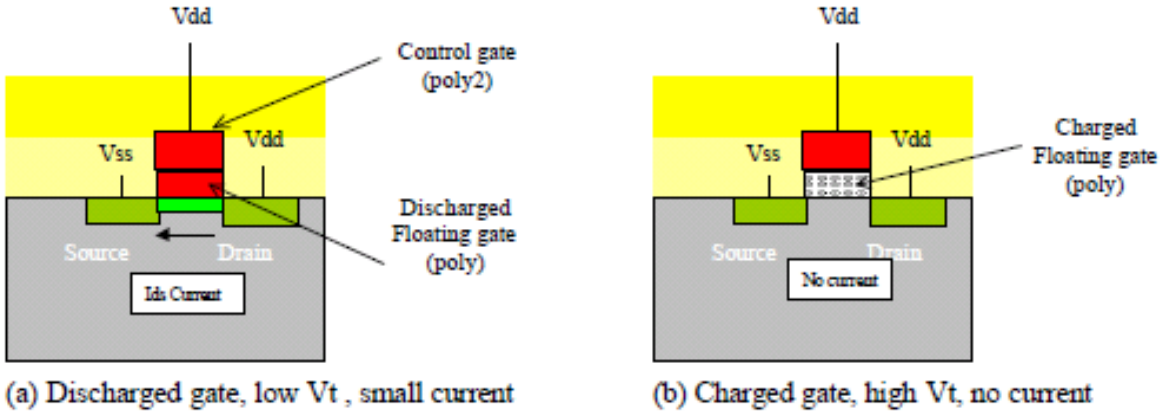


Fig. 7-23: Erase operation to remove electrons

For the erase operation (Figure 7-23), the poly2 gate is grounded and a high voltage (Around 10V) is applied to the source. Electrons are pulled off the floating gate thanks to the high electrical field between the source and the floating gate. This charge transfer is called Fowler-Nordheim electron tunneling.

From an operational point of view, the double-poly MOS device works as a normal MOS when the floating gate is discharged: a VDD gate voltage is high enough to turn the device on. Consequently, a Ids current flows between the drain and grounded source. If the floating gate is charged with electrons, the threshold voltage is very high, and a VDD gate voltage is not sufficient to turn the MOS on. Almost no Ids current is flowing.



(a) Discharged gate, low Vt , small current (b) Charged gate, high Vt, no current

Fig. 7-24: Read operation with a double-poly MOS device