

DIGITÁLIS TECHNIKA I

Dr. Lovassy Rita
Dr. Pődör Bálint

Óbudai Egyetem KVK
Mikroelektronikai és Technológia Intézet

12. ELŐADÁS



1

KOMPARÁTOROK

2

KOMPARÁTOROK

A komparátorok összehasonlító áramkörök.
Két szám esetén háromféle összehasonlítás lehetséges

$$X < Y$$

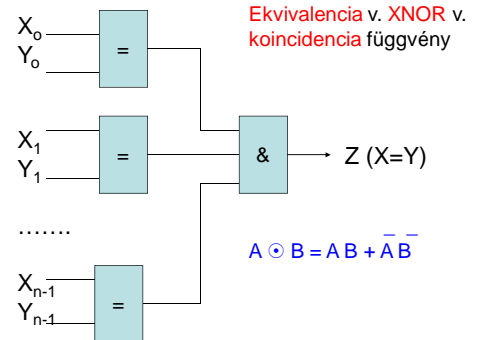
$$X = Y$$

$$X > Y$$

Két bináris szám azonos, ha minden bitjük megegyezik.
Komparátor kimenete 1 ha a két szám egyenlő, egyébként pedig 0.

3

N-HELYÉRTÉKES KOMPARÁTOR



4

AMPLITUDÓ-KOMPARÁTOR

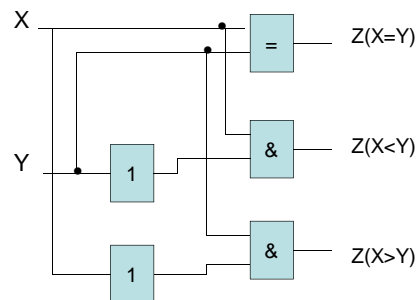
Eldönti, hogy a két (bináris) szám közül melyik a nagyobb.
1-bites számok esetén

X	Y	Z(X>Y)	Z(X=Y)	Z(X<Y)
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$Z(X>Y) = X\bar{Y} \quad Z(X=Y) = XY + \bar{X}\bar{Y} \quad Z(X<Y) = \bar{X}Y$$

5

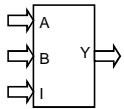
1-BITES AMPLITUDÓ KOMPARÁTOR



Az =, >, < relációt értékelő komparátor

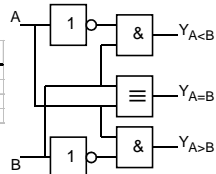
6

KOMPARÁTOROK



Egy bites komparátor

A	B	$Y_{A<B}$	$Y_{A=B}$	$Y_{A>B}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



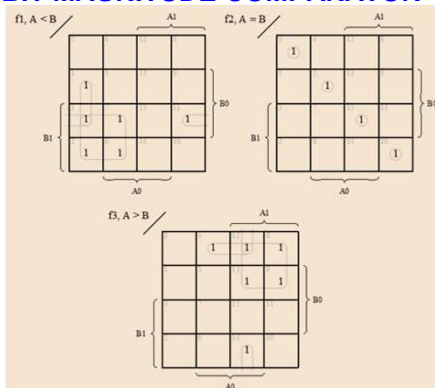
7

2-BIT MAGNITUDE COMPARATOR

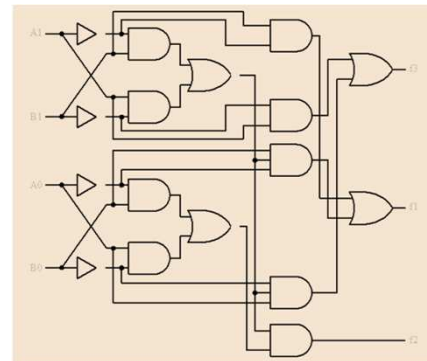
i	A1	A0	B1	B0	$f_1, A < B$	$f_2, A = B$	$f_3, A > B$
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	0	1
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

(b)

