http://www.electronics-tutorials.ws

ციფრული ლოგიკური ჩამკეტი **Tutorial: 1 of 10**

**შესავალი**

ციფრული ლოგიკური ჩამკეტი - ელექტრონული ხელსაწყოა, რომელიც ახორციელებს ლოგიკურ ფუნქციას მის შესავალ ელექტროდებზე არსებული სიგნალების კომბინაციებთან შესაბამისობაში. ამ ხელსაწყოს შესაძლებელია გააჩნდეს რამდენიმე შესავალი, მაგრამ მხოლოდ ერთი გამოსავალი. გაყიდვაში აესებული სტანდარტული ლოგიკური ჩამკეტები წარმოდგენილია ორი „ოჯახის “ , ან ფორმის სახით. პირველი ცნობილია **TTL**,ანუ ტრანზისტორ-ტრანზისტორული (*Transistor-Transistor*) ლოგიკის სახელით, მეორე **CMOS** (*Complementary Metal-Oxide-Silicon*) - დამატებითი მეტალ-ჟინგ-ნახევარგამტარის ლოგიკის სახელით. ეს ოჯახები წარმოდგენილია 7400 და 4000 მიკროსქემების სერიების სახით. ეს სერიები განსხვავებული ტექნოლოგიებითაა წარმოებული.

ციფრული ლოგიკური ჩამკეტი

ზოგადად, **TTL** ინტეგრალური მიკროსქემები (IC) ბიპოლარულ ტრანზისტორებზეა აწყობილი, ხოლო **CMOS** IC-ები,ველის ტრანზისტორების გამოყენებით. მარტივი ლოგიკური ჩამკეტები შესაძლებელია შეიქმნას დიოდების ტრანზისტორების და რეზისტორების გამოყენებით - ე.წ. რეზისტორ-ტრანზისტორული ლოგიკა (**RTL**, Resistor-Transistor logic gates), დიოდ-ტრანზისტორული ლოგიკა (**DTL**, Diode-Transistor logic gates), ემიტერით დაკავშირებული ლოგიკა (**ECL**, Emitter-Coupled logic gates).

ინტეგრალური მიკროსქემები შესაძლებელია დავაჯგუფოდ მათში გამოყენებულუ ტრანზისტორებიდ რაოდენობის მიხედვით. მაგალითად, მარტივი და (AND) ჩიპი შეიძლება შეიცავდეს რამდენიმე ტრანზისტორს, ხოლო რთული მიკროპროცესორი რამდენულე ათას ტრანზისტორულ ჩამკეტს.

**ინტეგრალური სქემების კლასიფიკაცია**

* მცირე ინტეგრაციის (SSI) - შეიცავს 10-მდე ტრანზისტორს ან ჩამკეტს ერთკორპუსში. ესენი შეიძლება იყოს ontain და (AND), ან (OR), არა(NOT) ჩამკეტები.
* საშუალო ინტეგრაციის (MSI) – 10 დან 100 ტრანზისტორამდე, ან რამდენიმე ათეულ ჩამკეტს ერთ კორპუსში ციფრული ოპერაციების შესასრულებლად (დეცოდირება (decoders), თვლა (counters), ტრიგერი (flip-flops) და მულტიპლექსორი (multiplexers).
* დიდი ინტეგრაციის (LSI) – 100 დან 1,000 ტრანზისტორამდე, ან რამდენიმე ასეული ჩამკეტი სპეციფიური ციფრული ოპერაციებისათვის (ინფორმაციის შეყვანა/გამოყვანა (I/O), დამახსოვრება, არითმეტიკულ-ლოგიკური ოპერაციები ).
* ძალიან დიდი ინტეგრაციის (VLSI) - 1,000 დან 10,000 ტრანზისტორამდე, ან რამდენიმე ათასი ჩამკეტი გამოთვლითი ოპერაციების შესასრულებლად ( პროცესორები, დიდი მოცულობის მეხსიერება, პროგრამირებადი ლოგიკური მოწყობილობები).
* სუპერ დიდი ინტეგრაციის (SLSI) - 10,000 დან 100,000 ტრანზუსტორამდე გამოთვლითი ოპერაციების შესასრულებლად (მიკროპროცესორები, მიკროკონტროლერები, კალკულატორები.
* ულტრა ინტეგრაციის (ULSI) – 1 მილიონზე მეტი ტრანზისტორი ( კომპიუტერების პროცესორები და ვიდეოპროცესორები, პროგრამირებადი მეხსიერება).

უფრო მაღალი დონის ინტეგრაციის ჩიპები ცნობილის სისტემური ჩიპების (**System-on-Chip** or (**SOC**)) სახელით. ამ შემთხვევაში ერთ სისტემაში მოქცეულია მრავალი სხვადასხვა დანიშნულების კვანძები, რომლების ქმნიან ერთიან ელექტრონულ სისტემას. ასეთი ჩიპებიო გამოიყენება მობილურ ტელეფონებში, ვიდეოკამერებში, მიკროკონტროლერებში. მათში 100 მილიონამდე ტრანზისტორია.

**მურის კანონი**

1965 წელს ცორპორაცია ინტელის თანადამფუძნებელმა გორდონ მურმა (Gordon Moore)გამოთქვა მოსაზრება, რომ ჩიპებში ტრანზისტორების ინტეგრაციის ხარისხი ორჯერ იზრდება ყოველ 18 თვეს. სეთი ტემპით ვითარდება ჩამკეტების წარმოების ტექნოლოგია. ყველაფერი დაიწყო 60 ტრანზისტორით ერთ ჩიპში. ამჟამად რიცხვი 2 მილიარდამდეა ასული (Quad-core **Itanium** 64-bit microprocessor chip) და კვლავ იზრდება.

**ციფრული ლოგიკის მდგომარეობები**

ციფრული ლოგიკური ჩამკეტი წარმოადგენს ციფრული ელექტრონიკის და მიკროპროცესორების ძირითად კონსტრუქციულ ელემენტს. ჩამკეტები აწარმოებენ ლოგიკურ ოპერაციებს AND, OR და NOT ორობით რიცხვებზე. ამ ელემენტების მუსაობის დროს დაშვებულია მხოლოდ ორი ლოგიკური მდგომარეობა ლოგიკური „1“ ( Logic "1") და ლოგიკური „0“ (Logic "0"), მაღალი (High) და დაბალი (Low), ჭეშმარიტი (True) და მცდარი (False). ბულის ალგებრაში ( [**Boolean Algebra**](http://www.electronics-tutorials.ws/boolean/bool_1.html)) და ჭეშმარიტების ცხრილებში (**Truth Tables)** ეს მნიშვნელობები წარმოდგენილია რიცხვებით „1“ და „0“. უმარტივესი მაგალითია ჩართული ან გამორთული ნათურა. მისი მდგომარეობა შეიძლება ნიშნავდეს „1“ ან „0“ და არა ორივეს ერთდროულად.

ციფრული სისტემების უმრავლესობა იყენება ე.წ. „დადებით ლოგიკას“ ("Positive logic), რომელშიც "0" ანr "LOW" წარმოდგენილია ნულოვანი ელექტრული ძაბვით, 0v ან დამიწებით. ლოგიკური დონე „1“, ან "HIGH" წარმოდგენილია უფრო მაღალი ძაბვით, მაგალითად +5 . ერთი დონიდან მეორეზე გადასვლა ხდება რაც შეძლება სწრაფად იმისათვის რომ გამოირიცხოს რაიმე სხვა გაურკვეველი შინაარსის მდგომარეობა. არსებობს ე.წ. უარყოფითი ლოგიკაც, სადაც ძაბვის ეს მნიშვნელობები შემობრუნებულია. ჩვენ განვიხილავთ მხოლოდ დადებით ლოგიკას, რომელიც უფრო ფართოდაა გავრცელებული.

