Bit: egy bináris számjegy, vagy olyan áramkör, amely egy bináris számjegy ábrázolására alkalmas.

Bájt (Byte): 8 bites egység, 8 bites szám.

Előjeles fixpontok számok:

2^8 = 256 különböző 8 bites szám lehetséges. Melyek jelentsenek negatív értékeket?

Előjeles számok szokásos ábrázolásai:

- előjeles abszolút érték,
- egyes komplemens,
- kettes komplemens,
- többletes.

Előjeles abszolút érték:

előjel és abszolút érték,

az első bit (balról) az előjel: 0: +, 1: -

Pl.: $+25_{10}$ = $0001\ 1001_2$, -25_{10} = $1001\ 1001_2$.

+ -

Jellemzők (8 bites szám esetén):

- a legkisebb szám -127, a legnagyobb 127,
- a nulla kétféleképpen ábrázolható:

  0000 0000 = 1000 0000

Egyes komplemens:

az első bit az előjel (0: pozitív, 1: negatív).

Egy szám -1-szerese (negáltja) úgy kapható meg, hogy a szám minden bitjét negáljuk (ellenkezőjére változtatjuk).

Pl.: $+25_{10}$ = $0001\ 1001_2$,

-25_{10} = $1110\ 0110_2$.

Jellemzők (8 bites szám esetén):

- a legkisebb szám -127, a legnagyobb 127,
- a nulla kétféleképpen ábrázolható:

  0000 0000 = 1111 1111

Kettes komplemens:

az első bit az előjel (0: pozitív, 1: negatív).

Egy szám negáltja úgy kapható meg, hogy az egyes komplemenséhez egyet hozzáadunk.

Pl.: $+25_{10}$ = $0001\ 1001_2$,

-25_{10} = $1110\ 0110_2$ egyes komplemens,

-25_{10} = $1110\ 0111_2$ kettes komplemens.

Jellemzők (8 bites szám esetén):

- a legkisebb szám -128, a legnagyobb 127,
- a nulla egyértelműen ábrázolható:

  0000 0000

Többletes: a szám és a többlet **összegét** ábrázoljuk előjel nélkül (ez már nem negatív!). **m** bites szám esetén a többlet általában $\mathbf{2^{m-1}}$ vagy $\mathbf{2^{m-1} - 1}$

Pl.: $+25_{10}$ = 10011001_2, 128-többletes ábrázolás

-25_{10} = 01100111_2 128-25=103

Jellemzők (128 többlet esetén):

- a legkisebb szám -128, a legnagyobb 127,
- a nulla egyértelműen ábrázolható.

Megjegyzés: a $\mathbf{2^{m-1}}$ többletes ábrázolás azonos a kettes komplemenssel fordított előjellel.

Használata: a lebegőpontos számok kitevő-részénél.

BCD (Binary Coded Decimal) ábrázolás: minden decimális számjegyet négy biten ábrázolunk.

Negatív számok BCD ábrázolása: 9 vagy 10 komplemens kóddal.

Pl.: $+301_{10}$ = 0000 0011 0000 0001,

-301_{10} = 1001 0110 1001 1000 (9 komplemens),

-301_{10} = 1001 0110 1001 1001 (10 komplemens).

Lebegőpontos számok

előjel karakterisztika törtrész

Sok ekvivalens ábrázolási lehetőség, leggyakrabban a törtrész első számjegye az adott szám első nullától különböző számjegye (normalizált alak).

Példa: 254 = $0{,}0254 \times 10^4$ = $\mathbf{0{,}254 \times 10^3}$ = $\mathbf{2{,}54 \times 10^2}$.

Megjegyzések:

- A nulla ábrázolásához külön megállapodásra van szükség (általában csupa nulla számjegyből áll).
- A lebegőpontos ábrázolásoknál is meghatározható a legkisebb és a legnagyobb ábrázolható szám, továbbá a legnagyobb hiba.

Feladat: Ábrázoljuk nyolcas számrendszerben 254_{10}-et előjelbittel kezdve, a kitevő legyen 1 jegyű (3 bites), 4-többletes, a törtrész 4 jegyű.

