

# Flash atmiņa

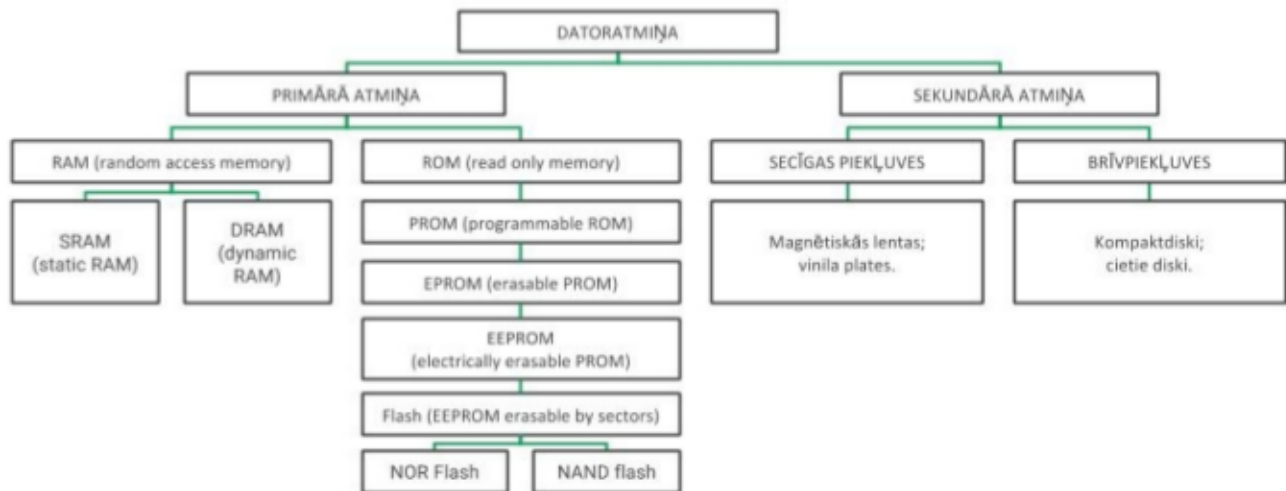
## Tās vieta atmiņu hierarhijā un tehniskās realizācijas vispārīgie principi

Mārtiņš Dundurs  
rect0 grupa  
apl. nr.

2017. gada 24. novembris

### 1 Digitālās atmiņas iedalījums

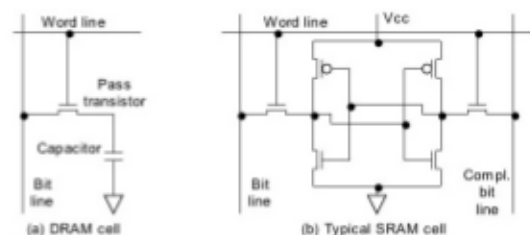
Pirms apskatām digitālās atmiņas iedalījumu jāpiemin, ka nosaukumi ir ieviesušies vēsturiski. Tāpēc nevienmēr tie raksturo zem tiem esošo ierīču galvenās īpašības un nozīmi.



Att. 1: Datoratmiņas iedalījums

Pirmkārt digitālo atmiņu var iedalīt *Primārajā* un *Sekundārajā*. Primāro atmiņu tieši izmanto mikroprocesors vai mikrokontrolieris, bet tas, kas saglabāts sekundārajā atmiņā, vispirms tiek pārkopēts uz primāro un lietots tikai tad. Sekundāro atmiņu sadala *Secīgas piekļuves* (SAM - sequential access memory) un *Brīvpiekļuves* (random access) atmiņā. Secīgas piekļuves atmiņā mēs nevaram uzreiz nolasīt kādu mums interesējošo informācijas bitu. Lai nolasītu kādu atmiņā pa vidu ierakstītu bitu, mums vispirms ir jānolasa visi biti pirms tā. Šāda atmiņa, piemēram, ir magnētiskās lentas vai gramofona ieraksts (vinilplates). Brīvpiekļuves atmiņā iespējams binārajai informācijai piekļūt jebkurā vēlamajā secībā. Tādas sekundārās atmiņas piemēri ir kompaktdiski, cietie diski. Šajā kategorijā var tikt iekļauta arī Flash atmiņa, ja tā tiek lietota kā sekundārā.