სტანდარტულ TTL ლოგიკაში მკაცრად განმარტებულია ნოლისა და ერთიანის ძაბვების მნიშვნელობების ინტერვალები..

**TTL შესავალი და გამოსავალი დონეების ძაბვები.**

|  |
| --- |
| TTL Logic Levels |

7400 და CMOS 4000 ლოგიკა შეიცავს მრავალ ფუნქციონალურ ჩიპს. TTL ჯგუფი, მაგალითად 74Lxx, 74LSxx, 74ALSxx, 74HCxx, 74HCTxx, 74ACTxx, +5 v კვების შემთხვევაში აღიქვამს როგორც 1 ძაბვებს ინტერვალში 2.0v and 5v, ხოლო 0 ინტერვალში 0.8 ვოლტზე დაბლა. მნიშვნელობები ამ ორ ინტერვალს შორის ითვლება განუმარტავად. ამ ინტერვალში მუშაობა მცდარ შედეგებს იძლევა. CMOS 4000 ლოგიკისათვის სათანადო ინტერვალებია 3.0 and 18 v, და 1.5 ვოლტზე დაბლა.

**იდეალური ლოგიკური დონეების ზაბვები**

ეს ძაბვებია 0 და +5 ვოლტი

|  |
| --- |
| Ideal Digital Logic Voltage Levels |

ნახატზე ნაჩვენებია უმარტივესი ლოგიკური ჩამკეტი, რომლის დონეები იდეალურია. რეზისტორს ეწოდება „ამწევი რეზისტორი“ ( "pull-up" resistor).

**ციფრული ლოგიკის უმარტივესი ჩამკეტები.**

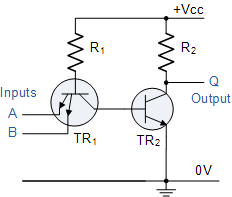
უმარტივესი ლოგიკური ჩამკეტები შესაძლებელია შეიქმნას დიოდების, რეზისტორების და ტრანზისტორების გამოყენებით. სქემები „და“ ((DRL) AND), და „და არა“ ( (DTL) ნაჩვენებია ნახატებზე.

|  |  |
| --- | --- |
| Diode-Resistor circuit  დიოდ-რეზისტორული | Diode-Transistor circuit  დიოდ-ტრანზისტორული |
| Diode Resistor Logic  2-input AND gate | Diode Transistor Logic  2-input NAND gate |

ტრანზისტორი უარყოფის ფუნქციას ასრულებს. ამიტომ მისი დამატებით „დას“ უარყოფა ხორციელდება. პრაქტიკაში, ინტეგრალურ სქემებში ასეთი სქემები არ გამოიყენება - ისინი ნელა მუშაობენ და ენერგიის დანაკარგებით ხასიათდებიან. ტრანზისტორი იზსნება 0.6 ვოლტიდან.

**საბაზო TTL ლოგიკური ჩამკეტები**

ზემოდ ნაჩვენებ სქემაში დიოდები შესაძლებელია შეიცვალის ტრანზისტორით, რომელსაც 2 ემიტერი გააჩნია. როდესაც მეორე ტრანზისტორის ბაზაზე დენი ნოლის ტოლია, გამოსავალზე 1-ია. ამის უზრუნველსაყოფად ორივე შესავალზე 0 უნდა იყოს. თუ ერთერთ შესავალზე ერთია - ტრანზისტორი გახსნილია და გამოსავალზე ნოლია.



**2-input NAND gate**

**ინტეგრალური სქემების "74" ქვეჯგუფი**

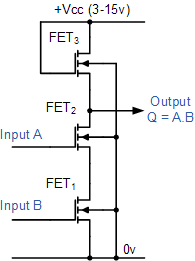
ეს TTL ოჯახი ძალიან მრავალფეროვანია თავისი სიმძლავრის, დროითი მახასიათებლების და შესავლის დატვირთვის შესაძლებლობლების მიხედვით. სახელწოდებაში 74-ს შემდეგ ასოებით აღინიშნება ტექნოლოგია, რომელიც განსაზღვრავს პარამეტრების კლასს. შემდები რიცხვი აღნიშნავს საკუთრივ მიკროსქემის სახელს. ამ სახელით შესაძლებელია მისი ფუნქციის იდენტიფიკაცია ცნობარის ან მონაცემთა ბაზის გამოყენებით.

* 74xx or 74Nxx: Standard TTL - ამ ჩამკეტებით დაიწყო TTL სერია 70-იანი წლების დასაწყისში. მათი სიგნალის ცვლილების დრო უდრის 10ns, ხოლო სიმძლევრე 10Mw. მიაქვიეთ ყურადღება ასოებს ჩიპის მარკირებაში.
* 74Lxx: Low Power TTL - მოხმარებული სიმძლავრე უფრო დაბალია, მაგრამ სიჩქარეც დაბალი გახდა.
* 74Hxx: High Speed TTL - ამ სერიაში მომატებულის სწრაფწმწდება, მაგრამ გაზრდილია მახმარებული სიმძლავრე..
* 74Sxx: Schottky TTL - სხვა, წინა სერიებთან შედარებით, შოტკის ტექნოლოგიის გამოყენებამ გააუმჯობესა შესავალი იმპედანსი, სწრაფქმედება და მოხმარებული სიმძლავრე.
* 74LSxx: Low Power Schottky TTL - იგივეა რაც 74Sxx, მაგრამ მკვეთრად დაწეულია მოხმარებული სიმძლავრე.
* 74ASxx: Advanced Schottky TTL - ეს სწრაფქმედების თვალსაზრისით გაუმჯობესებული 74Sxx Schottky სერიაა. გაუმჯობესება მოხდა მოხმარებული სინძლავრის გაზრდის ხარჯზე - 22mW.
* 74ALSxx: Advanced Low Power Schottky TTL - მოხმარება 1mW და სწრაფქმედება 4nS.
* 74HCxx: High Speed CMOS - CMOS ტექნოლოგიის გამოყენებით მოხმარება დაყვანილია 1uA (1 მიკროამპერი). სერია თავსებადია CMOS სერიაბთან.
* 74HCTxx: High Speed CMOS - CMOS ტექნოლოგია და ტრანზისტორები. 1uA-ზე ნაკლები დენი. შედარებით დაბალი - 16nS სწრაფქმედება TTL სერიებთან შესათანხმებლად.

**საბაზო CMOS ციფრული ლოგიკური ჩამკეტები**

TTL ლოგიკის ერთერთი ნაკლია ენერგიის მაღალი მოხმარება. ტრანზისტორებში დენის გადართვით ფუნქციონირებენ. ტრანზისტორებს აგრეთვე გააჩნია შემოსაზღვრული სწრაფქმედება. ამ სირთულეების გადასალახავად იყენებენ "CMOS" ლოგიკურ ჩამკეტებს, რომელთა ტექნოლოგია ეყრდნობა ე.წ. „ველის ეფექტის ტრანზისტორებს“ ("Field Effect Transistors", FET), ან „ველის ტრანზისტორებს“.

ეს ჩამკეტები შესავალზე იყენებენ ორივე P-არხისl და N-არხის ველის ტრანზისტორებს. ელექტრული ველით მართვის გამო, დენის მოხმარება თითქმის ნულის ტოლია (1 - 2uA). ასეთი მოხმარება იდეალურია გადამტან ელექტრონიკაში გამოსაყენებლატთ. ჩართვა-გამორთვის სიჩქარე 100MHz აღწევს, რაც აგრეთვე ძალიან ხელსაყრელია სწრაფქმედ კომპიუტერულ სქებებში გამოსაყენებლათ.



