Universiteti i Prishtinës "Hasan Prishtina" Kosovë

Arkitektura e kompjuterit

Pjesa 3 – Algjebra e Boole-it dhe logjika dixhitale

Prof. Asoc. Dr. Ermir Rogova

Objektivat

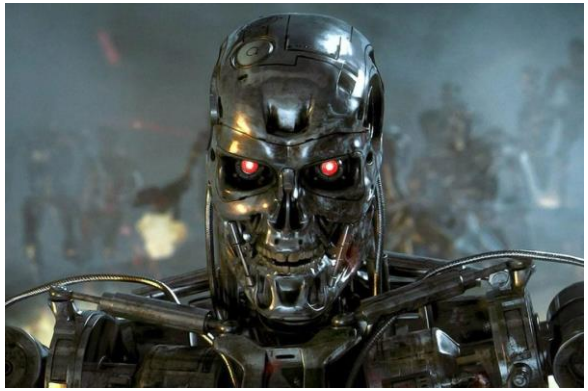
- Të kuptojmë raportin në mes të logjikës së Boole-it dhe çarqeve të kompjuterëve dixhitalë.
- Të mësojmë për disenjimin e çarqeve të thjeshta logjike.
- Të kuptojmë se si bashkëveprojnë çarqet dixhitale në kuadër të sistemeve komplekse kompjuterike.

Hyrje

- Në fund të shekullit të XIX, George Boole i provokoi rëndë matematikanët dhe filozofët e kohës duke sugjeruar se logjika mund të shprehet përmes formulave
  - Si guxon dikush të pohojë se mendimi njerëzor mund të shprehet dhe manipulohet si formulë matematikore?
- Kompjuterët, ashtu si i njohim ne sot janë zbatime praktike të teorisë së Boole-it:
  - John Atanasoff dhe Claude Shannon ishin ndër të parët që e vërejtën këtë lidhje

Hyrje

- Nga mesi i shekullit XX kompjuterët njiheshin se "makina që mendojnë" dhe "mendje elektronike"
  - Shumë njerëz frikësoheshin prej tyre
- Në kohën e sotme, ne rallë e vejmë në pyetje raportin në mes të kompjuterit elektronik dixhital dhe logjikës njerëzore. Kompjuterët pranohen si një pjesë e të përditshmes tonë
  - Përsëri, shumë njerëz frikësohen prej tyre.
- Këtu do të shohim se sa e thjeshtë është bërthama e një makine



Algjebra e Boole-it

- Algjebra e Boole-it është sistem matematikor për manipulim me ndryshore që mund të kenë njërën nga dy vlerat e mundshme
  - Në logjikën formale këto vlera janë "e saktë" dhe "jo e saktë", ndërsa ndryshoret quhen gjykime
  - Në sisteme dixhitale, vlerat janë "on" dhe "off", 1 dhe 0, ose "lartë" dhe "ulët"
- Shprehjet Boole-ane (logjike) fitohen duke kryer veprime me ndryshoret e Boole-it
  - Veprimet e zakonshme janë: NOT, AND dhe OR

Algjebra e Boole-it

- Operatorët e Boole-it (logjikë) përshkruhen me anë të tabelave të saktësisë
- Tabelat e saktësisë për AND dhe OR janë dhënë djathtas
- AND quhet edhe prodhimi logjik, ndërsa OR shuma logjike

X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Algjebra e Boole-it

- Tabela e saktësisë për NOT është dhënë djathtas
- Operatori NOT rëndom shënohet me "mbivizim"

NOT x

x	$\overline{x}$
0	1
1	0

Algjebra e Boole-it

- Funksioni logjik (i Boole-it) ka të paktën:
  - Një ndryshore logjike,
  - Një veprim logjik, dhe
  - Të paktën një të dhënë hyrëse (input) nga bashkësia {0,1}
- Ai prodhon një rezultat (output) poashtu nga bashkësia {0,1}

Tani e kemi të qartë se pse sistemi binar është aq i dobishëm për kompjuterët dixhitalë

Algjebra e Boole-it

- Tabela e saktësisë për funksionin logjik

$$F(x,y,z) = x\bar{z}+y$$

është dhënë djathtas

- Për lehtësi kalkulimi tabela përmban edhe shtylla ndihmëse që i ruajnë mesrezultatet

$F(x,y,z) = x\bar{z}+y$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Algjebra e Boole-it

- Si në aritmetikë veprimet logjike kanë rendin e përparësisë
- Operatori NOT ka përparësi, ndërsa pas tij vijnë AND dhe OR
- Në bazë të kësaj i zgjedhim edhe shtyllat në tabelë

$F(x,y,z) = x\bar{z}+y$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Algjebra e Boole-it

- Kompjuterët dixhitalë përbëhen prej çarqeve që implementojnë funksione logjike
- Sa më i thjeshtë që është funksioni logjik, aq më i thjeshtë do të jetë çarku përkatës
  - Çarqet e thjeshta janë me të lira për t'u prodhuar, shpenzojnë më pak energji dhe punojnë më shpejtë se çarqet komplekse
- Duke pasur parasysh këtë, ne gjithnjë tentojmë që funksioni logjik të reduktohet në formën më të thjeshtë të mundur
- Ekzistojnë shumë identitete logjike që na ndihmojnë ta bëjmë këtë

Algjebra e Boole-it

- Shumica e identiteteve logjike kanë edhe formën e prodhimit (AND), edhe të shumës logjike (OR).

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$x\bar{x} = 0$	$x + \bar{x} = 1$

Algjebra e Boole-it

Identity Name	AND Form	OR Form
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$

Identity Name	AND Form	OR Form
Absorption Law	$x(x+y) = x$	$x + xy = x$
DeMorgan's Law	$\overline{(xy)} = \bar{x} + \bar{y}$	$\overline{(x+y)} = \bar{x}\bar{y}$
Double Complement Law	$\overline{(\bar{x})} = x$	

Algjebra e Boole-it

- Identitetet logjike mund t'i përdorim për ta thjeshtësuar funksionin $F(X,Y,Z) = (X + Y)(X + \overline{Y})\overline{(X\overline{Z})}$

si vijon:

$(X + Y)(X + \overline{Y})\overline{(X\overline{Z})}$	Idempotent Law (Rewriting)
$(X + Y)(X + \overline{Y})(\overline{X} + Z)$	DeMorgan's Law
$(XX + X\overline{Y} + XY + Y\overline{Y})(\overline{X} + Z)$	Distributive Law
$((X + Y\overline{Y}) + X(Y + \overline{Y}))(\overline{X} + Z)$	Commutative & Distributive Laws
$((X + 0) + X(1))(\overline{X} + Z)$	Inverse Law
$X(\overline{X} + Z)$	Idempotent Law
$X\overline{X} + XZ$	Distributive Law
$0 + XZ$	Inverse Law
XZ	Idempotent Law

Algjebra e Boole-it

- Ndonjëherë është më lirë të ndërtohet çarku duke përdorur komplementin e një funksioni se sa të implementohet direkt funksioni
- Ligji i deMorgan-it ofron lehtësi për gjetjen e komplementit të funksionit logjik
- Sipas Ligjit të deMorganit:

$$\overline{(xy)} = \bar{x} + \bar{y} \quad \text{and} \quad \overline{(x+y)} = \bar{x}\bar{y}$$

Algjebra e Boole-it

- Ligji i deMorganit mund të zgjerohet për çfarëdo numri të ndryshoreve
- Për këtë qëllim zëvendësojmë çdo ndryshore me komplementin e saj dhe i ndërrojmë të gjithë operatorët AND në OR, ndërsa ata OR në AND
- Kështu gjejmë se komplementi i: $F(X,Y,Z) = (XY) + (\bar{X}Z) + (Y\bar{Z})$