Primāro atmiņu savukārt iedala *Brīvpiekļuves atmiņā* (RAM - random access memory) un *Lasāmatmiņā* (ROM - read only memory). RAM galvenā īpašība šajā sadalījumā ir nevis brīvpiekļuve, bet gan tās energoatkarība (volatility - viegli gaistošs). Energoatkarība nozīmē to, ka informācija tiek zaudēta tiklīdz tiek atslēgta sistēmas barošana. Prestatā ir *Energoatkarīgā* atmiņa (NVM non-volatile memory). Pie tādas pieskaitāmas arī minētās sekundārās atmiņas, bet no primārajām energoatkarīga ir lasāmatmiņa. RAM savukārt iedalās *Statiskajā* (SRAM - static RAM) un *Dinamiskajā* brīvpiekļuves atmiņā (DRAM - dynamic RAM).

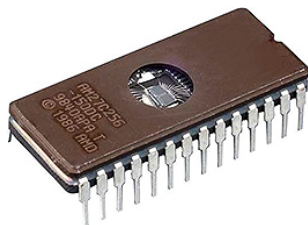


Att. 2: DRAM vs SRAM

SRAM atmiņas šūnas sastāv no triggeriem (flip-flops) un šādā atmiņā saglabātā informācija nezūd kamēr nav atslēgta barošana. Otra ir DRAM, kuras atmiņas šūnas sastāv no kondensatoriem. Informācija tiek saglabāta, balstoties uz to vai kondensators ir uzlādēts, vai nē. DRAM mīnus ir tas ka lādiņš kondensatorā ar laiku izplūst. Tādā veidā DRAM uzglabā informāciju tikai pāris milisekundes. Lai informāciju nezaudētu, DRAM tiek nepārtraukti nolasīts un pārrakstīts pa jaunam ik pēc noteikta īsa brīža - to sauc par atsvaidzināšanu (refreshing). Lai atsvaidzināšanu veiktu, ir nepieciešama sarežģītāka atmiņas matricu apkalpojošā shēma. Šo mīnusu daudzkārt atsvēr DRAM atmiņas šūnas vienkāršība un mazais izmērs. Tas nozīmē, ka to ir vieglāk un lētāk ražot, kā arī iespējams sasniegt daudzkārt lielāku bitu blīvumu kā SRAM. Rezultātā DRAM ir visplašāk sastopama patērētāju produktos.

Lasāmatmiņas jeb ROM nosaukums atkal ir maldinošs, jo rakstīšanas ierobežojums šī tipa atmiņā ir tikai tās vēsturiski pirmajam veidam - Mask ROM, kuras informācijas saturs tiek noteikts jau fotolitogrāfiskajā ražošanas procesā, kad ar noteiktā rakstā caurumotu masku tiek aizklāts apgaismojamais čips. Rezultātā tiek printētas tikai noteiktas atmiņas šūnas, veidojot attiecīgu nulļu 0 un vieninieku 1 rakstu ar attiecīgā tehnoloģiskā procesa šūnas sastāvu (diodi<sup>1</sup> vai FET).

ROM atmiņas ierakstīšana ražošanas procesā ir ļoti dārga un atmaksājas tikai lielos apjomos. Ja informācijā nepieciešami labojumi, tad ir jātaisa jauna atmiņas ierīce. Šī iemesla dēļ parādījās *Programmējamā* ROM (PROM - programmable ROM), kuru varēja ierakstīt, bet tikai vienreiz. Ražošanas procesā tika printētas diodes visās šūnās, bet lietotājam bija iespēja ar paaugstinātu spriegumu izsist noteiktus savienojumus, tādā veidā iegūstot sev nepieciešamo ierakstu.