**არა-და ჩამკეტი ორი სესავლით (2-input NAND gate)**

მოყვანილი CMOS ჩამკეტის მაგალითი შეიცავს სამ 3 N-არხის MOSFET ტრანზისტორს. თითო ტრანზისტორი FET1 და FET2 შესავალზე, და ერთი FET3 გამოსავალზე. როდესაც A და B შესავალზე არის ლოგიკური "0", FET1 და FET2 გამორთულია ("OFF"). ეს იძლევა ლოგიკურ "1" გამოსავალზე FET3. როდესაც ორივე შესავალზე არის ლოგიკური "1", დენი გადის ამ ტრანზისტორებში და Q გამოსავალზე გვაქვს ლოგიკური "0", ანუ სრულდება „არა-და“ (NAND) ფუნქცია.

TTL ლოგიკისაგან განსხვავებით, ყველა გაუმჯობესება, ენერგიის მოხმარების და სწრაფქმედების თვალსაზრისით, რეალიზებულია CMOS 4000 "CD" სერიაში. გამეორებულია TTL სერიის ყველა ლოგიკური ფუნქცია.

დამზადების ტექნოლოგიის თვალსაზრიშით, CMOS ტექნოლოგია უგრო მოხერხებულია: ორივე გამტარებკობის N-არხის და P-არხის ტრანზისტორი ერთ და იგივე ტიპის ნახევარგამტარ მასალას იყენება. ამ ლოგიკის ერთადერთი ნაკლია მგრძნობიარობა სტსტიკური ელექტრობის მიმართ. სტატიკური ელექტრონა აზიანებს ველის ტრანზისტორებს. TTL ლოგიკა მუშაობს ერთ ფიქსირებულ ძაბვაზე, ხოლო CMOS ლოგიკა +3 დან +18 ვოლტამდე დიაპაზონში.

ლოგიკური „და“ ჩამკეტი **Tutorial: 2 of 10**

**განმარტება**

ლოგიკური ჩამკეტი **„და“** ჩვეულებრივად გამოსავალზე "0" მდგომარეობით ხასიათდება, და იღებს "HIGH" ან ლოგიკურ "1" მდგომარეობას, როდესაც ყველა მის შესავალზე არის ლოგიკური "1". მისი მდგომარეობა ისევ უბრუნდება "0" დონეს, თუ ერთ შესავალზე მაინც გაჩნდა "0". ბულის ალგებრაში ეს ელემენტი „და“ (AND) ასრულებს ლოგიკური გამრავლების ფუნქციას, რომელიც აღინიშნება წერტილის სიმბოლოთი (.): ჩაიწერება: A.B = Q.

თუ ჩამკეტს ორი შესავალი გააჩნია, შეგვიძლია განვმარტოდ ეს ჩამკეტი:

**"თუ ორივე A და B შესავალი ჭეშმარიტია, გამოსავალი Q ჭეშმარიტია"**

**ტრანზისტორული „და“ ჩამკეტი ორი შესავლით**

ნახატზე ნაშვებებია უმარტივესი რეზისტორ ტრანზისტოტული (RTL) ჩამკეტი. რეზისტორები უშუალოდ ბაზასთან არიან შეერთებულები. გამოსავალი Q მხოლოდ მაშინ მიირებს ჭეშმარიტ მნიშვნელობას, როდესაც ორივე ტრანზისტორია ჩართული.

|  |
| --- |
| 2-input Transistor AND Gate |

ასეთი ლოგიკური ჩამკეტები რეალიზებულია ელექტრონულ ინტეგრალურ სქემებში, სქემებზე აღინიშნება სათანადო სიმბოლოთი

**ციფრული ლოგიკური ჩამკეტი „და“**

**„და“ ჩამკეტი 2 შესავლით**

|  |  |  |  |
| --- | --- | --- | --- |
| სიმბოლო, Symbol | ჭეშმარიტების სხრილი, Truth Table | | |
| 2-input AND gate  **2-input AND Gate** | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |
| Boolean Expression **Q = A.B** | Read as A **AND** B gives Q | | |

**„და“ ჩამკეტი 3 შესავლით**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | Truth Table | | | |
| 3-input AND gate  **3-input AND Gate** | **C** | **B** | **A** | **Q** |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **1** |
| Boolean Expression **Q = A.B.C** | Read as A **AND** B **AND** C gives Q | | | |

„და“ ოპერაცია განიმარტა როგორც ბულის ალგებრის გამრავლება (.) ორი შესავლისათვის. შეგვიძლია გავზარდოთ შესავლების რაოდენობა ამ მარტივი ჩამკეტების კასკადური შეერთებით. წარმოება ამზადებს ჩიპებს 2, 3 და 4 სესასვლელით, მაგრამ არ წარმოადგენს სირთულეს ნებისმიერი რაოდენობის შესავლების მქონე ჩამკეტის მომზადება. ნახატზე მოყვანილია ასეთი მაგალითი:

|  |
| --- |
| 6-input AND Gate |

ბულის გამოსახულებქა ამ 6 შესავლის მქონე „და“ ჩამკეტისათვის იქნება: **Q = (A.B).(C.D).(E.F)**

თუ საჭიროა კენტი რაოდენობის შესავალი, შეგვიძლია ნებისმიერი ორი ერთმანეთთან შევაერთოთ. შესაძლებელია რომელიმე წყვილის ერთი „გამოუყენებელი“ შესავალი შევაერთოდ ჩიპის კვებასთან. ამით ამ ელექტროდზე დაფიქსირდება „1“. ამ ჩართვისათვის იყენებენ დენის დამწევ რეზისტორს ("Pull-up" resistors).

ელექტრონიკაში გამოიყენება ცხრილში მოყვანილი „და“ ჩამკეტები:

|  |  |
| --- | --- |
| TTL Logic Types   * 74LS08 Quad 2-input * 74LS11 Triple 3-input * 74LS21 Dual 4-input | CMOS Logic Types   * CD4081 Quad 2-input * CD4073 Triple 3-input * CD4082 Dual 4-input |

**მაგალითი: ჩამკეტი 7408, ოთხი „და“ ელემენტი 1 ჩიპში**

|  |
| --- |
| 2-input AND gate 7408 |

ლოგიკური „ან“ ჩამკეტი (Logic "OR" Gates) **Tutorial: 3 of 10**

**განმარტება**

ლოგიკური „ან“ ჩამკეტი ცეულებრივად გამოსავალზე ლოგიკური „0“ -ის მდგომარეობაშია, და გადაირთვება „1“ მდგომარეობაში მაშინ, როდესაც ერთერთ შესავალზე მაინც ჩნდება „1“. გამოსაბვალი უბრუნდება ლოგიკურად დაბალ მდგომარეობას, თუ ყველა სესავალზე დაბალი ლოგიკური მდგომარეობაა. ბულის ალგებრაში ეს ჩამკეტი ასრულებს ლოგიკური აჯამვის ოპერაციას - (+). ამგვარად გვაქვს: A+B = Q.