është:

$$\begin{aligned}\bar{F}(X,Y,Z) &= \overline{(XY) + (\bar{X}Z) + (Y\bar{Z})} \\ &= \overline{(XY)}\,\overline{(\bar{X}Z)}\,\overline{(Y\bar{Z})} \\ &= (\bar{X}+\bar{Y})(X+\bar{Z})(\bar{Y}+Z)\end{aligned}$$

Algjebra e Boole-it

- Me shembuj mund të tregohet se ka shumë mënyra për prezantimin e shprehjes logjike të njëjtë
  - Këto forma janë logjikisht ekuivalente
  - Shprehjet logjikisht ekuivalente kanë tabela të njëjta të saktësisë
- Që të eliminohet konfuzioni, funksionet logjike shprehen në formë kanonike ose standarde

Algjebra e Boole-it

- Ekzistojnë dy forma kanonike për shprehjet logjike: shuma e prodhimeve dhe prodhimi i shumave
  - Të rikujtojmë se shuma logjike është veprimi OR, ndërsa prodhimi logjik është veprimi AND
- "Shuma e prodhimeve".
  - Shembull: $F(x,y,z) = xy + xz + yz$
- "Prodhimi i shumave":
  - Shembull: $F(x,y,z) = (x+y)(x+z)(y+z)$

Algjebra e Boole-it

- Nuk është vështirë të konvertohet funksioni në formën e shumës së prodhimeve duke përdorur tabelën e saktësisë
- Na interesojnë vlerat e ndryshoreve që e bëjnë funksionin të saktë (=1)
- Grupet e ndryshoreve të tilla "mblidhen" duke u lidhur me OR

$F(x,y,z) = x\bar{z}+y$

x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Algjebra e Boole-it

- Për funksionin tonë kemi:

$$F(x,y,z) = \bar{x}y\bar{z}+\bar{x}yz+x\bar{y}\bar{z}+xy\bar{z}+xyz$$

Edhepse ky funksion nuk është më i thjeshti, qëllimi ynë është që ta shkruajmë në formë kanonike

$$F(x,y,z) = x\bar{z}+y$$

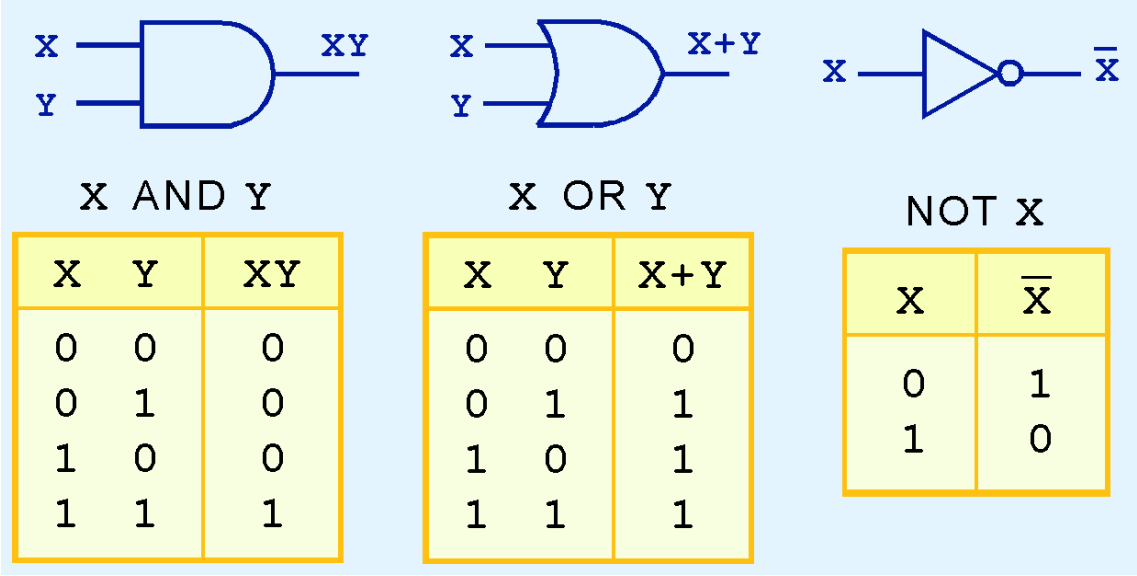
x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Portat logjike

- Këtu do të shohim se si implementohen funksionet logjike në çarqe të kompjuterëve dixhitalë të quajtura porta (gates)
- Porta është një pajisje elektronike që prodhon rezultat bazuar në dy ose më shumë vlera hyrëse
  - Në realitet, porta përbëhet prej 1-6 transistorëve, por dizanjuesit e konsiderojnë atë si njësi të vetme
  - Çarqet e integruara përmbajnë grupe të portave të ndërtuara për qëllime të ndryshme

Portat logjike

- Tri portat më të thjeshta janë: AND, OR dhe NOT



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

X	$\overline{X}$
0	1
1	0

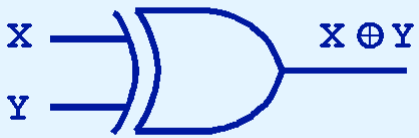
- Ato i përgjigjen veprimeve logjike të definuara me tabelat e saktësisë

Portat logjike

- Një portë tjetër shumë e dobishme është disjunksioni ekskluziv (XOR)

X XOR Y

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0



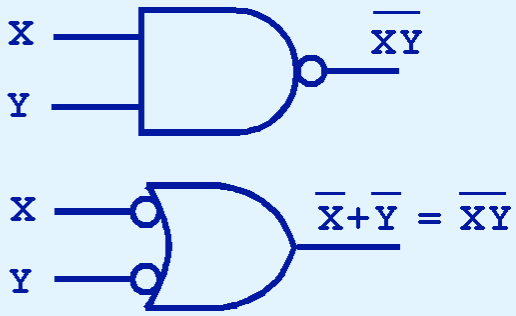
Simboli special për XOR është ⊕

Portat logjike

- NAND dhe NOR janë dy porta tjera të rëndësishme

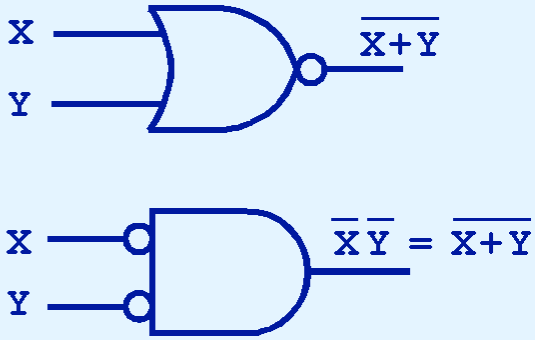
X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