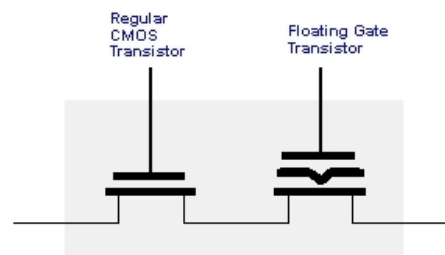


Att. 3: UVEPROM

Nākamais solis bija kaut kādā veidā panākt informācijas izdzēšanu un ierakstīšanu no jauna. Tas tika realizēts *Izdzēšamajā* PROM (EPROM - erasable PROM). Informācijas ierakstīšanai izmantoja jau PROM atmiņā pielietotos *Peldošā aizvara* lauktranzistorus (FG MOSFET - floating gate metal-oxide-semiconductor field effect transistor), kuros informācija tiek saglabāta uzkrāta lādiņa veidā peldošajā aizvarā. Peldošais aizvars ir dielektriski izolēts aizvars starp aizvaru un kanālu. Lai to programmētu, vajadzēja izmantot paaugstinātu spriegumu. Šo iemeslu dēļ ierīci nevarēja pārrakstīt turpat uz shēmas. Izdzēšanas process arī nebija tas vienkāršākais - čipu bija nepieciešams apgaismot ar ultravioleto gaismu (UV). UV gaismas ietekmē metāla oksīds - peldošo aizvaru izolējošais dielektriķis jonizējas, gūst pietiekamu vadītspēju un spēj izvadīt uz peldošā aizvara uzkrāto lādiņu prom. Praktiski atmiņas ierīci vajadzēja izņemt,  $\approx 30$  minūtes apgaismot ar UV gaismu un tad ievietot speciālā programmēšanas aparātā, lai ierakstītu vajadzīgo informāciju. Loģisks nākamais solis bija panākt, lai visu šo procesu varētu veikt uz shēmas.

Nākamais uzdevums tika īstenots t.s. elektriski izdzēšamajās un programmējamajās atmiņās (EEPROM jeb  $E^2$ PROM - electrically erasable PROM). Tas tika panākts izmantojot t.s. FLOTOX EEPROM šūnas, kurās papildus FG MOSFET tika izmantots arī atlasošais MOSFETs, kas atlasa, kuras šūnas tajā brīdī tiks izmantotas. Tas, protams, prasīja lielāku šūnu un rezultātā bija mazāks atmiņas blīvums.

Flash atmiņa principiāli ir tā pati EEPROM ar atšķirību to, ka tās izdzēšana/ierakstīšana norisinās pa sektoriem - tādā veselās šūnu kopnes uzreiz. Mēs apskatīsim Flash EEPROM atmiņas šūnas uzbūvi sīkāk atsevišķi. Tāpat mēs apskatīsim kāda ir Flash atmiņas arhitektūra. Arhitektūras ziņā ir diva veida Flash atmiņas NOR un NAND. Tās nosauktas attiecīgo loģiskās disjunktijas un konjunktijas operāciju vārdā dēļ šūnu savstarpējā izvietojuma.