2 შესავლის შემთხვებაში შეგვიძლია ვისარგებლოდ განმარტებით:

**"თუ A ან B ჭეშმარიტია - Q ჭეშმარიტია"**

**ტრანზისტორული „ან“ ჩამკეტი**

ისევ RTL ლოგიკის გამოყენებით შეგვიძლია ავაგოთ ჩამკეტი. ორივე ტრანზისტორი ერთმანეთს პარალელირად უერთდება. თუ ერთი მაინც ჩართულია, Q გამოსავალზე ვიღებთ „1“.

|  |
| --- |
| 2-input Transistor OR Gate |

ელექტრონიკაში მისაწვდომია ამ ფუნქციის შემსრულებელი სტანდარტული ჩიპები. არსებობს სათანადო სიმბოლური აღნიშვნა.

**სიფრული ლოგიკური „ან“ ჩამკეტი (Digital Logic "OR" Gate)**

**„ან“ ჩამკეტი 2 შესავლით**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input OR gate  **2-input OR Gate** | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |
| Boolean Expression **Q = A+B** | Read as A **OR** B gives Q | | |

**„ან“ ჩამკეტი 3 შესავლით**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | Truth Table | | | |
| 3-input OR gate  **3-input OR Gate** | **C** | **B** | **A** | **Q** |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** |
| **0** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** |
| Boolean Expression **Q = A+B+C** | Read as A **OR** B **OR** C gives Q | | | |

ელექტრონული ინდუსტრია აწარმოებს 2, 3 და 4 შესავლის მქონე ჩამკეტებს. სირთულეს არ წარმოადგენს მრავალი შესავლის მქონე ჩამკერტის სინთეზირება მარტივი ჩამკეტების გაერთიანებით.

|  |
| --- |
| 6-input OR Gate |

სათანადო ბულის გამოსახულება იქნება: **Q = (A+B)+(C+D)+(E+F)**

გამოუყენებელი შესავლები უნდა ჩაირტოს კვების „მიწის“ ელექტროდთან.

ელექტრონიკაში გამოიყენება ცხრილში მოყვანილი „ან“ ჩამკეტები:

|  |  |
| --- | --- |
| TTL Logic Types   * 74LS32 Quad 2-input | CMOS Logic Types   * CD4071 Quad 2-input * CD4075 Triple 3-input * CD4072 Dual 4-input |

**მაგალითი: ჩამკეტი 7432, 4 „ან“ ელემენტი 1 ჩოპში**

|  |
| --- |
| 2-input OR gate 7432 |

ლოგიკური ჩამკეტი „არა“ **Tutorial: 4 of 10**

**განმარტება**

საბაზო ლოგიკური ჩამკეტი „არა“ ყველაზე გავრცელებულია ელექტრონიკაში. მას ხშირად უძახიან „მაინვერტირებელ ბუფერს“, ან ციფრულ ინვერტორს. ჩვეილებრივად მის გამოსავალზე არის ლოგიკური „1“. ლოგიკური „0“ ჩნდება თუ შერსავალს მიეწოდება „1“. სხვა სიტყვებით, ეს ჩამკეტი საწინააღმდეგოთ „აბრუნებს“ შესავალ სიგნალს. ბულის გამოსახულება იქნება A = Q, სადაც Q წარმოადგენს A-ს უარყოფას.

**"თუ A არ არის ჭეშმარიტი, Q ჭეშმარიტია"**

**ტრანზისტორული „არა“ ჩამკეტი**

მარტივი „არა“ ჩამკეტი იქმნება RTL ლოგიკით. ტრანზისტორის შესავალზე უნდა იყოს „0“. იმისათვის რომ გამოსავალზე გაჩნდეს „1“.

|  |
| --- |
| Transistor NOT Gate |

ეს ლოგიკური ჩამკეტი ფართოდ გამოიყენება ლოგიკური ფუნქციების შესასრულებლად. ელექტრონიკაში უარყოფის ფუნქცია გრაფიკულად აღინიშნება პატარა წრით შესავალ ან გამოსავალ ელექტროდზე. ამ ოპერაციით შეგვიძლია -შევქმნათ ფუნქციები „არა“ (NOT), „და-არა (NAND), „ან-არა“ (NOR).

**ციფრული ინვერტორი**

|  |  |  |
| --- | --- | --- |
| Symbol | Truth Table | |
| The NOT gate  **Inverter or NOT Gate** | **A** | **Q** |
| **0** | **1** |
| **1** | **0** |
| Boolean Expression **Q = not A** | Read as inverse of **A** gives Q | |

რადგან ინვერტორს ერთი შესავალი ელექტროდი გააჩნია, ის განსხვავდება ადრე განხილული „გადაწყვეტილების მიმღები“ ჩასმკეტებისაგან, რომლებსაც სხვადასხვა მდგომარეობა გააჩნიათ შესავალი ელექტროდების ნდგომარეობის ვარიანტებიდან გამომდინარე. ელექტრონიკაში გავრცელებულია ჩიპები, რომლებშიც 4 ან 6 ინვერტორია. სიმბოლო (o) შეიძლება შესავალზეც იყოს და ჩამკეტის გამოსავალზეც. ორივე შემთხვევაში ერთი და იგივე ფუნქცია სრულდება.

**სიგნალის ინვერსიის გრაფიკული გამოსახულება**

|  |
| --- |
| Signal Inversion NOT gate  **Bubble Notation for Input Inversion** |

**„არა“ ჩამკეტის ექვივალენტები**

ლოგიკური „არა“ შესაძლებელია შეიქმნას განხილული ჩამკეტების გამოყენებით:

|  |  |
| --- | --- |
| Inverter using NAND and NOR Gates | |
| Single Stage Transistor Inverter | უმარტივესი ინვერტორი შეიძლება შეიქმნას მარტივი ტრანზისტორული სქემის გამოყენებით. როდესაც სქემის შესავალზე არის ლოგიკური „1“, ტრანზისტორი ატარებს დენს და მასზე დაცემული ძაბვა მინიმალურია, ანუ ლოგიკურ „0“-ს უდრის. თუ შესავალზე ნოლია, ტრანზისტორი დაკეტილია და მისი დიდი წინაღობის გამო, გამოსავალი "Q" კვების +Vcc. მნიშვნელობას მიიღებს. |

ამგვარად, "A"-ს მაღალ ლოგიკურ მნიშვნელობაზე გამოსავალი პასუხობს დაბალით და პირიქით, ანუ ინვერტირებით.

**შმიტის ინვერტორები (Schmitt Inverters)**

სტანდარტული ინვერტორი იყენებს ტრანზისტორულ სქემას, რომელსაც, ჩვეულებრივად, გააჩნია სიგნალის შეყოვნება, ანუ სქემა არ იცვლის მდგომარეობას მყისიერად. რადგან ტრანზისტორი დენის გამაძლიერებელია, რომელიც წრფივ რეჟიმში მუშაობს, ნებისმიერი მცირე ვარიაცია შესავალზე აისახება გამოსავალ სიგნალში, ანუ შესაძლებელია ინვერტორმა რამდენიმე ჯერ შეცვალოს თავისი მდგომარეობა ხმაურიანი შესავალი სიგნალის პირობებში. ერთერთი გამოსავალია შმიტის ინვერტორის, ანუ „ეშმაკის“ ინვერტორი.