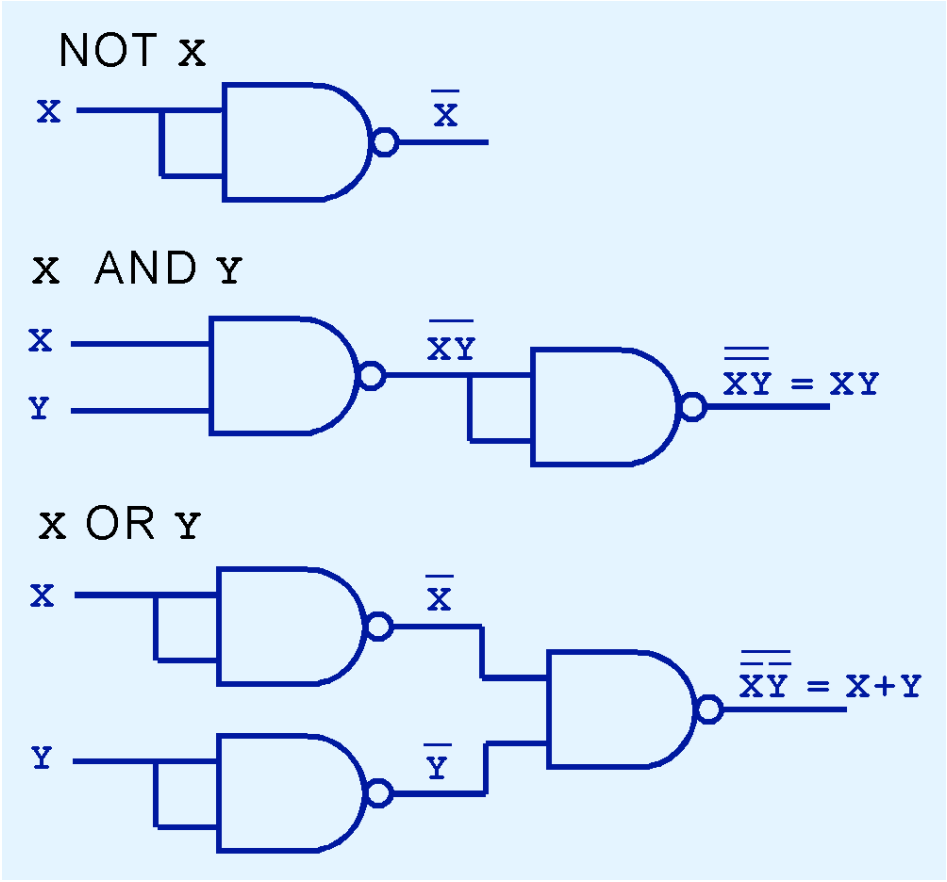
X NOR Y

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



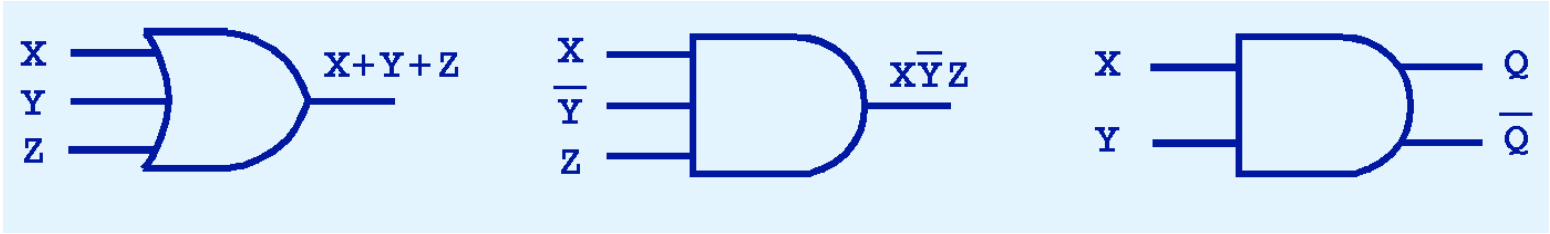
Portat logjike

- NAND and NOR njihen si porta universale sepse prodhimi i tyre nuk është i shtrenjtë, ndërsa çdo funksion logjik mund të konstruktohet duke përdorur vetëm portat NAND dhe NOR



Portat logjike

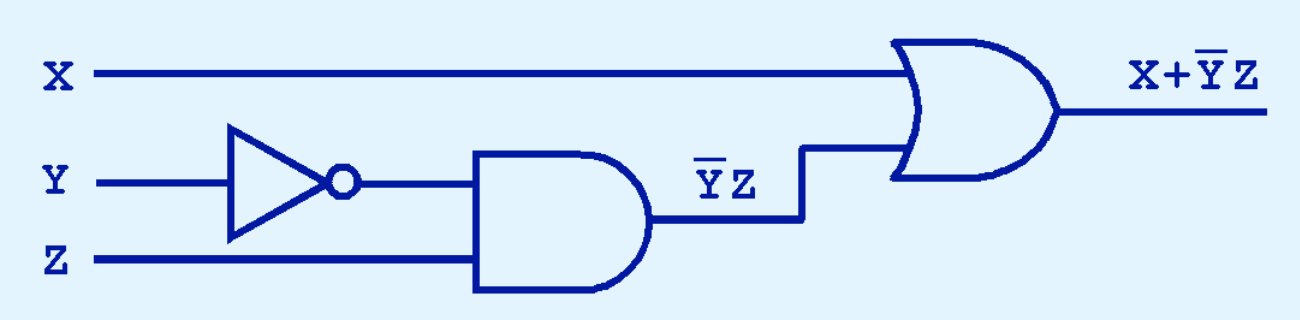
- Portat mund të kenë shumë hyrje dhe më shumë se një dalje
  - Dalja e dytë mund të shërbejë për komplementin e veprimit logjik
  - Këtë do ta shohim më vonë..



Komponentet dixhitale

- Kombinimi i portave është në funksion të implementimit të funksioneve logjike
- Çarku i mëposhtëm implementon funksionin logjik:

$$F(X,Y,Z) = X+\overline{Y}Z$$



Do t'i thjeshtësojmë shprehjet tona logjike, në mënyrë që t'i implementojmë më lehtë

Çarqet e kombinuara

- Kemi disenjuar çarkun që implementon funksionin logjik:

$$F(X,Y,Z) = X + \overline{Y}Z$$

- Ky çark është shembull i çarkut logjik të kombinuar
- Çarqet e kombinuara prodhojnë vlerë specifike dalëse pothuajse në momentin kur aplikohen vlerat hyrëse
  - Do të shohim edhe shembuj kur kjo nuk është e vërtetë...

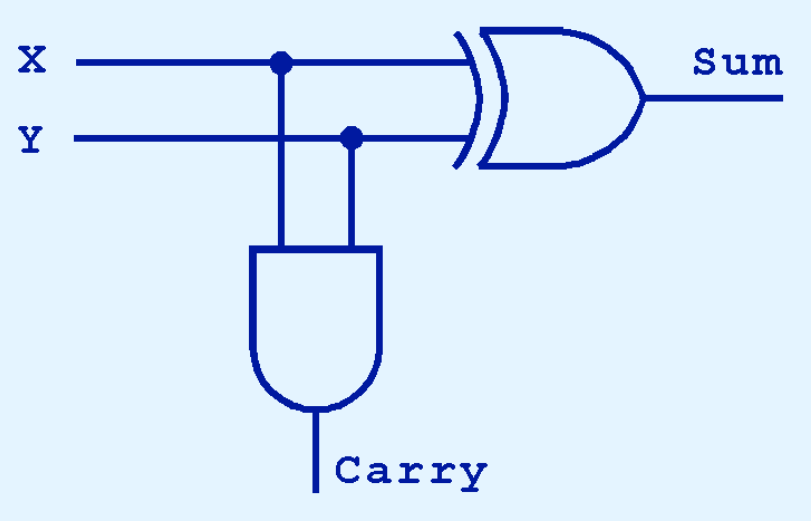
Çarqet e kombinuara

- Ka shumë pajisje të dobishme që punojnë me çarqe të kombinuara
- Një pajisje e tillë është gjysmë-mbledhësi (half adder), i cili gjen shumën e dy bitëve
- Djathtas është tabela e gjysmë-mbledhësit, prej të cilës mund të fitojmë ide për konstruktimin e këtij çarku

Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Çarqet e kombinuara

- Shuma mund të gjendet duke përdorur portën XOR, ndërsa bartja mund të bëhet me anë të portës AND



Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

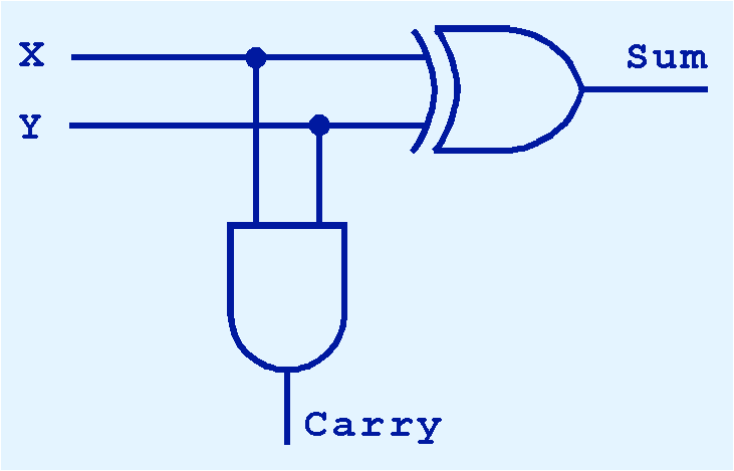
Çarqet e kombinuara

- Gjysmë-mbledhësi mund të shndërrohet në mbledhës të plotë (full adder), duke i përfshirë portat që përpunojnë bitin e bartur
- Ja tabela e saktësisë për mbledhësin e plotë.

Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Çarqet e kombinuara

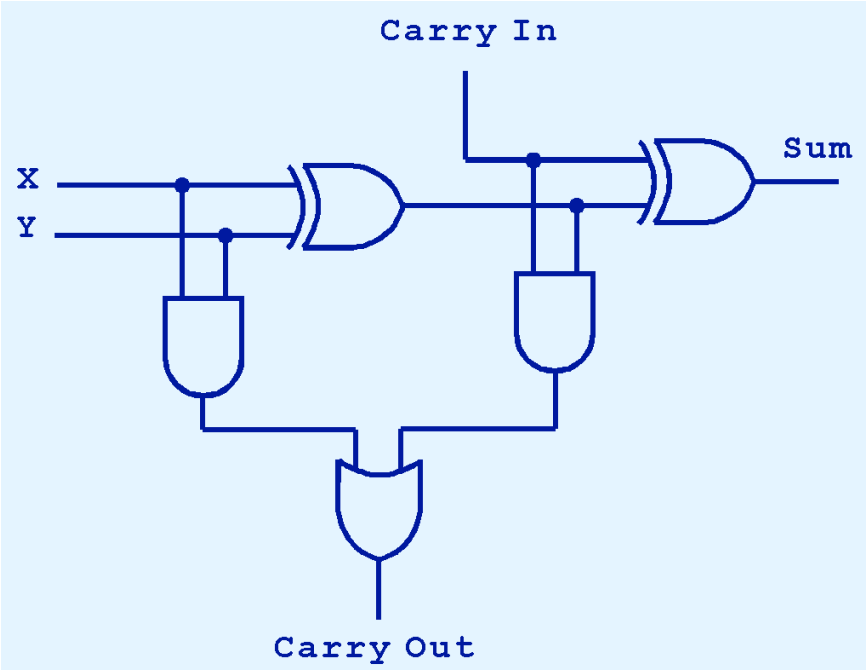
- Si mund ta shndërrojmë gjysmë-mbledhësin në mbledhës të plotë?



Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Çarqet e kombinuara

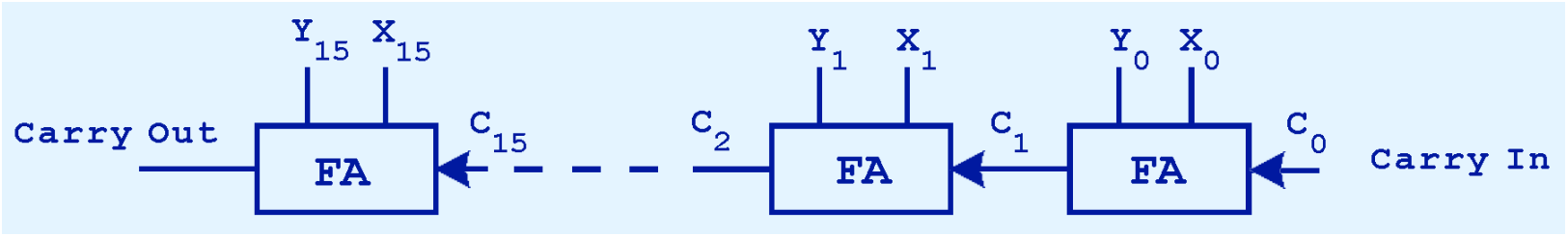
- Kështu:



Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Çarqet e kombinuara

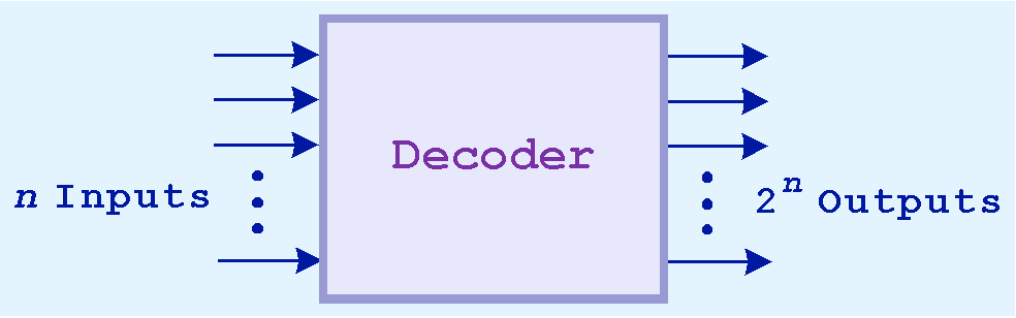
- Ashtu si i kombinuam gjysmë-mbledhësit për ta bërë mbledhësin e plotë, këta të fundit mund të lidhen në vargje
- Shembull i mbledhësit që i bartë bitet prej një mbledhësi të plotë në tjetrin është "ripple-carry adder"