Att. 4: FLOTOX EEPROM šūna

## 2 Peldošā aizvara lauktranzistora uzbūve

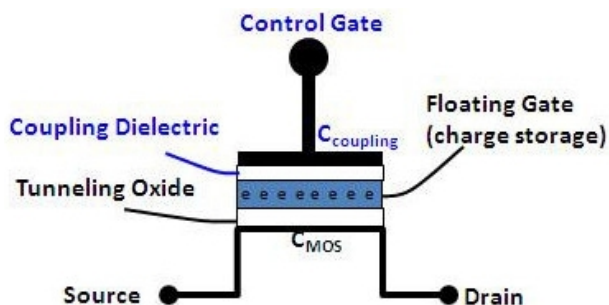
FG MOSFET tāpat ir MOSFETs ar peldošo aizvaru. Peldošais aizvars ir izolēts no visām pusēm ar dielektriķi un ir kapacitīvi saistīts ar aizvaru<sup>2</sup>. Ierīces struktūra redzama 5. attēlā. Šāda ierīce spēj saglabāt informāciju kā uzkrāto lādiņu peldošajā aizvarā. Tā kā peldošais aizvars ir izolēts no visām pusēm, tam nepieciešams pievadīt lielu spriegumu, lai elektroni spētu pārvarēt dielektriķa barjeru ( $\approx 12V$ ). Ja šādu spriegumu nepievada, tad peldošais aizvars spēj lādiņu saglabāt ļoti ilgu laiku. Piemēram,  $6nm$  oksīda dielektriķa biezums nodrošinās ka lādiņš saglabāsies vismaz 10 gadus<sup>3</sup>. Ražotāji var variēt ar šo oksīda slāņa biezumu, ja tik ilgs laiks nav jāgarantē vai arī lai samazinātu nepieciešamo spriegumu barjeras pārvarēšanai. Parasto aizvaru, kurš no peldošā aizvara

<sup>1</sup><https://www.google.ca/patents/US5441907>

<sup>2</sup>Kapacitīvā saite ļauj pārnēst enerģiju, izmantojot nobīdes strāvu

<sup>3</sup>Paulo Cappelletti; Carla Golla; Piero Olivo (30 June 1999). Flash Memories. Springer US. ISBN 978-0-7923-8487-8.

ir izolēts, sauc par *vadības aizvaru* (Control Gate). To dielektriķi, kurš izolē vadības aizvaru no peldošā, sauc par *Saites dielektriķi* (Coupling Dielectric). To dielektriķi, kurš izolē peldošo aizvaru no kanāla, sauc par *Tunelēšanas oksīdu* (Tunneling Oxide). Tunelēšanas tāpēc, ka elektroni, kuri to pārvar izmanto t.s. *Kvantu tunelēšanas* (Quantum Tunneling) parādību no kvantu mehānikas.



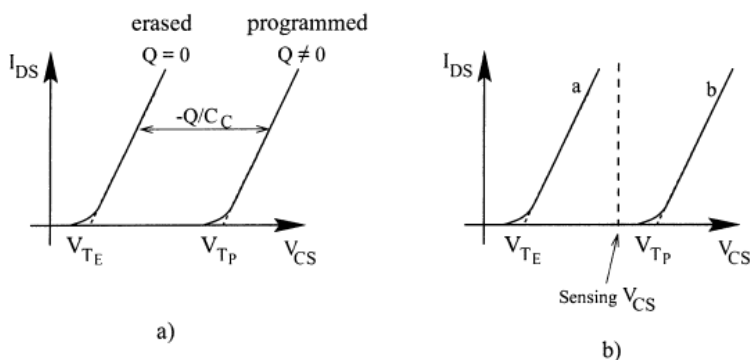
Att. 5: Peldošā aizvara MOSFETs

Šāda veida tranzistors principā veido individuālo atmiņas šūnu Flash atmiņā. Informācijas kodēšanu gan var veikt vairākos veidos. Vienkāršākais un drošākais kodēšanas veids ir ar diviem sprieguma līmeņiem - kā "0" saprot zemu sprieguma līmeni, proti, negatīvi uzlādētu peldošo aizvaru; kā "1" saprot augstu sprieguma līmeni, kas īstenībā ir elektriski neitrāls, proti, FG nav uzlādēts. Lai palielinātu atmiņas bitu blīvumu, tiek izmantota arī t.s. *vairāklīmeņu* kodēšana (MLC - multilevel coding). FG šādā kodēšanā var tikt uzlādēts ar vairākiem atšķirīgiem spriegumiem. Lai informāciju nolasītu, izmanto ADC. Tomēr MLC gadījumā ražotājiem ir jāreķinās ar paaugstinātu kļūdu daudzumu<sup>4</sup>.