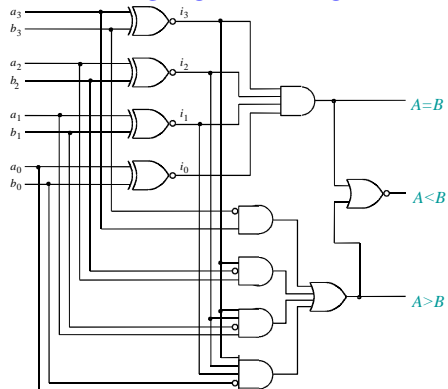
2-BIT MAGNITUDE COMPARATOR



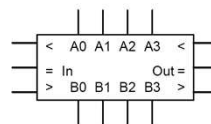
2-BIT MAGNITUDE COMPARATOR



4-BITES KOMPARÁTOR

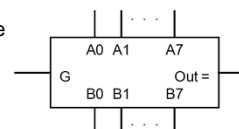


4- AND 8-BIT COMPARATORS: MSI

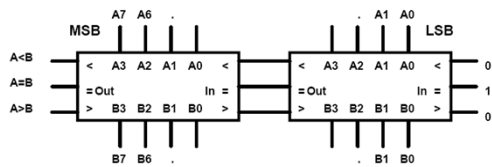


4-bit comparator three relation outputs ($A > B$, $A = B$, $A < B$) and three relation inputs for chaining purposes.

8-bit comparator with one relation ($A = B$) output, and one relation input for chaining purposes.



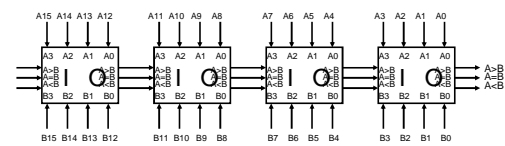
KOMPARÁTOROK KASZKÁDOSÍTÁSA



A legelső komparátor bemeneteit úgy kell beállítani mintha egy előző komparátor egyenlőséget jelezne.

13

KOMPARÁTOROK SOROS BŐVÍTÉSE



FUNCTION TABLES									
COMPARING INPUTS				GASICATING INPUTS			OUTPUTS		
A1,B1	A2,B2	A1,A1	A0,B0	A>B	A<B	A=B	A<B	A>B	A=B
A1-B1	X X	X	X	X	X	X	H	L	H
A1-B1	X X	X	X	X	X	X	L	H	L
A1-B1	A2-B2	X	X	X	X	X	H	L	L
A1-B1	A2-B2	X	X	X	X	X	L	H	L
A1-B2	A2-B1	X	X	X	X	X	H	L	L
A1-B2	A1-B1	X	X	X	X	X	L	H	L
A1-B2	A1-B2	A0-B0	A0-B0	X	X	X	H	L	L
A1-B2	A1-B2	A1-B1	A0-B0	X	X	X	L	H	L
A1-B2	A1-B1	A1-B2	A0-B0	X	X	X	H	L	L
A1-B2	A1-B1	A1-B2	A0-B0	X	X	X	L	H	L
A1-B2	A1-B1	A1-B2	A0-B0	X	X	X	H	L	L
A1-B2	A1-B1	A1-B2	A0-B0	X	X	X	L	H	L

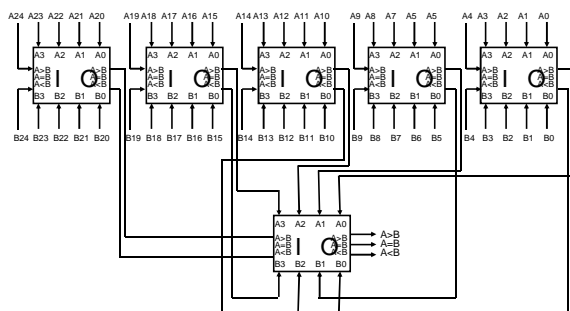
'85, '1585, '585									
A3-B3	A2-B2	A1-B1	A0-B0	X	X	H	L	L	H
A3-B3	A2-B2	A1-B1	A0-B0	H	H	L	L	L	L
A3-B3	A2-B2	A1-B1	A0-B0	L	L	L	H	H	L

LMS									
A3-B3	A2-B2	A1-B1	A0-B0	L	H	H	L	H	H
A3-B3	A2-B2	A1-B1	A0-B0	H	L	H	H	L	H
A3-B3	A2-B2	A1-B1	A0-B0	H	H	H	H	H	H
A3-B3	A2-B2	A1-B1	A0-B0	H	H	L	H	H	L
A3-B3	A2-B2	A1-B1	A0-B0	L	L	L	L	L	L

H = high level, L = low level, X = irrelevant

14

KOMPARÁTOROK PÁRHUZAMOS BŐVÍTÉSE



15

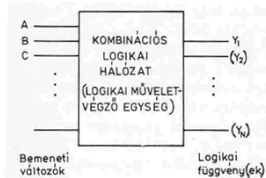
LOGIKAI HÁLÓZATOK

A logikai hálózatok két csoportra oszthatók:

1. Kombinációs logikai hálózatok
2. Sorrendi (szekvenciális) logikai hálózatok

16

KOMBINÁCIÓS LOGIKAI HÁLÓZAT



A legegyszerűbb logikai áramkörtípus a **kombinációs logikai hálózat**. Ez azonnal elvégzi a bemenetre jutó jeleken a "logikai műveletet", az eredmény azonnal (a belső működésből eredő késleltetési idő után) megjelenik a kimeneten.