ადრე განხილულ ჩამკეტებში, ყველა ლოგიკური ციფრული ჩამკეტი იყენებდა ორ ძაბვას ლოგიკური მდგომარეობის განსასაზღვრავად. განსხვავდება ძაბვები ლოგიკური „1“ და „0“-სათვის. TTL ლოგიკისათვის ინტერვალი 2.0v - 5v წარმოადგენს "1", და 8v დაბლა წარმოადგენს "0". შმიტის ინვერტორი ისეა აწყობილი, რომ მისი მდგომარეობა იცვლება, როდესაც სიგნალი შესავალზე აღემატება „ზედა საზღვრის ძაბვას“. გამოსავალი იცვლება, გადადის დაბალ ლოგიკურ მდგომარეობაში და რჩება ამ მდგომარეობაში სანამ შესავალი სიგნალი არ აღმოჩნდება უფრო დაბალი ვიდრე „ქვედა საზღვრის ძაბვა“. ამ მომენტიდან გამოსავალი გადაირთვება მაღალ ლოგიკურ მდგომარეობაში. სხვა სიტყვებით, შმიტის ინვერტორს გააჩნია „გისტერეზისი“. ასეთ შემთხვევაში ჩართვა/გამორთვის პროცესი გაცილებით უფრო მკვეთრია და შეცდომები გამორიცხულია. ეს ინვერტორი იდეალურია დაბალი სიხშირის ანალოგური და ციფრული სიგნალებისათვის: გადართვა ერთი მდგომარეობიდან მეორეში მაინც სწრაფად ხდება.

**შმიტის ინვერტორი**

|  |
| --- |
| Schmitt Inverter |

შმიტის ინვერტორი სასარგებლოა სიგნალების გენერატორებში და სინუსოიდალური სიგნალის მართკუთხა სიგნალშიო გარდასაქმნელად.

**შმიტის ინვერტორი ოსცილატორი & გარდამქმნელი**

|  |
| --- |
| Schmitt Inverter Oscillator and Square Wave Generator |

პირველ სქემაზე ნაჩვენებია მარტივი RC ტიპის ოსცილატორი. საწყის მდგომარეობაში C განმუხტულია, ანუ შესავალს გააჩნია დაბალი ლოგიკური მდგომარეობა. გამოსავალზე მაღალი ლოგიკური მდგომარეობაა. გამოსავალთან არსებული უკუკავშირის გამო, კონდენსატორი იწყებს დამუხტვას დენით, რომელიც გადის R რეზისტორში. როგორც კი დამუხტვის ძაბვა მიაღწევს ზედა საზღვარს, ინვერტორი შეიცვლის მდგომარეობას გამოსავალი დაბალ ლოგიკურ მდგომარეობაში გადავა და კონდენსატორი დაიწყებს განმუხტვას რეზისტორის მეშვეობით. როგორც კი მიღწეული იქნება შმიტის ინვერტორის ქვედა საზღვარი, ინვერტორი ისევ შეიცვლის მდგომარეობას და პროცესი ახალ ციკლს დაიწყებს. შმიტის ინვერტორის ასეთი ჩართვისათვის, დროის 33% ინვერტორი მაღალ ლოგიკურ მდგომარეობაში იმყოფება, ხოლო მის მიერ გენერირებული სიხშირე გამოითვლება ფორმულით: ƒ = 680/RC.

მეორე სქემა გარდაქმნის შესავალ სინუსოიდალურ სიგნალს (ან სხვა ფორმის სიგნალს) მართუთხა სიგნალში. ინვერტორის შესავალი შეერთებულია კვების ძაბვასთან გამყოფის საშუალებით. ეს გაკეთებულია მისი სტატიკური მნიშვნელობის დასაფიქსირებლათ. რადგან შესვალი მიერთებულია სიგნალთან კონდენსატორის მეშვეობით, სიგნალის მუდმივი მნიშვნელობა დაბლოკილია. სიგნალი უშუალოდ ინვერტორის შესავალზე იცვლება სტატიკური წერტილის გარშემო ქვემოდ და ზემოდ. როგორც კი სიგნალი შემცირდება და მიაღწევს ქვედა საზღვარს, ინვერტორი გადაირთვება მაღალ მდგომარეობაში. როდესაც სიგნალი დაიწყებს ზრდას და ავა ზედა საზღვრამდე, გამოსავალი გადაირთვება დაბალ ლოგიკურ მდგომარეობაში. ასე შეიქმნება მართკუთხა ფორმის სიგნალი. ეს გენერატორი ქმნის სიგნალს, რომელიც ფაზაშია შესავალი სიგნალის აღმავალ პერიოდთან. თუ საჭიროა ფაზის შემობრუნება, შეგვიძლია გამოსავალზე ჩავრთოთ მეორე შმიტის ინვერტორი.

ფართოდ გავრცელებულია ლოგიკური ინვერტორები:

|  |  |
| --- | --- |
| TTL Logic Types   * 74LS04 Hex Inverting NOT Gate * 74LS04 Hex Inverting NOT Gate * 74LS14 Hex Schmitt Inverting NOT Gate * 74LS1004 Hex Inverting Drivers | CMOS Logic Types   * CD4009 Hex Inverting NOT Gate * CD4069 Hex Inverting NOT Gate |

**მაგალითი: შმიტის ინვერტორი 7404**

|  |
| --- |
| Inverter NOT gate 7404 |

ლოგიკური „არა-და“ ჩამკეტი  **Tutorial: 5 of 10**

**განმარტება**

ლოგიკური „არა-და“ ჩამკეტი წარმოადგენს „და“ და „არა“ ჩამკეტების კომბინაციას. ეს ჩამკეტები მიმდევრობით უნდა შევაერთოდ. ჩამკეტი (NAND (Not - AND)) ჩვეულებრივ მდგომარეობაში ლოგიკურ "1" გვიჩვენებს გამოსავალზე და გადადის დაბალ ლოგიკურ მდგომარეობაში, როდესაც **ყველა** მისი შესავალი მაღალ ლოგიკურ დონეზეა . ეს ჩამკეტი ადრე განხილული „და“ ჩამკეტის რევერსულია.

**ლოგიკური „არა-და“ჩამკეტის ექვივალენტი**

|  |
| --- |
| 2-input NAND Gate |

ამ ჩამკეტის ბულის გამოსახულება წარმოადგენს ლოგიკურ გამრავლებას რომელიც უარყოფითია „და“ ოპერაციის მიმართ. უნდა გამოვიყენოთ სიმბოლო (.) და ხაზი, ( ‾‾ ), რომელიც უარყოფას ნიშნავს და ნამრავლის გამოსახულების ზემოდ უნდა იყოს. ბულის შედეგი ასე დაიწერება: A.B = Q.

ჩასმკეტის განმარტებას ასეთი სახე ექნება:

**"თუ A ან B მცდარია, მაშინ Q ჭეშმარიტია"**

**ტრანზისტორული „არა-და“ ჩამკეტი**

მარტივი ჩამკეტი ორი შესავალი ელექტროდით იქმნება RTL ლოგიკის გამოყენებით. ერთიც და მეორე ტრანზისტორიც გამორთავს დენს და ქმნის მაღალ წინაღობას. ორივე ერთად რთავენ დენს და „ამიწებენ“ გამოსავალს.

|  |
| --- |
| 2-input Transistor NAND Gate |

„არა-და“ ჩამკეტსგააჩნია მრავალი ფუნქცია ციფრულ ელექტრონიკაში. გააჩნია თავისი სიმბოლო, რომელიც „და“ ჩამკეტის სომბოლოს წარმოადგენს გამოსავლის უარყოფით.