Sistemet e sotme përdorin mbledhës më efektivë

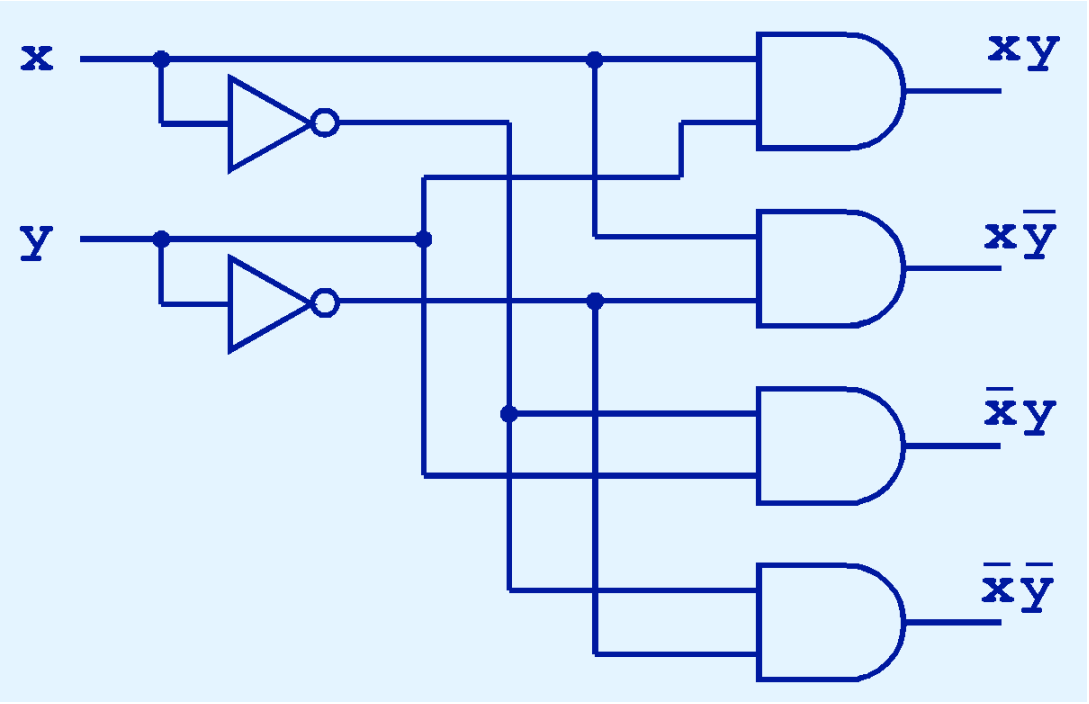
Çarqet e kombinuara

- Dekoduesit janë një tip i rëndësishëm i çarqeve të kombinuara
- Përveç tjerash, ata shërbejnë për përzgjedhjen e lokacionit të memories në pajtim me vlerën binare të vënë në linjat adresore të magjistrales së memories
- Dekoduesit e adresave me n hyrje mund të zgjedhin cilindo prej 2^n lokacioneve



Çarqet e kombinuara

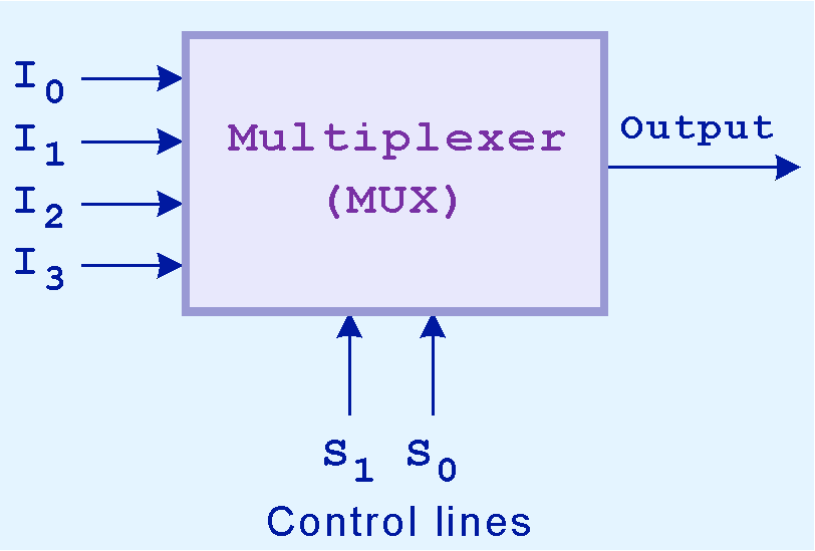
- Ja si duket dekoduesi 2-në-4



Nëse x = 0 dhe y = 1,
cila dalje është aktive?

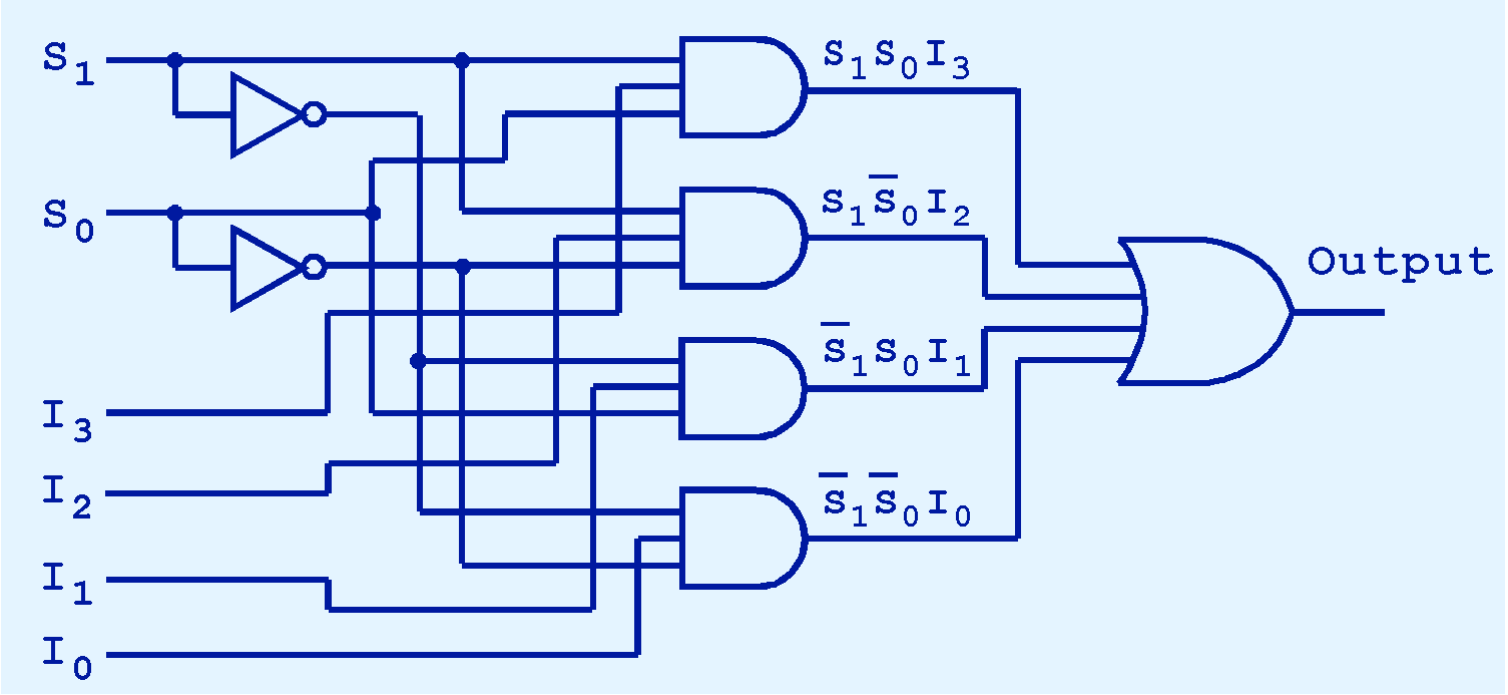
Çarqet e kombinuara

- Multiplekseri e bën të kundërtën e dekoduesit
- Ai zgjedh një dalje nga shumë hyrje
- Hyrja e zgjedhur për dalje përcaktohet nga vlera e linjave kontrolluese të multiplexerit
- Që të zgjedhim nga n hyrje, nevojiten $\log_2 n$ linja kontrolluese



Çarqet e kombinuara

- Multiplexeri 4-në-1



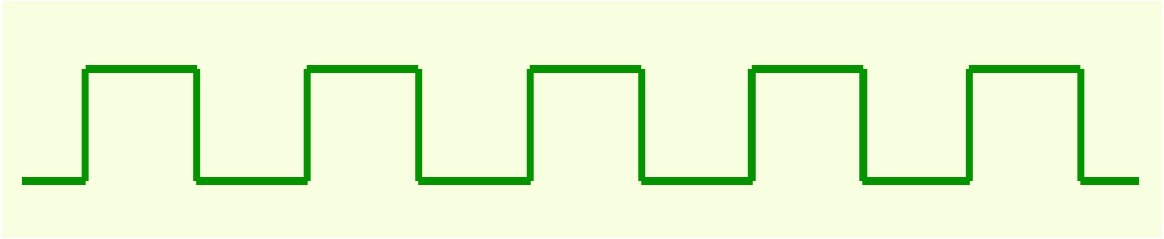
Nëse $S_0 = 1$ dhe $S_1 = 0$, cila hyrje transferohet në dalje?