Megoldás: 254/8 | 6
31/8 | 7
3/8 | 3

376_8, normalizált alak: $0{,}376_8 \times 8^3$,

a megoldás: $0\ \underbrace{111}_{+}\ \underbrace{011\ 111\ 110\ 000}_{\text{törtrész}}{}_2 = 073760_8$

4-többletes kitevőrész

IEEE 754 standard

single 32 bites, double 64 bites (, extended 80 bites).

típus	előjel	kitevőrész	törtrész
single	**1 bit**	**8 bit** **127-többletes**	**23 bit**
double	**1 bit**	**11 bit** **1023-többletes**	**52 bit**

single: Ha 0 < a kitevőrész < 255, a szám normalizált.

Normalizált szám vezető 1-es bitje nincs ábrázolva!

Normalizált számok (IEEE 754, single)

0 < kitevőrész < 255

kitevőrész = kitevő + 127, **127 többletes**.

Lehetséges kitevők: -126, -125, ... , +127.

Közvetlenül a törtrész elé kell képzelni egy ***1***-est (**implicit bit**) és a bináris pontot.

Az ábrázolt szám: $\pm$ (*1* + törtrész) * $2^{\text{kitevő}}$

Pl.: 1 $0011\ 1111\ 1000 \ldots 0000_2 = 3F80\ 0000_{16}$

0,5 $0011\ 1111\ 0000 \ldots 0000_2 = 3F00\ 0000_{16}$

-1,5 $\underbrace{1011\ 1111}_{\pm\ \text{kitevő-rész}}\ \underbrace{1}_{1.}\ \underbrace{100 \ldots 0000}_{\text{tört-rész}}{}_2 = BFC0\ 0000_{16}$

Normalizálatlan számok (IEEE 754, single)

Ha a kitevőrész = 0

Ilyenkor a kitevő **-126** (! nem 127),

a bináris pontot implicit ***0*** előzi meg (nincs ábrázolva).

Az ábrázolt szám: $\pm$ (törtrész) * 2^{-126}

Pl.: $2^{-127} = 2^{-126} * 0{,}100 \ldots 0000_2 =$

$= \underbrace{0000\ 0000}\ \underbrace{0100 \ldots 0000}_2 = 0040\ 0000_{16}$

$\pm$ kitevőrész *0.* törtrész (2^{-1})

$-2^{-149} = -2^{-126} * 0{,}000 \ldots 0001_2 =$

$= \underbrace{1000\ 0000}\ \underbrace{0000 \ldots 0001}_2 = 8000\ 0001_{16}$

$\pm$ kitevőrész *0.* törtrész (2^{-23})

kitevő = -126

A legkisebb pozitív (single) normalizált szám:

$2^{-126} = 2^{-126} * 1{,}000 \ldots 0000_2 =$

$= \underbrace{0000\ 0000}\ \underbrace{1000 \ldots 0000}_2 = 0080\ 0000_{16}$

$\pm$ kitevőrész *1.* törtrész

A legnagyobb pozitív (single) normalizálatlan szám:

$2^{-126} * 0{,}111 \ldots 1111_2 =$

$= \underbrace{0000\ 0000}\ \underbrace{0111 \ldots 1111}_2 = 007F\ FFFF_{16}$

$\pm$ kitevőrész *0.* törtrész

$\approx 2^{-126}$

A különbségük csupán 2^{-149}.

Normalizálatlan számok (IEEE 754, single)

Ha a kitevőrész = 255

Túl nagy számok (túlcsordulás):

- ∞ (végtelen): pl. 1/0,
- **NaN** (Not a Number): pl. ∞ / ∞

Normalizát	±	0<kitevőrész<Max	bitminta
Nem normalizált	±	0	nem nulla bitminta
Nulla	±	0	0
Végtelen	±	111…1	0
Nem szám	±	111…1	nem nulla bitminta

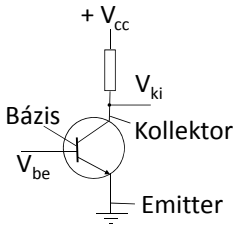
B.6. ábra. IEEE numerikus típusok

Digitális logikai szint

Digitális áramkör: két érték – általában
0-1 volt között az egyik (pl. 0, hamis),
2-5 volt között a másik (1, igaz).
Más feszültségeket nem engednek meg.