17

SORRENDI LOGIKAI HÁLÓZATOK

A logikai áramkör kimeneti jele(i) a bemeneten fellépő jelkombinációkon kívül az előzőleg felvett állapotától is azaz az előzőleg kialakult kimeneti jelkombinációtól is függ.

Sorrendi vagy szekvenciális logikai hálózat.

Bemeneti változók:

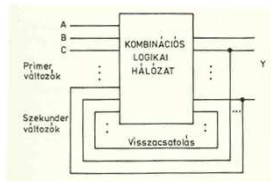
Visszacsatolt kimeneti változók:

primer változók.

szekunder változók.

18

SORRENDI LOGIKAI HÁLÓZAT VISSZACSATOLÁSSAL: ASZINKRON SORRENDI HÁLÓZAT



A kimeneteken lévő jelek visszacsatolás révén a bemenetre kerülnek (szekunder változók).
Aszinkron működés.

19

SORRENDI LOGIKAI HÁLÓZATOK TULAJDONSÁGAI

A sorrendi logikai hálózatok, a szekunder kombinációk révén képesek arra, hogy **ugyanazon bemeneti kombinációhoz más-más kimeneti kombinációt szolgáltatassanak** attól függően, hogy a bemeneti kombináció fellépte esetén milyen az éppen érvényes szekunder kombináció.

A szekunder kombináció pillanatnyi értékét pedig a logikai hálózat bemenetére jutott **korábbi** bemeneti kombinációk és azok sorrendje is befolyásolja, mivel a **szekunder kombinációk a működés során változnak**.

Innen ered a **sorrendi logikai hálózat** elnevezés.

20

SORRENDI HÁLÓZAT

A sorrendi hálózat, a kombinációs hálózattal szemben emlékezettel (memóriával) rendelkező hálózat.

A z_i kimeneti állapotot nemcsak a pillanatnyi x_i bemeneti állapot határozza meg, hanem a korábbi bemeneti állapotok, pontosabban a **bemeneti állapotok** (nem végtelen) **sorozata** azaz **szekvenciája**.

Ezért nevezik **szekvenciális hálózatnak**.

21

SORRENDI HÁLÓZAT

A bemeneti változók nem határozzák meg egyértelműen a kimeneti változók értékét, ezért ezeket újabb belső (szekunder) változókkal kell kiegészíteni.

A belső változók rögzítik (tárolják) a hálózat előző vezérlési állapotait, és a bemenő változókkal együtt egyértelműen meghatározzák a kimenő változókat.

22

SZINKRON ÉS ASZINKRON SORRENDI HÁLÓZATOK

A sorrendi hálózatok két csoportja:

1. Aszinkron, órajel nélkül működő hálózatok.
2. Szinkron, órajellel működő sorrendi hálózatok;

23

ELEMI SORRENDI HÁLÓZATOK (FLIP-FLOPOK)

24

ELŐÍRT TANKÖNYV-IRODALOM

Sorrendi hálózatok, flip-flopok, regiszterek, számlálók, stb.

Arató: Logikai rendszerek ..., 158-189. old.

Zsom: Digitális technika I, 318-345 old.

Rőmer: Digitális rendszerek ..., 98-116 old.

Rőmer: Digitális ... példatár, 30-36 old.

25

ELEMI SORRENDI HÁLÓZATOK

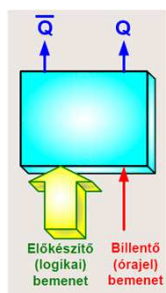
Kombinációs hálózatok: elemi kombinációs hálózatokból azaz kapukból építhetők fel.

Sorrendi (szinkron és aszinkron) hálózatok: szintén felépíthetők elemi sorrendi hálózatokból (is).