Çarqet sekuenciale

- Çarqet e kombinuara janë ideale për situatat që kërkojnë aplikim të menjëhershëm të një funksioni logjik në një bashkësi të të dhënave hyrëse
- Sidoqoftë, ka raste të tjera kur nevojitet që çarku ta ndryshojë vlerën e tij në raport me gjendjen ekzistuese dhe të dhënat hyrëse.
  - Këto çarqe duhet ta "mbajnë në mend" gjendjen e vet aktuale.
- Çarqet logjike sekuenciale kanë mundësi të bëjnë diçka të tillë.

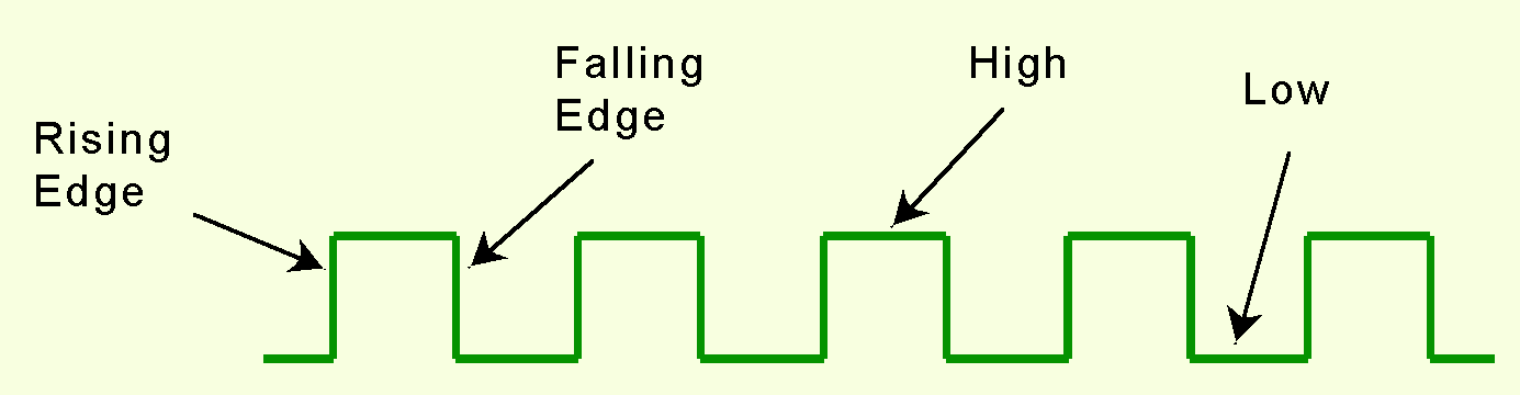
Çarqet sekuenciale

- Si tregon emërtimi, çarqet logjike sekuenciale kërkojnë një mënyrë për të krijuar një rënditje (sekuencë) të ngjarjeve.
- Ndryshimi i gjendjes kontrollohet me anë të orës.
  - „Ora" është një çark i veçantë që iu dërgon impulse elektrike çarqeve.
- Ora prodhon valë elektrike si kjo më poshtë.



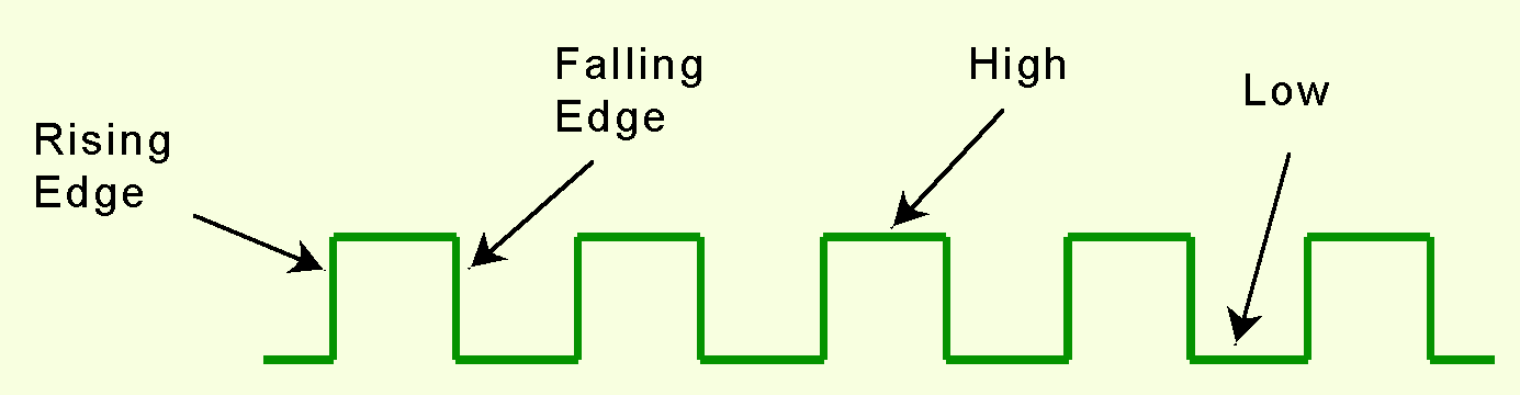
Çarqet sekuenciale

- Ndryshimi i gjendjes në çarqet sekuenciale ndodhë vetëm kur pulson ora
- Çarqet mund ta ndryshojnë gjendjen si në fazën e ngritjes ashtu edhe në fazën e rënies të pulsit të orës, ose kur ora arrin tensionin më të lartë.



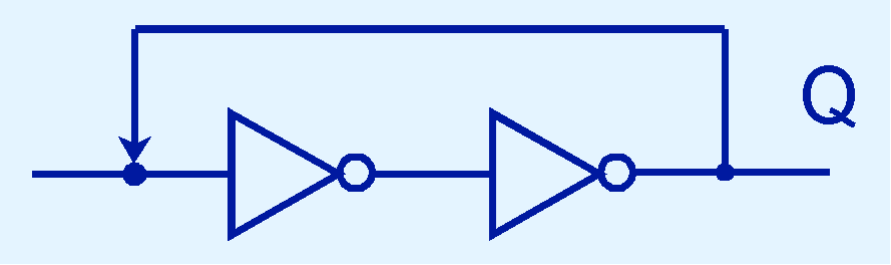
Çarqet sekuenciale

- Çarqet që ndryshojnë gjendjen në fazën e ngritjes ose në fazën e rënies të pulsit të orës quhen „edge-triggered".
- Çarqet „level-triggered" e ndërrojnë gjendjen kur pulsi i orës arrin tensionin më të lartë ose më të ulët.