Kapu (gate): kétértékű jelek valamilyen függvényét tudja meghatározni.
Kapcsolási idő néhány ns (nanoszekundum = 10^{-9} s)

Tranzisztor



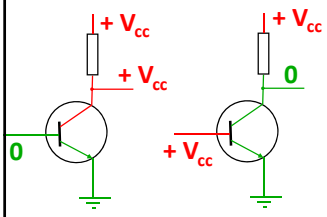
Ha V_{be} = föld, akkor a tranzisztor szigetel ⇒ $V_{ki} = V_{cc}$

Ha $V_{be} = V_{cc}$, akkor a tranzisztor vezet ⇒ V_{ki} = föld

Az igaz és hamis, az 1 és 0 kétféle reprezentációja:

- V_{cc} = igaz = 1, föld = hamis = 0 (pozitív logika),
- V_{cc} = hamis = 0, föld = igaz = 1 (negatív logika).

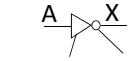
NEM (*NOT*) kapu, inverter (3.1-2. ábra, 3.2a.swf)



Szimbolikus jelölése:

Igazság tábla:

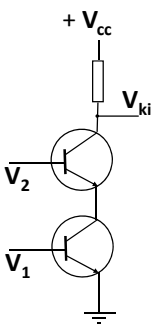
A	X
0	1
1	0



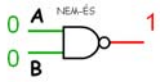
erősítő **Inverziós gömb**

(a)

NEM-ÉS (*NAND*) kapu (3.1-2. ábra, 3.2b.swf)



Szimbolikus jelölése:



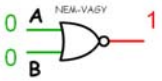
Igazság tábla:

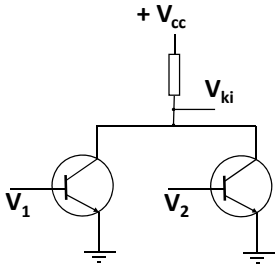
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

NEM-VAGY (*NOR*) kapu (3.1-2. ábra, 3.2c.swf)

Szimbolikus jelölése:





Igazság tábla:

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

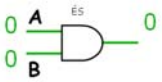
Visszalép

Előrelép

(c)

ÉS (*AND*) kapu (3.2. ábra, 3.2d.swf)

Szimbolikus jelölése:



Igazság tábla:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

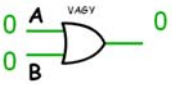
Visszalép

Előrelép

(d)

VAGY (*OR*) kapu (3.2. ábra, 3.2e.swf)

Szimbolikus jelölése:



Igazság tábla:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Visszalép

Előrelép

(e)

Boole-algebra

Olyan algebra, amelynek változói és függvényei csak a 0, 1 értéket veszik fel, a műveletei:

- ***ÉS*** (konjunkció),
- ***VAGY*** (diszjunkció),
- ***NEM*** (negáció).

Igazságtábla: olyan táblázat, amely a változók összes lehetséges értéke mellett megadja a függvény vagy kifejezés értékét.

Pl. 3 változós többségi függvény (**3.3. ábra**): értéke 1, ha legalább két argumentuma 1

Igazság tábla:

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Boole-algebrai alakja (diszjunktív normálforma)

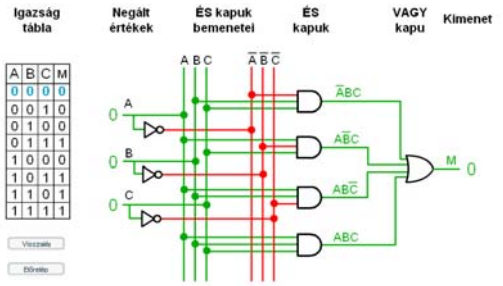
$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

A fölülvonás a ***NEM*** *(negáció)*, az egymás mellé írás az ***ÉS***, a + a ***VAGY*** művelet jele.

Boole-függvény megvalósításának lépései (3.3. ábra):

$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

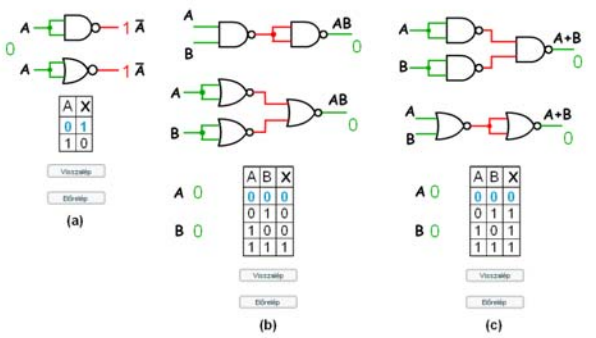
3.3.swf



3.3. ábra. Háromváltozós többségi függvény

***NAND* és *NOR* előnye:** teljesség (**3.4. ábra, 3.4.swf**)

3.4 ábra. a) *NEM*, b) *ÉS* c) *VAGY* kapuk csak NEM-ÉS vagy csak NEM-VAGY kapukkal történő megvalósítása



Definíció: Akkor mondjuk, hogy két Boole-függvény **ekvivalens**, ha az összes lehetséges bemenetre a két függvény azonos kimenetet ad.