**ციფრული ლოგიკური „არა-და“ ჩამკეტი**

**„არა-და“ ჩამკეტი 2 შესავალი ელექტროდით**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input NAND gate  **2-input NAND Gate** | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |
| Boolean Expression | Read as A **AND** B gives **NOT** Q | | |

**„არა-და“ ჩამკეტი 3 შესავალი ელექტროდით**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | Truth Table | | | |
| 3-input NAND gate  **3-input NAND Gate** | **C** | **B** | **A** | **Q** |
| **0** | **0** | **0** | **1** |
| **0** | **0** | **1** | **1** |
| **0** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **0** |
| Boolean Expression **Q =** | Read as A **AND** B **AND** C gives **NOT** Q | | | |

ისევე როგორც „და“ ჩამკეტისათვის, წარმოება უშვებს 3, 3 და 4 შესავლიან ჩამკეტებს, მაგრამ მათი გაერთიანება და ლოგიკური შესავლების რაოდენობის გაზრდა არ წარმოადგენს სირთულეს. ასეთი კასკადი ნაჩვენებია ნახატზე:

**„არა-და“ ჩამკეტი 4 შესავალი ელექტროდით**

|  |
| --- |
| 4-input NAND Gate |

ბულის გამოსახულება: **Q =**

გამოუყენებელი ელექტროდი უნდა ჩავრთით კვების წყაროსთან ლოგიკური „1“ -ის უზრუნველსაყოფად.

ამ ჩამკეტის ოპერაცია ცნობილია აგრეთვე შეფერის შტრიხის სახელით: |, ან ↑. მაგალითად A NAND B = A|B ან A↑B.

**უნივერსალური „არა-და“ ჩამკეტი**

ეს ჩამკეტი უნივერსალურია, რადგან ყველა სხვა ჩამკეტი შესაძლებელია შეიქმნას ამ ჩამკეტის გამოყენებით. ჩამკეტი წარმოადგენს ძირითად ზაბაზო ელემენტს პრაქტიკული ციფრული ელექტრონიკისათვის.

**სხვადასხვა ჩამკეტი აგებული „არა-და“ ჩამკეტის გამოყენებით**

|  |
| --- |
| Logic Gates using NAND Gates |

ჩამკეტი წარმოდგენილია ჩიპებით:

|  |  |
| --- | --- |
| TTL Logic Typ   * 74LS00 Quad 2-input * 74LS10 Triple 3-input * 74LS20 Dual 4-input * 74LS30 Single 8-input | CMOS Logic Types   * CD4011 Quad 2-input * CD4023 Triple 3-input * CD4012 Dual 4-input |

**მაგალითი: ჩამკეტი NAND Gate 7400**

|  |
| --- |
| 2-input NAND gate 7400 |

ლოგიკური „ან-არა“ ჩამკეტი **Tutorial: 6 of 10**

**განმარტება**

ადვილი მისახვედრია, რომ ეს ჩამკეტი უკვე განხილული „ან“ ჩამკეტის უარყოფაა (ინვერტირება ან რევერსირება), და მიიღება ამ ორი ოპერაციის მიმდევრობით განხორციელებით. ჩვეულებრივად ჩამკეტის გამოსავალი მაღალ ლოგიკურ მდგომარეობაშია - „1“, და გადადის დაბალ მდგომარეობაში, როდესაც ერთი მაინც შესავალი ელექტროდი მაღალ მდგომარეობაშია.

**„ან-არა“ ჩამკეტის ექვივალენტი**

|  |
| --- |
| 2-input NOR Gate |

ამ ჩამკეტის ბულის გამოსახულება ლოგიკური შეკრების (+) და უარყოფის ( ‾‾ ) ოპერაციებს იყენებს, A+B = Q.

**"თუ A და B მცდარია, Q მცდარია"**

**ტრანზისტორული „ან-არა“ ჩამკეტი**

RTL ლოგიკით აგებული ჩამკეტი ნაჩვენებია ნახატზე. ორივე შესავალი პარალელურად ჩართული ტრანზისტორების ბაზებს მართავს

|  |
| --- |
| 2-input Transistor NOR Gate |

**ციფრული „ან-არა“ ჩამკეტი**

**2 შესავალი ელექტროდის მქონე ჩამკეტი (2-input NOR Gate)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input NOR gate  **2-input NOR Gate** | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **0** |
| Boolean Expression **Q =** | Read as A **OR** B gives **NOT** Q | | |

**3 შესავალი ელექტროდის მქონე ჩამკეტი (3-input NOR Gate)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | Truth Table | | | |
| 3-input NOR gate  **3-input NOR Gate** | **C** | **B** | **A** | **Q** |
| **0** | **0** | **0** | **1** |
| **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** |
| Boolean Expression **Q =** | Read as A **OR** B **OR** C gives **NOT** Q | | | |

მისაწვდომია 2, 3, და 4 შესავალის მწონე ჩიპები. კასკადური ჩართვით შესაძლებელია შესავალი ელექტროდების რაოდენობის გაზრდა.

**4 შესავალის მქონე ჩამკეტი**

|  |
| --- |
| 4-input NOR Gate |

ბულის გამოსახულება **Q =**

გამოუყენებელი ელექტროდები უერთდება „მიწას“ დამწევი რეზისტორების გამოყენებით.

ამ ჩამკეტის ფუნქცია ცნობილია აგრეთვე პერსის ფუნქციის სახელწოდებით: A↓B.

**უნივერსალური „ან-არა“ ჩამკეტი**

ეს ჩამკეტიც უნივერსალურია და იძლევა ყველა სხვა ჩამკეტის აგების საშუალებას:

**მაგალითები:**

|  |
| --- |
| Logic Gates using NOR Gates |

ჩამკეტების ჩიპები:

|  |  |
| --- | --- |
| TTL   * 74LS02 Quad 2-input * 74LS27 Triple 3-input * 74LS260 Dual 4-input | CMOS   * CD4001 Quad 2-input * CD4025 Triple 3-input * CD4002 Dual 4-input |

**მაგალითი: ჩამკეტი 7402**

|  |
| --- |
| 2-input NOR gate 7402 |

„გამომრიცხავი ან“ ჩამკეტი **Tutorial: 7 of 10**

**განმარტება**

ადრე განვიხულეთ ორი შესავალი ელექტროდის მქონე „ან“ ჩამკეტი. თუ A = "1", ან B = "1", ან ორივე A + B = "1" მაშინ გამოსავალზე იქნება "1". გამოსავალზე მაღალი ლოგიკური დონეა ორ შემთხვევაში. შესაძლებელია შევქმნათ ისეთი ჩამკეტი, რომელიც იღებს მაღალ ლოგიკურ დონეს მხოლოდ ერთ შემთხვევაში - როდესაც მხოლოდ A = "1" ან მხოლოდ B = "1" და არა ორივე ერთდროულად. ეს „გამომრიცხავი ან“ (Exclusive-OR ან Ex-Or). ეს ჩამკეტი გასმორიცხავს შემთხვევას როდესაც ორივე შესავალი მაღალია.

სხვა სიტყვებით, ჩამკეტის გამოსავალზე მაღალი ლოგიკური მდგომარეობაა მხოლოდ იმ შემთხვევაში, როდესაც შესავლების ლოგიკური დონეები განსხვავდებიან. გამორიცხულია შემთხვევა როდესაც ორივე შესავალი ერთდროულად ან ჭეშმარიტია ან მცდარია.

ამ ჩამკეტის ბულის გამოსახულებაა: = . + .

ჩამკეტის სქემა იქმნება ცნობილი ჩამკეტების კომბინაციით. ქვემოთ მოყვანილია მაგალითები.

**ციფრული ლოგიკური „გამომრიცხავი-ან“ ჩამკეტი ("Ex-OR" Gate)**

**ჩამკეტი 2 შესავლით**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input Ex-OR Logic Gate  **2-input Ex-OR Gate** | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |
| Boolean Expression Q = AB | Read as A **OR** B but NOT **BOTH** gives Q | | |

შესაძლებელია ბულის გამოსახულება ასე წარმოვადგინოთ: Q = (A B) = (A+B).() . ეს გვაძლევს საშუალებას შევქმნად ჩამკეტის ექვივალენტური სქემა უკვე ცნობილი ჩამკეტების გამოყენებით.