Lai ievadītu iekš FG elektronus izmanto divas metodes. Viena paredz pievadīt augstu pozitīvu spriegumu pie vadības aizvara. Tādā veidā tiek tiek izvilkti elektroni no kanāla. Šeit notiek t.s. Fowler-Nordheim elektronu tunelēšanās no kanāla uz FG. Būtībā tiek samazināta enerģijas barjera, kas elektronam jāpārvar, lai nokļūtu iekš FG. Cits fenomens, kas tiek izmantots, lai ievadītu elektronus iekš FG ir t.s. karstā lādiņnesēju injekcija (CHE - channel hot electron injection). Arī šajā metodē elektroni tiek ņemti no kanāla, tomēr te tiek izmantota to kinētiskā enerģija. Vadot pietiekami lielu strāvu kanālā, elektroni gūst lielu kinētisko enerģiju, kas tiem ļauj pārvarēt dielektriķa barjeru. Tomēr, kanālā var nebūt pietiekams daudzums elektronu. Tāpēc tiek pievadīts pozitīvs spriegums vadības aizvarā. Tas palielina strāvas vadīšanai nepieciešamo elektronu daudzumu kanālā. Tas vajadzīgs arī, lai ievilkto elektronus no kanāla iekš FG.

Lai divu stāvokļu FG MOSFET atmiņas šūnā ierakstītu loģisko nulli "0", nepieciešams FG negatīvi uzlādēt. Kā tika iepriekš teikts, to var darīt divos veidos. Ražotāji izmanto abus, bet dažādos gadījumos. NOR arhitektūras Flash atmiņā FG uzlādēšanai izmanto CHE injekciju, bet NAND arhitektūrā - Fowler-Nordheim tunelēšanas.

Informācijas nolasīšana balstās uz tranzistor-effektu. Pie vadības aizvara tiek pievadīts noteikts spriegums  $V_T$ , kas ir mazāks kā dzēšanai vai rakstīšanai nepieciešamais. Jāatceras, ka jebkuram MOSFETam piemīt savs aizvara-izteces spriegums, kas izraisa strāvas plūšanu kanālā. Ja nepieciešamais spriegums netiek sasniegts, tad strāva notecē neplūst un to var uzskatīt par loģisko "0". Normāli, ja FG nebūs negatīvi uzlādēts, proti, uz tā nebūs elektronu, tad pie vadības aizvara pievadītais spriegums būs pietiekams, lai parādītos strāva notecē - šo situāciju uzskata par loģisko "1". Bet, ja iekš FG ir elektroni, tad tie kompensē daļu no vadības aizvara pozitīvā sprieguma un nepieciešamais vadības spriegums  $V_T$  no kanāla perspektīvas netiek sasniegts un strāva notecē neplūst.



Att. 6: Informācijas nolasīšana. Sensing norāda testēšanas spriegumu.