Két Boole-függvény ekvivalenciája könnyen ellenőrizhető az igazság táblájuk alapján.

Pl.: AB + AC és A(B + C) ekvivalens (**3.5. ábra**).

Az első függvény megvalósításához két ***ÉS*** és egy ***VAGY*** kapura van szükség, a másodikhoz elegendő egy ***ÉS*** és egy ***VAGY*** kapu.

A Bool-algebra legfontosabb azonosságai (3.6. ábra)

Név	ÉS forma	VAGY forma
Identitás szabály	$1A = A$	$0 + A = A$
Nullszabály	$0A = 0$	$1 + A = 1$
Idempotens szabály	$AA = A$	$A + A = A$
Inverz szabály	$A\bar{A} = 0$	$A + \bar{A} = 1$
Kommutatív szabály	$AB = BA$	$A + B = B + A$
Asszociatív szabály	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Disztribucios szabály	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Abszorpciós szabály	$A(A + B) = A$	$A + AB = A$
De Morgan-szabály	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

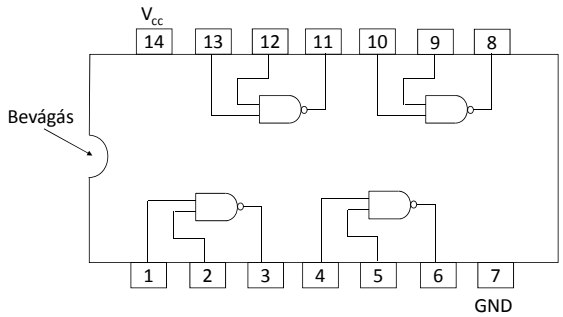
A	B	C	BC	A + BC	A + B	A + C	(A + B)(A + C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Disztribúciós szabály:

$$A+(BC)=(A+B)(A+C)$$

Jelölje az ***ÉS*** műveletet $\wedge$, a ***VAGY*** műveletet $\vee$, akkor

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$



3.10. ábra SSI lapka négy ***NAND*** kapuval

V_{cc}: feszültség, **GND**: föld.

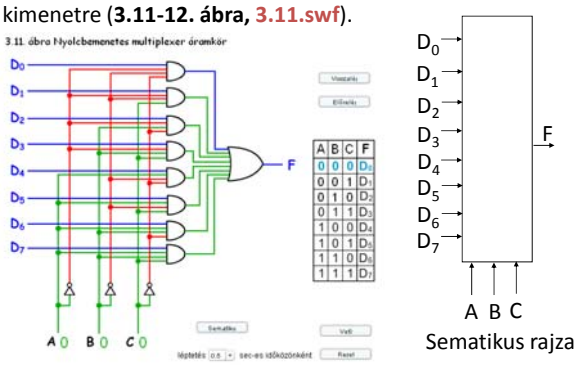
Kívánalom: sok kapu – kevés láb

<u>Kombinációs áramkörök</u>

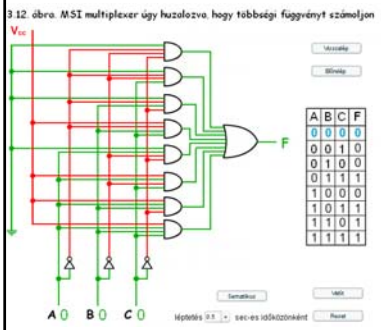
Definíció:

A pillanatnyi bemenetek egyértelműen meghatározzák a kimeneteket.

- **Multiplexer:** ***n*** vezérlő bemenet, ***2^n*** adatbemenet, ***1*** kimenet. Az egyik adatbemenet kapuzott (gated) a kimenetre (**3.11-12. ábra, 3.11.swf**).