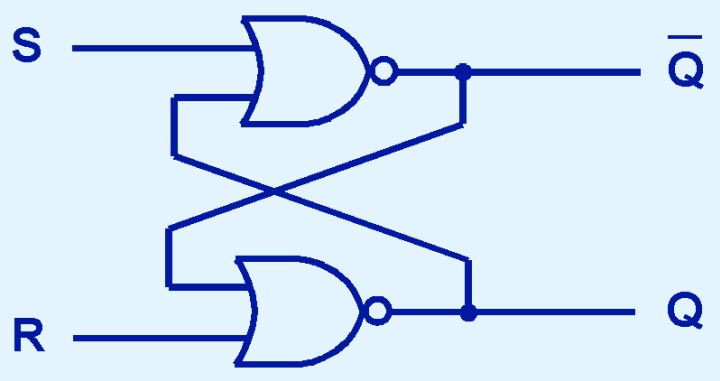
Çarqet sekuenciale

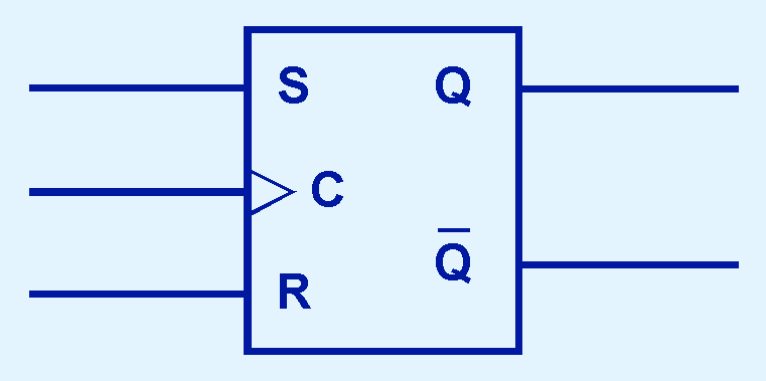
- Që t´i ruajnë vlerat e gjendjes, çarqet sekuenciale mbështeten në informatën kthyese (feedback).
- Tek çarqet dixhitale feedbacku paraqitet kur dalja kthehet prapa në hyrje.
- Më poshtë është dhënë një shembull
  - Nëse Q është 0, gjithnjë do të jetë 0; nëse është 1 gjithnjë do të jetë 1.
  - Pse?



Çarqet sekuenciale

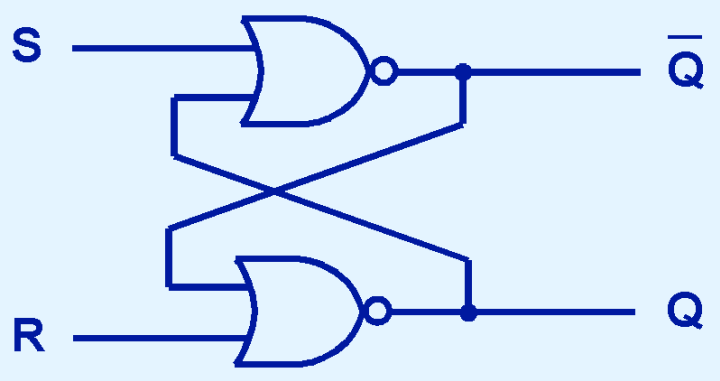
- Do të shohim se si funksionon feedbacku në flip-flopin „SR" (set-reset)





Çarqet sekuenciale

- Q(t) d.m.th. vlera e daljes në kohën t.
- Q(t+1) është vlera e Q pas pulsit të ardhshëm të orës.



S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

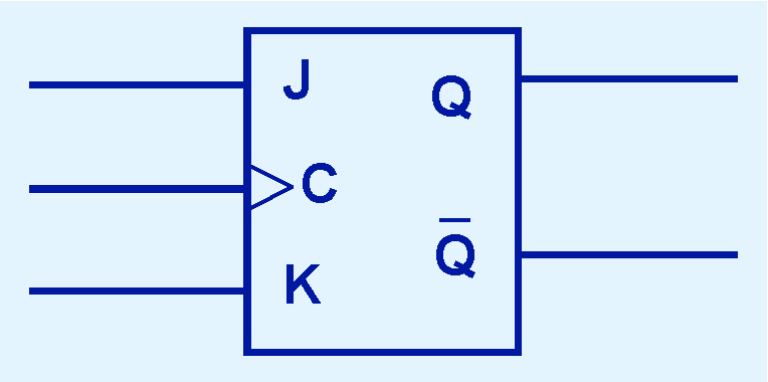
Çarqet sekuenciale

- SR i ka tri hyrje: S,R dhe daljen Q.
- Tabela e saktësisë është dhënë djathtas.
- Vëreni dy vlera të padefinuara. Kur S dhe R janë 1, SR është jostabil.

Present State			Next State
S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

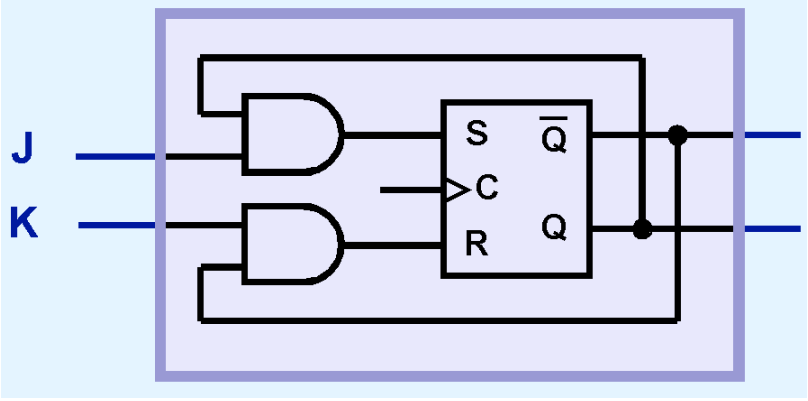
Çarqet sekuenciale

- Nëse sigurohemi që të dy hyrjet nuk do të jenë 1, atëherë flip-flopi SR do të jetë gjithnjë stabil
- SR mund të modifikohet që të jetë stabil edhe në rastin kur të dy hyrjet janë 1
- Flip-flopi i modifikuar quhet JK



Çarqet sekuenciale

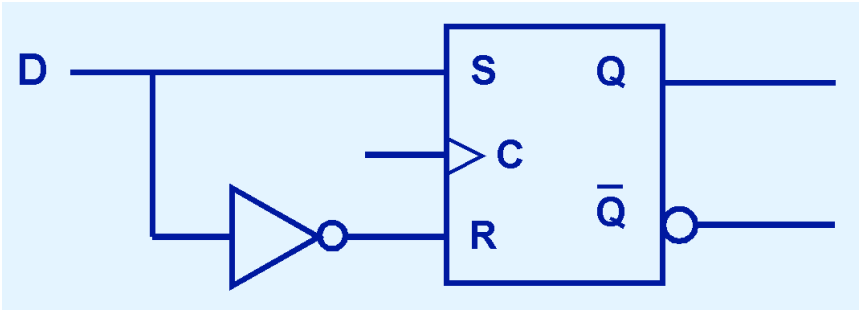
- Kështu modifikohet flip-flopi SR në JK
- JK është stabil për cdo vlerë të hyrjeve



J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}$(t)

Çarqet sekuenciale

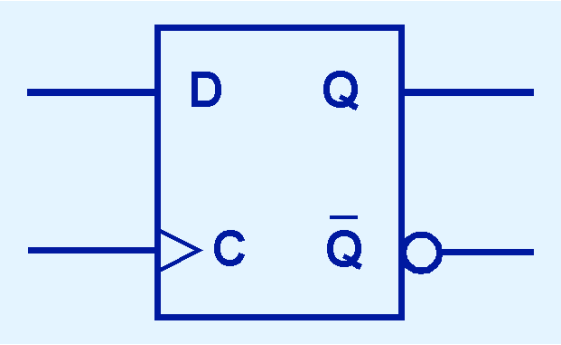
- Modifikim tjetër i flip-flopit SR është flip-flopi D
- Dalja e këtij flip-flopi mbetet e njëjtë gjatë pulsimeve të orës.
- Dalja ndryshon vetëm nëse ndryshon D.