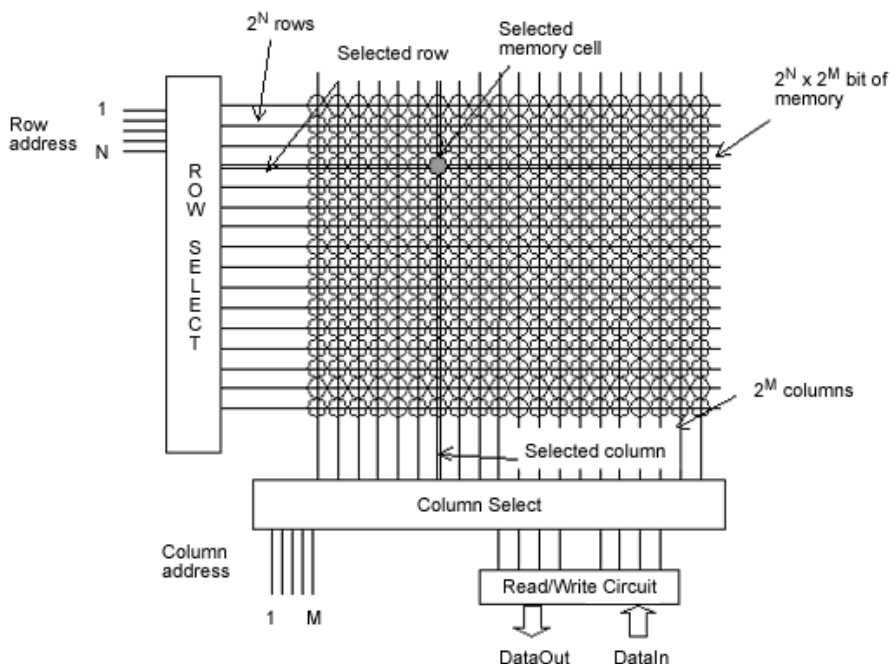
Informācijas dzēšana balstās uz elektronu izstumšanu no FG. To dara, vadības aizvaram pievadot augstu pozitīvu spriegumu, kas ir augstāks kā noteikta robeža. Robežu nosaka dielektriķa aizliegtās zonas platums. Palielinot pozitīvu spriegumu uz vadības aizvara, tiek samazināta tā enerģijas barjera, kas elektronam ir jāpārvar, lai ceļotu prom no FG uz izteci.

Flash atmiņā izdzēšanas process notiek pa sektoriem. Tas ir neliels solis atpakaļ no citām EEPROM atmiņām, kurās bija iespējams dzēst bitu-selektīvi, tomēr priekš Flash tas ļāva ievērojami samazināt šūnas izmērus. Vispār Flash atmiņa ir diezgan dabisks EEPROM atmiņas atzars, tāpēc atšķirības nav īpaši lielas. Atšķirības visvienkāršāk ir skaidrot nevis šūnas, bet gan arhitektūras līmeņi.

<sup>4</sup> Jānorāda, ka gala lietotājs ar šīm kļūdām nesaskarsies. Flash atmiņas darbojas speciāla loģika, kas katru reizi, kad tiek ierakstīta informācija, pārbauda vai tā ir ierakstīta pareizi. Ja ierakstīta nepareizi, tad tiek mēģināts rakstīt no jauna noteiktu skaitu reižu. Ja pēc tā arī neizdodas ierakstīt pareizi, tad tiek padots kļūdas signāls.

### 3 Flash arhitektūra

Flash atmiņas arhitektūrai ir divi nozīmīgi atzari NOR un NAND. Ir arī cita veida arhitektūras, bet tām ir ļoti specifiski pielietojumi. Tām Flash atmiņas ierīcēm, kuras mēs lietojam ikdienā, ir NAND arhitektūra. Sākotnēji NOR arhitektūra bija vairāk izplatīta, bet pēdējā desmitgadē NAND Flash atmiņa, pateicoties savai ietilpībai ir iekarojusi plaša patēriņa tirgu. Pirms aplūkosim Flash arhitektūru, atcerēsimies dažus atmiņas ierīču pamatprincipus. Atmiņas viselementārākais uzdevums ir saglabāt bitus. Katrā atmiņas šūnā var tikt saglabāts 1 bits. Ja mums ir, piemēram, virkne ar atmiņas šūnām, tad kā mēs varam nolasīt informāciju no kādas konkrētas šūnas  $n$ ? Lai tai piekļūtu, mums tai ir jāzin tās adrese. Šūnu adreses ir binārajā kodā. Mēs šūnai gribam pievadīt lasīšanas spriegumu pie tās vadības aizvara un tikai tai vienai šūnai. Šo uzdevumu var veikt *dekoders* - tā ir ierīce, kura saņem bināru kodu un, pamatojoties uz tā, aktivizē vienu no savām izejām. Šādā veidā mēs caur dekoderu nogādājam mūsu spriegumu pareizajai šūnai. Tagad ieslēdzot kaut vai visām šūnām spriegumu starp izteci un noteci, mēs varam vai nu saņemt signālu vai nesaņemt signālu.