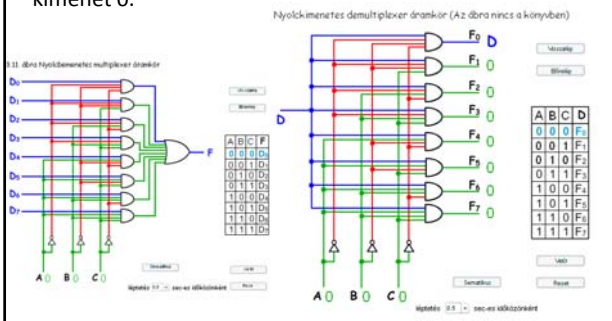
n vezérlő bemenetű multiplexerrel tetszés szerinti ***n*** változós Boole-függvény megvalósítható az adatbemenetek megfelelő választásával. Pl. a 3 változós többségi függvény:



3.12.swf

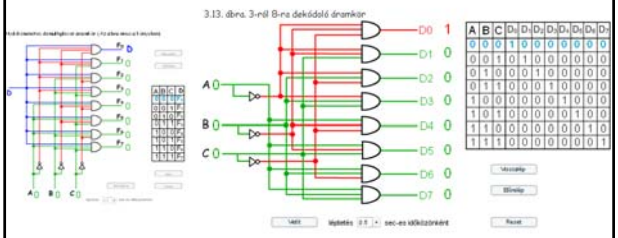
Párhuzamos-soros átalakítás: vezérlő vonalakon rendre: 000, 001, … 111.

- **Demultiplexer:** egy egyedi bemenetet irányít az ***n*** vezérlő bemenet értékétől függően a ***2^n*** kimenet egyikére, a többi kimenet 0.



multiplexer demultiplexer **dem.swf**

- **Dekódoló:** ***n*** bemenet, ***2^n*** kimenet. Pontosan egy kimeneten lesz 1, a többin 0 (**3.13. ábra, 3.13.swf**). Demultiplexerrel: a bemenetet igazra állítjuk.



demultiplexer dekódoló

- **Összehasonlító (comparator): (3.14. ábra, 3.14.swf)**

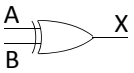
KIZÁRÓ VAGY

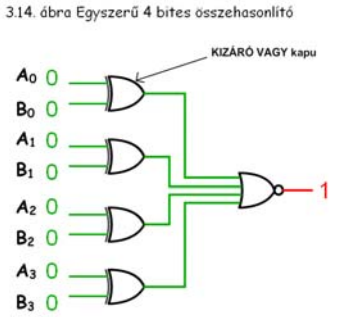
(***XOR eXclusive OR***) kapu

Igazság tábla:

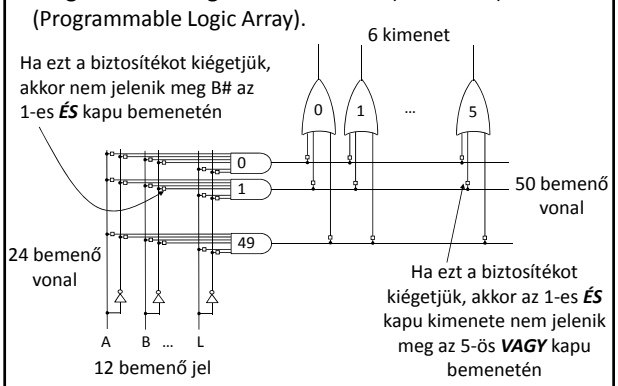
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Szimbolikus jelölése



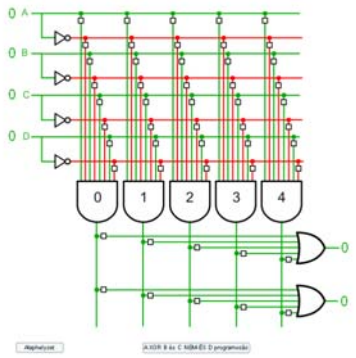


- **Programozható logikai tömbök: PLA (3.15. ábra)** (Programmable Logic Array).



3.15. ábra. 4 bemenetű, 2 kimenetű programozható logikai tömb

3.15.swf



Aritmetikai áramkörök

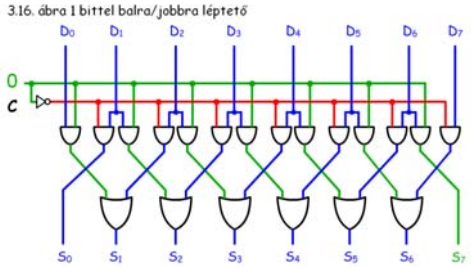
A kombinációs áramkörökön belül külön csoportot alkotnak.