D	Q(t+1)
0	0
1	1

Çarqet sekuenciale

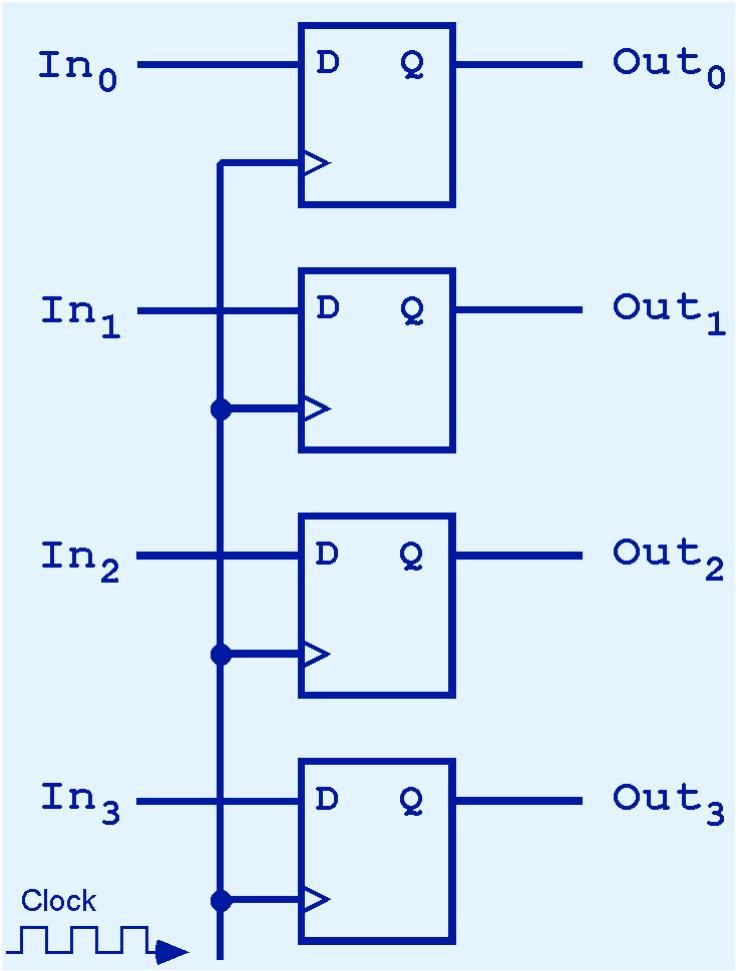
- Flip-flopi D është një çark fundamental në memorien e kompjuterit
- Të shohim se si kombinohen këto çarqe për të krijuar regjistra



D	Q(t+1)
0	0
1	1

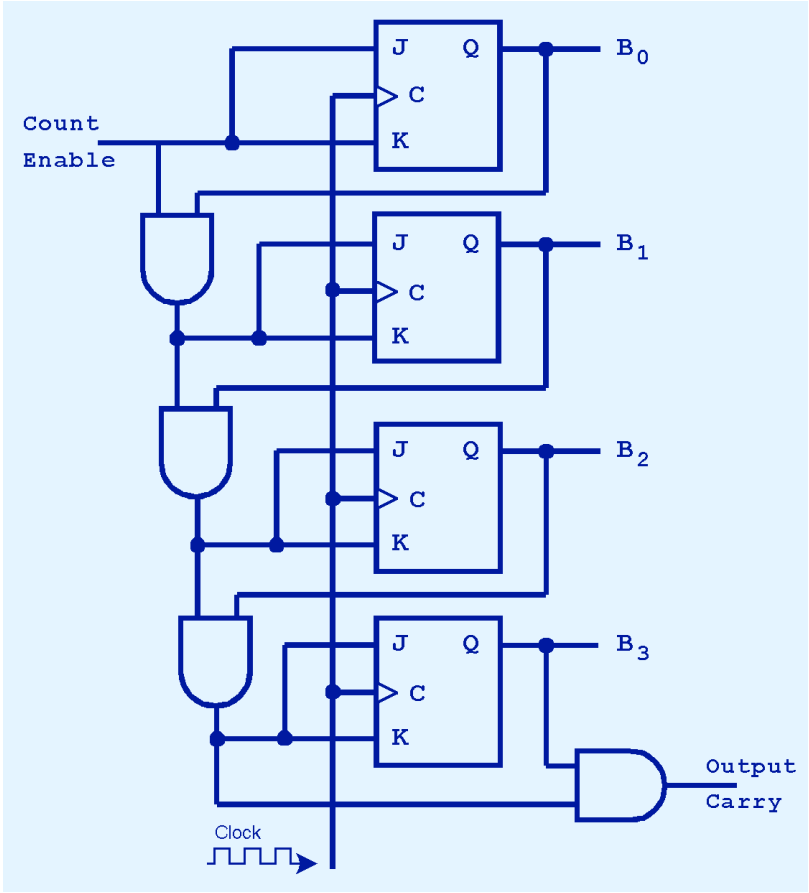
Çarqet sekuenciale

- Regjistri 4-bitësh i përbërë prej flip-flopeve D.



Çarqet sekuenciale

- Numëratori binar është shembull tjetër i çarkut sekuencial.
- Biti i nivelit më të ulët ndryshon në çdo pulsim të orës.
- Sa herë që ky bit ndryshon prej 0 në 1, biti i ardhshëm ndryshon dhe kjo vazhdon në flip-flopet tjera.



Dizanjimi i çarqeve

- Çarqet dixhitale i kemi shikuar nga dy aspekte: analiza dixhitale dhe sinteza dixhitale.
  - Analiza dixhitale hulumton raportin në mes të hyrjeve dhe daljeve të çarkut.
  - Sinteza dixhitale krijon diagrame logjike duke përdorur vlerat e specifikuara në tabelat e saktësisë.

Dizanjimi i çarqeve

- Dizanjuesit e çarqeve dixhitale mbështeten në softver të specializuar për të krijuar çarqe efektive
  - Pra, softveri mundëson konstruktimin e hardverit më të mirë
- Në realitet, softveri paraqet një bashkësi algoritmesh të cilat mund të implementohen në hardver
  - Kujtoni parimin e ekuivalencës së harduerit dhe softuerit

Dizanjimi i çarqeve

- Kur dëshirohet të implementohet një algoritëm i thjeshtë me shpejtësi të madhe ekzekutimi atëherë preferohet zgjidhja hardverike
- Kjo është idea e sistemeve të integruara (embedded systems), që janë kompjuterë të specializuar që i kemi në përdorim të përditshëm
- Këto sisteme programohen në mënyrë të vecantë, e cila kërkon të kuptuarit e mënyrës së funksionimit të çarqeve dixhitale

Përmbledhje

- Kompjuterët janë implementime të logjikës së Boole-it
- Funksionet logjike shpjegohen plotësisht me anë të tabelave të saktësisë
- Portat logjike janë çarqe të vogla që implementojnë veprime logjike
- Portat bazike janë AND, OR dhe NOT
  - Porta XOR është shumë e dobishme për verifikimin e paritetit dhe mbledhësit e ndryshëm
- Portat universale janë NOR dhe NAND
- Çarqet kompjuterike përbëhen prej çarqeve të kombinuara dhe çarqeve sekuenciale
- Çarqet e kombinuara prodhojnë dalje të re me të ndryshuar të vlerave hyrëse
- Ndryshimi i gjendjes në çarqet sekuenciale kontrollohet nga ora
- Njësia themelore e çarqeve sekuenciale është flip-flopi
  - Flip-flopet më karakteristikë janë SR,JK dhe D

Pyetje???