Elemi sorrendi hálózatok: önmagukban igen egyszerű logikai feladatok megoldására képesek csak, egy szekunder változójuk van. Tehát csak két állapotuk van, bemeneteik száma egy vagy kettő. Nevük **billenőkör**, **bistabil multivibrátor**, **tároló**, vagy **flip-flop**.

26

ELEMI SZINKRON TÁROLÓELEM (FLIP-FLOP) MŰKÖDÉSE

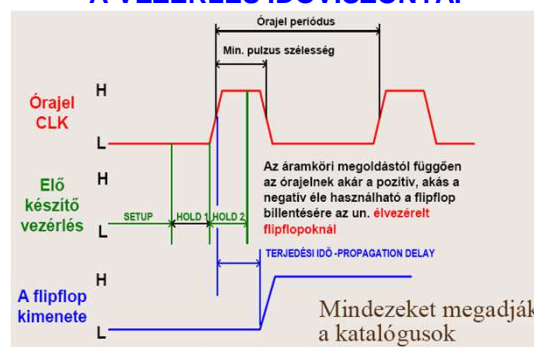


A logikai vezérlés hatása mindaddig nem érvényesül a kimeneten, amíg az órajel el nem indítja a flip-flop belső állapotváltozásait.

Ezen tranziens folyamat ideje alatt nem szabad a hálózat logikai vezérlését változtatni.

27

A VEZÉRLÉS IDŐVISZONYAI



Az órajel periódusideje hosszabb legyen mint a leghosszabb tranziens ideje.

28

FLIP-FLOPOK (TÁROLÓK)

Kétállapotú billenő elemek, **flip-flop**-ok (bistabil multivibrátor, billenőkör).

Leggyakrabban használt flip-flopok (logikai működés szerint):

S-R (vagy R-S) flip-flop	set-reset
J-K flip-flop	
T flip-flop	toggle
D flip-flop	delay, data
D-G flip-flop	gated D

Mindegyik **szinkron** módon működik, de az **S-R** és a **D-G** flip-flopok működhetnek **aszinkron** módon is.

29

FLIP-FLOPOK (TÁROLÓK) MŰKÖDÉSE

- Az aszinkron működésű tárolók állapotváltozása a bemenetre adott vezérlőjel hatására közvetlenül jön létre a késleltetési idő elteltével.

- A szinkron (órajellel vezérelt) flip-flopok állapotváltozása csak akkor jön létre, ha a szinkronizáló (óra, CLOCK) bemenetükre megérkezik az órajel.

30

FLIP-FLOPOK: STATIKUS ÉS DINAMIKUS VEZÉRLÉS

A FF-ok vezérlése kétféle lehet:

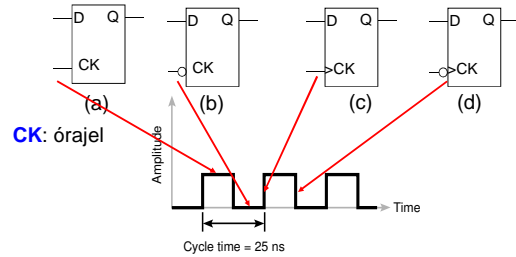
statikus vagy **dinamikus**.

- A **statikus** vezérlő bemenetekre a vezérlési táblázat szerint logikai 0 vagy logikai 1 **egyenszinteket** kell adni az állapotváltozás létrehozására.

- **Dinamikus** vezérlés: a FF billenése a dinamikus vezérlő bementre adott jel meghatározott irányú változásának hatására jön létre ("élre" billenő, **edge-triggered**).

31

TÁROLÓK ÉS FLIP-FLOPOK: JELÖLÉSEK



(a) **CK=1**, (b) **CK=0** szint esetén írja be **D**-t,

(c) **CK** emelkedő, (d) **CK** lefelé menő élénél.

Sokszor **S** (set, **PR** preset), **R** (reset, **CLR** clear) be- és **Q#** kimenet is van.

32

SR FLIP-FLOP: BEVEZETÉS

Az **SR** (set-reset) flip-flop a digitális rendszerekben használt egyik legegyszerűbb tároló, amely egy kombinációs hálózat direkt visszacsatolásával, azaz aszinkron sorrendi hálózattal valósítható meg.

- Két bemenet: **Set**, **Reset** és két kimenet
- Visszacsatolt kapcsolás
- Három megengedett és egy tiltott állapot
- A megengedett állapotok stabilak
- A tiltott állapot instabil lehet

33

SET- RESET (S-R) FLIP-FLOP (1)

Egyszerű igazságtábla

S	R	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	X

SET beírás, **RESET** törlés, függetlenül attól, mi volt az előző állapota.