Léptető:

C=1: jobbra,

C=0: balra léptet.

3.16. ábra 1 bittel balra/jobbra léptető

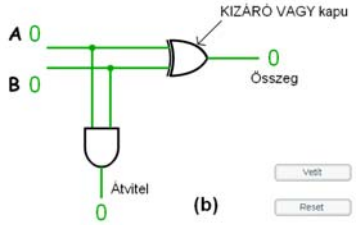


3.16.swf

Összeadók:

3.17. ábra (a) 1 bites összeadás igazságtáblája. (b) Fél összeadó áramkör

A	B	Összeg	Átvitel
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



(a) (b)

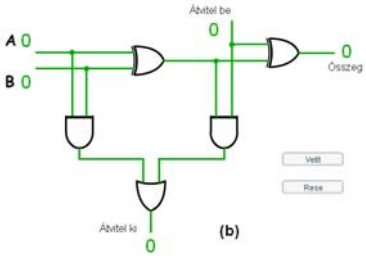
léptetés 0.5 sec-es időközönként

Fél-összeadó (half adder, **3.17. ábra,** 3.17.swf)

Összeadók:

3.18. ábra (a) A teljes összeadó igazságtáblázta (b) Teljes összeadó áramkör

A	B	Átvitel be	Összeg	Átvitel ki
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



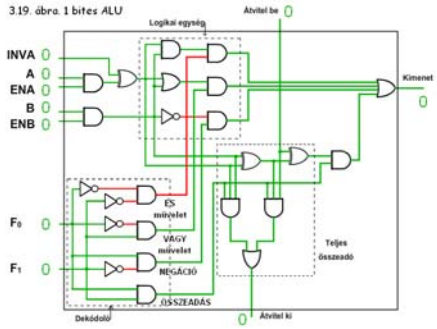
(a) (b)

léptetés 0.5 sec-es időközönként

Teljes-összeadó (full adder, **3.18. ábra,** 3.18.swf)

Aritmetikai-logikai egység: bitszelet (bit slice, **3.19. ábra**), **F0**, **F1** -től függően ***ÉS***, ***VAGY***, ***NEGÁCIÓ*** vagy +

3.19. ábra. 1 bites ALU



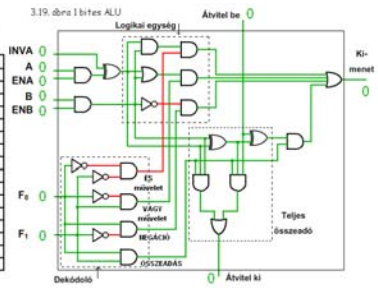
Csak fölismerni

3.19.swf

- átvitel továbbterjesztő összeadó (ripple carry adder):

4.2. ábra. Az ALU jeleinek hasznos kombinációi és az elvégzett tevékenység

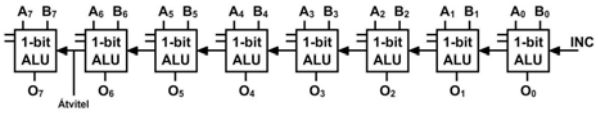
F0	F1	ENA	ENB	INVA	INC	Tevékenység
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	A
1	0	1	1	0	0	B
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	1	1	0	1	1	-A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1
0	1	0	0	1	0	-1



Csak a „Tevékenység” oszlop ismerete szükséges.

3.19b.swf

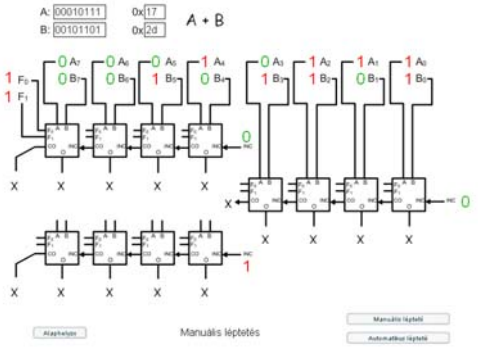
- átvitel továbbterjesztő összeadó (ripple carry adder):



Egy helyértéken csak akkor helyes az eredmény, ha az alacsonyabb helyértékeken már kialakult az átvitel értéke.

Minél több bites az ALU, annál tovább tart az összeadás elvégzése.

- átvitel kiválasztó összeadó (carry select adder) eljárás:



3.19c.swf

Feladatok

Mi a bit?

Mi a bájt (byte)?