**ჩამკეტის ექვივალენტური სქემა**

|  |
| --- |
| 2-input EX-Or Gate Equivalent |

ამ სქემაშიგამოყენებულია სამი სხვადასხვა ტიპის ჩამკეტი. ეს შეიძლება უხერხული აღმოჩნდეს კონკრეტული ელექტრონული მოწყობილობის შექმნის დროს. შესაძლებელია სხვა ვარიანტის რეალიზაცია. აქ ერთი ტიპის ჩამკეტებია, თანაც არსებობს ჩიპი რომელშიც ზუსტად ოთხი „და-არა“ ჩამკეტია. ამგვარად „გამომრიცხავი-ან“ ჩამკეტი უფრო მოხერხებულად, მხოლოდ ერთ ჩიპზე იქნებს განხორციელებული.

**ჩამკეტის ფუნქციის რეალიზაცია „და-არა“ ჩამკეტების გამოყენებით**

|  |
| --- |
| 2-input EX-OR Gate Equivalent using NAND Gates |

„გამომრიცხავი ან“ ჩიპებო:

|  |  |
| --- | --- |
| TTL   * 74LS86 Quad 2-input | CMOS   * CD4030 Quad 2-input |

**მაგალითი: ჩამკეტი 7486**

|  |
| --- |
| 2-input Ex-OR gate 7486 |

„გამომრიცხავი ან არა“ ჩამკეტები **Tutorial: 8 of 10**

**განმარტება**

„გამომრივხავი ან არა“ ჩამკეტი წარმოადგენს „გამომრივხავი ან“ ჩამკეტის რებერსს. ანუ მისი გამოსავალი შეცვლილია საწიმააღმდეგოზე.

**„გამომრიცხავი არა-ან“ ჩამკეტის ექვივალენტი**

|  |
| --- |
| 2-input Ex-NOR Gate |

**„გამომრიცხავი არა-ან“ ჩამკეტი 2 შესავალით**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input Ex-NOR Logic Gate  **2-input Ex-NOR Gate** | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |
| Boolean Expression Q = | Read if A **AND** B the **SAME** gives Q | | |

**„გამომრიცხავი არა-ან“ ჩამკეტის ექვივალენტური წრედი**

|  |
| --- |
| 2-input EX-NOR Gate Equivalent |

Aეს სქემაც იყენებს სხვადასხვა ტიპის ჩამკეტებს, ამიტომ უკეტესია ამ ფუნქციის შესრულება ერთი ტიპის ჩამკეტის გამოყენებით.

**ფუნქციის რეალიზაცია „არა-და“ ჩამკეტების გამოყენებით**

|  |
| --- |
| 2-input EX-NOR Gate Equivalent using NAND Gates |

ფუნქციის ინტეგრალური ჩიპები:

|  |  |
| --- | --- |
| TTL Logic Types   * 74LS266 Quad 2-input | CMOS Logic Types   * CD4077 Quad 2-input |

**მაგალითი: ჩამკეტი 74266**

|  |
| --- |
| 2-input Ex-NOR gate 74266 |

სამი მდგომარეობის მქონე ბუფერები **Tutorial: 9 of 10**

**განმარტება**

ადრე განხილულ „არა“ ჩამკეტს, რომელიც იცვლის გამოსავლის ნდგომარეობას შესავალი ელექტროდის მდგომარეობის საწინააღმდეგოთ, ემატება ერთი ჩამკეტი, რომელიც „ბუფერის“, „ციფრული ბუფერის“, ან „არა მაინვერტირებელი ბუფერის“ სახელებით არის ცნობილი.

ციფრული ბუფერი არ ცვლის ან აწარმოებს ლოგიკურ ოპერაციას, პირიქიტ, ლოგიკური მდგომარეობა მის გამოსავალზე ზუსტად უნდა ემთხვევოდეს მდგომარეობას შესავალზე. მისი ბულის გამოსახულებაა: Q = A.

შეგვიძლია ასე განვმარტოდ ეს ფუნქცია:

**"თუ A ჭეშმარიტია, მაშინ Q ჭეშმარიტია"**

**სამი მდგომარეობის მქონე ბუფერი**

|  |  |  |
| --- | --- | --- |
| Symbol | Truth Table | |
| A Tri-state Buffer  **Buffer** | **A** | **Q** |
| **0** | **0** |
| **1** | **1** |
| Boolean Expression **Q = A** | Read as **A** gives **Q** | |

ციფრული ბუფერი შეგვიძლია შევქმნათ ორი „არა“ ჩამკეტისაგან. ორჯერ ინვერტირება ისევ საწყის ლოგიკურ მდგომარეობას დაუბრუნებს გამოსავალს.

**ორმაგი ინვერსია „არა“ ჩამკეტების გამოყენებით (Double Inversion)**

|  |
| --- |
| Tri-state Buffer using NOT gates |

პირველი შეხედვით, ბუფერის გამოყენება უაზროა - ლოგიკის თვალსაზრისით არაფერი არ იცვლება. მაგრამ ტექნიკური თვალსაზრისით ბუფერების არსებლობა აუცილებელია სიმძლავრეების შესათანხმებლად. ბუფერი პრაქტიკულად წარმოადგენს სიმძლავრის გამაძლიერებელს. შესაძლებელია ჩამკეტს არ ეყოს სიმძლავრე სქემის მრავალი ლოგიკური განშტივების უზრუნველსაყოფად. ამ შემთხვევაში ბუფერს ვიყენებთ.

**ბუფერის განშტოვების მაგალითი**

|  |
| --- |
| Buffer Fan-out example |

ბუფერების გამოყენებით 50-მდე განშტოვების შექმნაა შესაძლებელი.

**სამი მდგომარეობის ბუფერი ("Tri-state Buffer")**

ელექტრონულ სქემებში კიდევ ართი ტიპის ბუფერს იყენებენ. გარდა ლოგიკური დონეებისა, ელექტრონიკაში დენის ჩამრთველები არსებობენ. ბუფერს შეუძლია ეს ფუნქციაც შეასრულოს. ამისათვის ბუფერს მართვის ელექტროდი უნდა გააჩნდეს. ამ ელექტროდით ბუფერი უნდა გადაიოდეს ე.წ. მაღალი იმპედანსის მდგომარეობაში, ანუ გამორთული ელექტრული წრედის მსგავს მდგომარეობაში. ასეთ ბუფერს სამი მდგომარეობა გააჩნია: ორი ლოგიკური და ერთი მაღალი იმპედანსის. ამ მდგომარეობას "Hi-Z" მდგომარეობა, ან "Z" მდგომარეობა ეწოდება.

**აქტიური „მაღალი“ სამი მდგომარეობის ბუფერი (Active "HIGH" Tri-state Buffer)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| A Tri-state Buffer  **Tri-state Buffer** | Enable | A | Q |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | Hi-Z |
| 0 | 1 | Hi-Z |
| Read as Output = Input if Enable is equal to "1" | | | |

ასეთი ბუფერი შეიძლება იმართებოდეს მართველი ელექტროდის როგორც მაღალი ასევე დაბალი ლოგიკური დონით.