Definiált működés:

S = 1 a FF állapotát 1-re állítja be, a vezérlés megszűnése után is 1-ben marad

R = 1 a FF állapotát 0-ra állítja be, és 0-ban is marad

Ha egyidejűleg **S** és **R** értéke 0 akkor az állapot nem változik (billenés nem történik), a flip-flop az **előző állapotát tárolja**, (állapotmegőrzés).

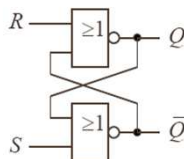
Ha **S** és **R** egyidejűleg 1 akkor a FF működése **definiálatlan**, tehát ez a vezérlési mód logikailag tiltott.

34

SR FLIP-FLOP S = R = 1 ESET

S = R = 1 esetén nincs definiálva a kimenet, ezért ez a bemeneti kombináció nem megengedett.

Ennek ellenére egy adott implementáció nyilván jól definiált értéket produkál a kimeneteken. Pl. a NOR alapú megoldás mindkét kimeneten 0-át, a NAND alapú megoldás 1-t, azonban mindkét esetben a két kimenet nem lesz egymás komplemente, mindkettő 1 illetve 0 lesz.



35

SR FLIP-FLOP (2)

n-edik (n+1)-edik ütem Összetett igazságtábla

S	R	Q^n	Q^{n+1}	
0	0	0	0	változatlan
0	0	1	1	változatlan
0	1	0	0	billen
0	1	1	0	változatlan
1	0	0	1	változatlan
1	0	1	1	billen
1	1	0	tiltott	
1	1	1	tiltott	

36

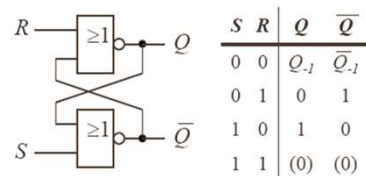
SR FLIP-FLOP: NOR KAPUS MEGVALÓSÍTÁSBAN

Az R "oldalán" van a Q !

1 bit tárolására alkalmas

R – RESET

S – SET



NOR kapu:

Bármelyik bemeneten 1 a kimenetet 0-ba kényszeríti

S	R	Q	Q̄
0	0	Q ₋₁	Q̄ ₋₁
0	1	0	1
1	0	1	0
1	1	(0)	(0)

37

INVERZ SR FLIP-FLOP

S R Qⁿ⁺¹

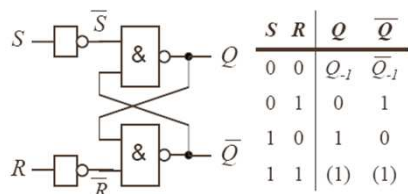
0	0	X
0	1	1
1	0	0
1	1	Q ⁿ

Az **inverz** SR flip-flopot a 0 szintek vezérik.
Megvalósítása két kereszt-csatolású **NAND kapu**.
A logikailag nem értelmezett vezérlésnél mindkét NAND kapu kimenet 1.

38

INVERZ SR TÁROLÓ

Az S "oldalán" van a Q !



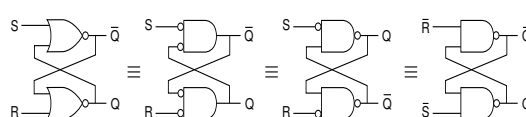
NAND kapu:

Bármelyik bemenetén 0 a kimenetet 1-be kényszeríti

S	R	Q	Q̄
0	0	Q ₋₁	Q̄ ₋₁
0	1	0	1
1	0	1	0
1	1	(1)	(1)

39

NOR S-R NAND S-R KONVERZIÓ



Active High
NOR Impl.