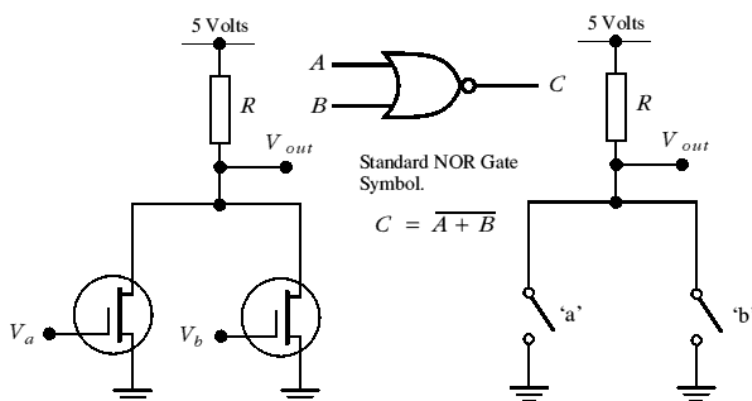


Att. 7: Atmiņas bitu nolasīšanas princips

vietā ir atkarīgs no arhitektūras. Flash atmiņās vārdu līnijas ir pievadītas vadības aizvaram. Bitu līnijas ir pievienotas notecei. Izteces līnija ir tā pati bitu līnija tikai no otra gala. Mēs izteces līnijai novērojam spriegumu. Ja spriegums ir virs noteiktas robežas, tas nozīmē ka strāva caru attiecīgo šūnu neplūst un nolasāma loģiskā "0".

#### 3.1 NOR Flash arhitektūra

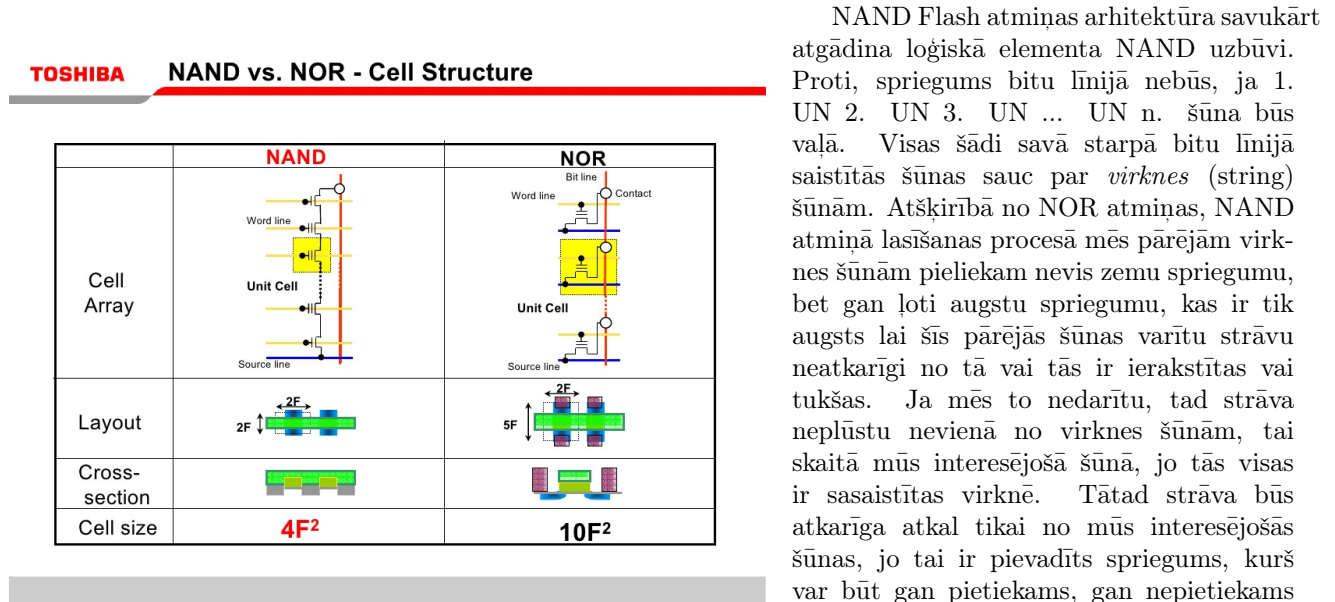
NOR Flash atmiņas arhitektūra īpaši neatšķiras no iepriekšējā apraksta. Par cik katrai šūnai ir sava adrese un sava piekļuve, tad tajā iespējams lasīt, rakstīt un arī dzēst informāciju brīvpiekļuves veidā tāpat kā RAM. NOR atmiņa ir arī ātrāka. Tomēr šūnas izmēri tai ir lielāki kā NAND. NOR tāpat kā NAND atmiņas nosaukums cēlies no tā, ka to šūnu izvietojums atgādina tranzistoru izvietojumu attiecīgi NOR un NAND loģiskajos elementos (skatīt Att. 8). NOR atmiņu lieto, kad nepieciešama ātra atmiņa. Tas ir kritiski atmiņas elementiem, kuri ir tuvu procesoram. Lasīšanas process NOR atmiņā paredz ka mēs aktivizējam