**აქტიური „დაბალი“ სამი მდგომარეობის ბუფერი (Active "LOW" Tri-state Buffer)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| A Tri-state Buffer  **Tri-state Buffer** | Enable | A | Q |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | Hi-Z |
| 1 | 1 | Hi-Z |
| Read as Output = Input if Enable is **NOT** equal to "1" | | | |

**სამი მდგომარეობის მქონე ბუფერების მართვა**

სამი მდგომარეობის მქონე ბუფერების გამოყენება ძალიან მნიშვნელოვანია სისტემებში სადაც საჭიროა ინფორმაციის ნაკადის მართვა ან ინფირმაციულ მაგისტრალებთან ურთიერთქმედება. ბუფერების სინქრონიზაციით მიიღწევა ინფორმაციის სხვადასხვა მოწყობილობებიდან წალითხვა და სათანადო ადგილას გადაცემა.

**სამი მდრომარეობის ბუფერის მართვა**

|  |
| --- |
| Tri-state Buffer Control |

ნახატზე ნაჩვენები ბუფერები უერთდებიან ერთინფორმაციულ მაგისტრალს. მართვის სათანადო ორგანიზაციით შესაძლებელია ინფორმაციის სხვადასხვა წყაროებიდან ერთ მაგისტრალში ჩართვა, გადაცემა და სათანადო მოწყობილობით წაკითხვა.

ჩამკეტები წარმოდგენილია ჩიპებით:

|  |  |
| --- | --- |
| TTL Logic Types   * 74LS07 Hex Non-inverting Buffer   74LS17 Hex Buffer/Driver   * 74LS244 Octal Buffer/Line Driver * 74LS245 Octal Bi-directional Buffer | CMOS Logic Types   * CD4050 Hex Non-inverting Buffer * CD4503 Hex Tri-state Buffer * HEF40244 Octal Buffer with 3-state Output |

**მაგალითი: ციფრული არამაინვერტირებელი ბუფერი Buffer 7407**

|  |
| --- |
| Digital Hex Buffer 7407 |

**მაგალითი: სამი მდგომარეობის მქონე ბუფერი Buffer 74244**

|  |
| --- |
| Octal Tri-state Buffer 74244 |

ლოგიკური ჩამკეტები, დასკვნა  **Tutorial: 10 of 10**

ჩვენ დავინახეთ, რომ არსებობს ლოგიკური ჩამკეტების სამი ძირითადი საყრდენი ტიპი: „და“, „ან“, აგრეთვე „არა“ ჩამკეტი. დავინახეთ აგრეთვე, რომ არსებობენ ამ ჩამკეტების უარმყოფი (დამატებითი, რევერსიული) ფორმები „არა-და“, „არა-ან“, და „ბუფერი“. ველა ეს ჩამკეტი შეგვიძლია შევაერთოდ ერთმანეთთან და შევქმნათ ე.წ. „ცომბინატორული ლოგიკის“ რთული წრედები. .

დავინახეთ, აგრეთვე, რომ „არა-და“, „არა-ან“ ჩამკეტები წარმოადგენენ უნივერსალურ ჩამკეტებს, ანუ მათი გამოყენებით შესაძლებელია სხვა ჩამკეტების შექმნა. ამასთან ერთად „არა“ ჩამკეტი და „ბუფერი“, მიუხედავად იმისა, რომ ერთი შესავალი გააჩნიათ, შესაძლებელია სამ მდგომარეობაში იმყოფებოდნენ (**Tri-state**). ერთერთი მათგანი - მაღალი იმპედანსის მდგომარეობა, ხელსაყრელია ინფორმაციის ნაკადის კონტროლისათვის ინფორმაციულ მაგისტრალებში.

ციფრული ლოგიკური ჩამკეტები შეიძლება შეიქმნას რეზისტორების, ტრანზისტორების და დიოდების გამოყენებით, მაგრამ თანამედროვე ციფრული ჩიპების 74xxx სერია იყენებს **TTL** (ტრანზისტორ-ტრანზისტორულ) ლოგიკას აგებულს NPN ბიპოლარულ ტრანზისტორებზე. გამოიყენება, აგრეთვე, უფრო სწრაფი და ეკონომიური ტექნოლოგია ველის ტრანზისტორების გამოყენებით.

ქვემოდ, დასკვნის სახით, წარმოდგენილია რვა ძირითადი სტანდართული ციფრული ლოგიკური ჩამკეტი, მოყვანილია მათი გრაფიკული აღნიშვნა და ჭეშმარიტების ცხრილები.

**„და“ ჩამკეტი (AND Gate)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input AND gate | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |
| Boolean Expression Q = A.B | Read as A **AND** B gives Q | | |

**„არა და“ ჩამკეტი (NAND Gate)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input NAND gate | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |
| Boolean Expression Q = | Read as A **AND** B gives **NOT** Q | | |

**„ან“ ჩამკეტი (OR Gate)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input OR gate | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |
| Boolean Expression Q = A + B | Read as A **OR** B gives Q | | |

**„არა ან“ (NOR)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input NOR gate | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **0** |
| Boolean Expression Q = | Read as A **OR** B gives **NOT** Q | | |

**„გამომრიცხავი ან“ (Exclusive-OR Gate (Ex-OR))**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input Ex-Or Logic Gate | **B** | **A** | **Q** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |
| Boolean Expression Q = AB | Read as A **OR** B but not **BOTH** gives Q | | |

**„გამომრიცხავი არა და“ ჩამკეტი (Exclusive-NOR Gate (Ex-NOR)**

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Truth Table | | |
| 2-input Ex-Nor Logic Gate | **B** | **A** | **Q** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |
| Boolean Expression Q = | Read if A **AND** B the **SAME** gives Q | | |

**ბუფერი**

|  |  |  |
| --- | --- | --- |
| Symbol | Truth Table | |
| A Buffer | **A** | **Q** |
| **0** | **0** |
| **1** | **1** |
| Boolean Expression Q = A | Read as **A** gives **Q** | |

**არა (NOT) ჩამკეტი (ინვერტორი)**

|  |  |  |
| --- | --- | --- |
| Symbol | Truth Table | |
| The NOT gate | **A** | **Q** |
| **0** | **1** |
| **1** | **0** |
| Boolean Expression Q = not A or A | Read as inverse of **A** gives Q | |

მოყვანილი სიფრული ლოგიკური ჩამკეტების და მათი ბულის გამოსახულებები შესაძლებელია გამოვსახოთ ერთიანი ცხრილით. ამ ცხრილში მოყვანილია შესავალი და გამოსავალი ელექტროდების მდგომარეობების ყველა კომბინაცია.

**ჭეშმარიტების ცხრილები, დასკვნა**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | Truth Table Outputs for 2-input Logic Gates | | | | | |
| **B** | **A** | **AND** | **NAND** | **OR** | **NOR** | **EX-OR** | **EX-NOR** |
| **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** | **0** | **0** | **1** |

|  |  |  |
| --- | --- | --- |
| Truth Table Output for Single-input Gates | | |
| **A** | **NOT** | **Buffer** |
| **0** | **1** | **0** |
| **1** | **0** | **1** |

**ანწევი და დამწევი რეზისტორები**

დაიმახსოვრედ, რომ ლოგიკური სქემების სინთეზის დროს ყველა გამოუყენებელი შესავალი უნდა იყოს შეერთებული ან ლოგიკურ ერთიანთან ან ნოლთან სათანადო „ამწევი“("Pull-up"), ან „დამწევი“ (Pull-down") რეზისტორების გამოყენებით. ამით უზრუნველყოფილი იქნება ფიქსირებული ლოგიკური სიგნალი. ამით გამოირიცხება ჩამკეტების უმართავი ქცევა.

|  |
| --- |
| Pull-up and Pull-down Resistors |

|  |
| --- |
|  |