Push Bubbles
(DeMorgan's)

Rearrange
Bubbles

Convert
from Bubbles
to Active Low
Signal Names

NOR, aktív
H vezérlés

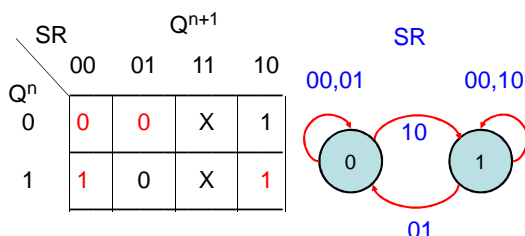
Átalakítás De
Morgan szerint

Negálások
áthelyezése

NAND, aktív
L vezérlés

40

SR FLIP-FLOP: ÁLLAPOTTÁBLA ÉS ÁLLAPOTGRÁF



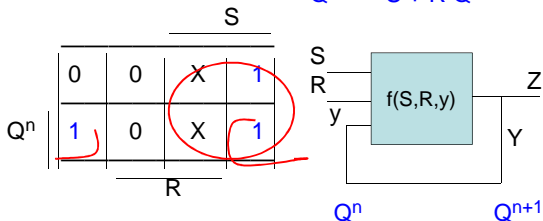
Egyetlen oszlopban sem alakul ki oszcilláció, így a működés mind szinkron, mind aszinkron módon értelmezhető, és ugyanaz. (Piros: stabil állapotok.)

41

ASZINKRON SR FF REALIZÁLÁSA KOMBINÁCIÓS HÁLÓZATTAL

Az állapottáblát Karnaugh táblázatnak tekintve

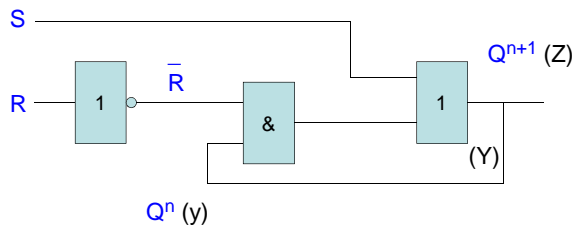
$$Q^{n+1} = S + \bar{R} Q^n$$



42

ÉS-VAGY KAPUS REALIZÁLÁS: ELVI LOGIKAI RAJZ

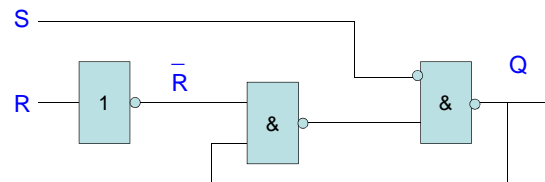
A $Q^{n+1} = S + \bar{R} Q^n$ karakterisztikus egyenlet szerint



43

ÉS-VAGY ÁTALAKÍTÁSA NAND-NAND KAPUS REALIZÁLÁSRA

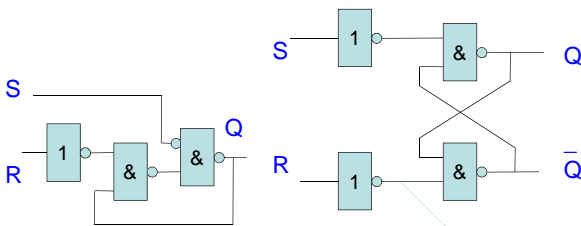
Átalakítás a De Morgan tétel szerint



44

ÉS-VAGY ÁTALAKÍTÁSA NAND-NAND KAPUS REALIZÁLÁSRA

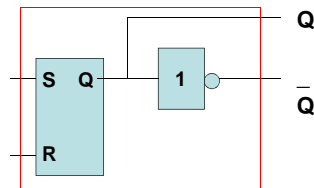
Átalakítás a De Morgan tétel szerint egyenlet szerint és ábrázolva



45

SR FLIP-FLOP

The problem of logically undefined state ($S = R = 1$) can be handled by using an additional inverter to generate \bar{Q} from Q (for NAND gate realization):



For the excitation $SR = 1$ the output will be $Q = 1$, i.e. the S input has the priority.

SR FLIP-FLOP: ÖSSZEFOGLALÓ

- Az SR flip-flop tipikusan **aszinkron** áramkör.
- Nem készítenek szinkron SR flip-flopot mert a tiltott vezérlési kombinációkat jobban is ki lehet használni.
- Azonban a szinkron flip-flopokba is beépítenek statikus (nem szinkronizált) nullázási (**RESET**) és beírási (**SET** illetve **PRESET**) lehetőséget a szinkron vezérlés mellett.
- A félvezető memóriák egyik alapeleme az SR flip-flop.

47

SR FLIP-FLOP: TOVÁBBLÉPÉS

A beírás nem egyértelmű
Nem egy lépésben történik
Az áramkörü műveleteknek lehet késleltetése

Megoldás

- Beírás órajelre: kapuzott SR tároló
- JK flip-flop készítése
- D flip-flop készítése

48