Att. 8: NOR loģiskā elementa uzbūve

mūs interesējošo šūnu ar augstu spriegumu pie šīs šūnas vadības aizvara, bet visām pārējām šūnām mēs pievadām zemu sprieguma līmeni. Tādā veidā mēs zinām ka, ja mēs izmērīsim kādu strāvu bitu līnijā, tad šī strāva noteikti plūdis caur mūs interesējošo šūnu. Tā mēs zinām ka šūna ir ierakstīta vai tukša.

### 3.2 NAND Flash arhitektūra



Att. 9: NOR un NAND Flash atmiņu arhitektūru salīdzinājums

mazāk vietas, pateicoties tam, ka katrai šūnai nav vajadzīgs atsevišķs izteces savienojums.

NAND šobrīd izmanto visplašāk. Tā kalpo kā masu glabāšanas ierīce par cik tai ir ļoti liels informācijas blīvums. Dēļ tās arhitektūras NAND atmiņā iespējamās vairāk kļūdas un arī tā noliektas ātrāk, par cik visas atmiņas šūnas tiek pārslēgtas katru reizi. Tomēr to lietošanas īpatnības atsvēr šos mīnus.

NAND Flash atmiņas arhitektūra savukārt atgādina loģiskā elementa NAND uzbūvi. Proti, spriegums bitu līnijā nebūs, ja 1. UN 2. UN 3. UN ... UN n. šūna būs vaļā. Visas šādi savā starpā bitu līnijā saistītās šūnas sauc par *virtnes* (string) šūnām. Atšķirībā no NOR atmiņas, NAND atmiņā lasīšanas procesā mēs pārējām virtnes šūnām pieliekam nevis zemu spriegumu, bet gan ļoti augstu spriegumu, kas ir tik augsts lai šīs pārējās šūnas varētu strāvu neatkarīgi no tā vai tās ir ierakstītas vai tukšas. Ja mēs to nedarītu, tad strāva neplūstu nevienā no virtnes šūnām, tai skaitā mūs interesējošā šūnā, jo tās visas ir sasaistītas virknē. Tātad strāva būs atkarīga atkal tikai no mūs interesējošās šūnas, jo tai ir pievadīts spriegums, kurš var būt gan pietiekams, gan nepietiekams strāvas vadīšanai atkarībā no tā vai FG ir attiecīgi tukšs vai arī uzlādēts. 9 attēlā ir parādīts ka NAND atmiņas šūna aizņem