Hány különböző pozitív egész szám fejezhető ki *k* számjegy segítségével *r* alapszám esetén?

Alakítsa át binárissá, oktálissá, hexadecimálissá a következő decimális számokat: 371, 4095, 255

Alakítsa át decimálissá a következő számokat:

$1111\ 1111_2$, 377_8, 0FFH (=$0FF_{16}$), 0C7EDH

Milyen fixpontos előjeles számábrázolásokat ismer? Jellemezze ezeket!

Feladatok

Hajtsuk végre a következő számításokat 8 bites kettes komplemensű számokon:

00101101 + 01101111,
11111111 + 11111111,
00000000 – 11111111,
11110111 – 11110111.

Feladatok

Mit tud az IEEE 754 standardról?

Mi az „implicit 1”, „implicit 0”?

Melyik a legkisebb/legnagyobb pozitív normalizált/normalizálatlan egyszeres pontosságú szám?

Konvertáljuk a következő számokat IEEE egyszeres pontosságúba! Az eredményt 8 jegyű hexadecimális számként adjuk meg: 9, 5/32, -5/32, 6,125.

Konvertáljuk a következő egyszeres pontosságú IEEE lebegőpontos számokat decimális számokká: 42E48000H, 3F880000H, 00800000H, C7F00000H.

Feladatok

Mi az igazság tábla? Írja fel a ***NEM***, ***ÉS***, ***VAGY***, ***NAND***, ***NOR***, ***XOR*** művelet igazság tábláját!

Milyen kapukat ismer? Rajzolja föl ezeket! Írja föl az igazságtáblájukat!

Mi a Bool algebra?

Mit jelent a ***NAND*** és ***NOR*** művelet teljessége?

Írja fel a 3 változós többségi függvény igazság tábláját és Bool-algebrai alakját!

Mikor mondunk két Boole-függvényt equivalensnek?

Hogy valósítható meg egy Boole-függvény?

Feladatok

Mit jelent az áramköri ekvivalencia?

Írja föl a Boole-algebra legfontosabb azonosságait!

Írja föl a disztribúciós szabályt!

Írja föl a De Morgan szabályt!

Mit nevezünk kombinációs áramkörnek?

Sorolja föl a tanult kombinációs áramköröket!

Egy ***k*** vezérlő bemenetű multiplexer hány adat bemenetet tud vezérelni?

Hány vezérlő bemenetel rendelkező multiplexer szükséges 11 adat bemenet vezérléséhez?

Feladatok

Rajzoljon 2 vezérlő bemenetes multiplexert, demultiplexert, dekódolót! Hogy működik?

Mire használható a multiplexer, és hogyan?

Rajzoljon 2 bites összehasonlítót!

Hogy épül föl egy PLA? Mire használható, és hogyan?

Milyen aritmetikai áramköröket ismer?

Rajzoljon 4 bites léptetőt! Hogy működik?

Rajzoljon fél összeadót! Írja föl az igazságtábláját!

Mi indokolja a „fél összeadó” elnevezést?

Rajzoljon teljes összeadót! Írja föl az igazságtábláját!

Feladatok

Milyen részei vannak az **ALU**-nak?

Milyen be- és kimenetei vannak az 1 bites **ALU**-nak? Melyik mire szolgál?

Milyen műveletek végezhetők el az **ALU**-val?

Az **ALU** milyen vezérlő bemenetei esetén lesz **1** az eredmény?

Rajzoljon sematikusan egy 4 bites átvitel továbbterjesztő összeadót! Hogy működik?

Hogyan csökkenthető az összeadásnál az átvitelekből származó idő?

Rajzoljon sematikusan egy 4 bites átvitel kiválasztó összeadót! Hogy működik?

Az előadáshoz kapcsolódó

Fontosabb témák

Negatív számok ábrázolásai

Lebegőpontos számok. IEEE 754 szabvány

Digitális logikai szint. Kapuk. Boole algebra. Boole függvények megvalósítása, áramköri ekvivalenciája. A Boole algebra legfontosabb azonosságai, dualitás

Kombinációs áramkörök, multiplexer, demultiplexer, dekódoló, összehasonlító

Az előadáshoz kapcsolódó

Fontosabb témák

Programozható logikai tömbök

Aritmetikai áramkörök. Léptető, fél és teljes összeadó, ALU, az ALU-val végezhető műveletek, átvitel továbbterjesztő és kiválasztó összeadás