

Kíváncsi: sok kapu – kevés láb

Kombinációs áramkörök

Definíció:
A kimeneteket egyértelműen meghatározzák a pillanatnyi bemenetek.

Máté: Architektúrák 3. előadás 1

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

3.11. ábra Nyolcbemenetes multiplexer áramkör

Máté: Architektúrák 3. előadás 2

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

3.12. ábra. MSI multiplexer úgy huzalozva, hogy többségi függvényt számoljon

3.12.swf

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

Máté: Architektúrák 3. előadás 3

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

3.11. ábra Nyolcbemenetes multiplexer áramkör

3.12. ábra Nyolckimenetes demultiplexer áramkör (Az ábra nincs a könyvben)

Máté: Architektúrák 3. előadás 4

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

3.13. ábra. 3-nál B-n a dekádló áramkör

Máté: Architektúrák 3. előadás 5

- 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

Máté: Architektúrák 3. előadás 6

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

Ha ezt a biztosítékot kiégetjük, akkor nem jelenik meg B# az 1-es **ÉS** kapu bemenetén

Ha ezt a biztosítékot kiégetjük, akkor az 1-es **ÉS** kapu kimenete nem jelenik meg az 5-ös **VAGY** kapu bemenetén

24 bemenő vonal
12 bemenő jel
6 kimenet
50 bemenő vonal

Máté: Architektúrák 3. előadás 7

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

Máté: Architektúrák 3. előadás 8

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

S₀ S₁ S₂ S₃ S₄ S₅ S₆ S₇

Máté: Architektúrák 3. előadás 9

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)

Máté: Architektúrák 3. előadás 10

Összeadók:

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

A	B	Átvitel be	Összeg ki	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)

Máté: Architektúrák 3. előadás 11

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

3.19. ábra. 1 bites ALU

Máté: Architektúrák 3. előadás 12

• átvitel továbbterjesztő összeadó (ripple carry adder):

4.2. ábra. Az ALU jellemző hasznos kombinációjai és az elővezetett tevékenység

FO	Fi	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	\bar{A}
1	0	1	1	0	0	\bar{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 + \bar{B}
1	1	0	1	0	1	\bar{A} + B
1	1	0	1	1	1	B - A
1	1	0	1	1	0	B - \bar{A}
1	1	1	0	1	0	\bar{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
0	1	0	0	1	1	1
0	1	0	0	1	0	-1

3.19. ábra. 1 bites ALU

3.20. ábra. Nyolc 1 bites ALU-szelet összekapcsolása 8 bites ALU-vé

3.19b.swf

Máté: Architektúrák 3. előadás 13

• átvitel kiválasztó összeadó (carry select adder) eljárás:

A: 0x A + B
 B: 0x

3.19c.swf

Máté: Architektúrák 3. előadás 14

Nem kombinációs áramkörök

Óra (clock, 3.21. ábra): ciklusidő (cycle time).
 Pl.: 500 MHz - 2 nsec.
 Finomabb felbontás késleltetéssel.
 Aszimmetrikus óra.

Máté: Architektúrák 3. előadás 15

Memória: „Emlékszik” az utolsó beállításra.

Tároló: Szint vezérelt (level triggered).

SR tároló (Set Reset latch, 3.22. ábra, 3.22.swf).
 Stabil állapot: a két kimenet **0, 1** vagy **1, 0**.
 S (set), R (reset) bemenet. ($Q\# \equiv \bar{Q}$)

3.22. ábra NEM-VAGY tároló

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Máté: Architektúrák 3. előadás 16

3.23. ábra Időzített SR-tároló

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

3.24. ábra Időzített D-tároló

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Máté: Architektúrák 3. előadás 17

3.25. ábra (a) Pulzsgenerátor, (b) Az áramkör négy pontjának az idődiagramm

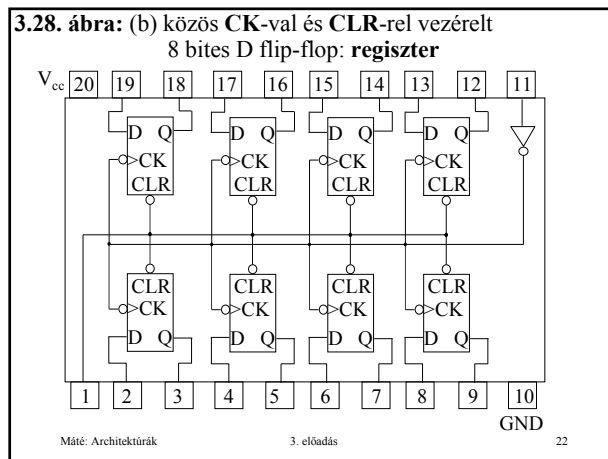
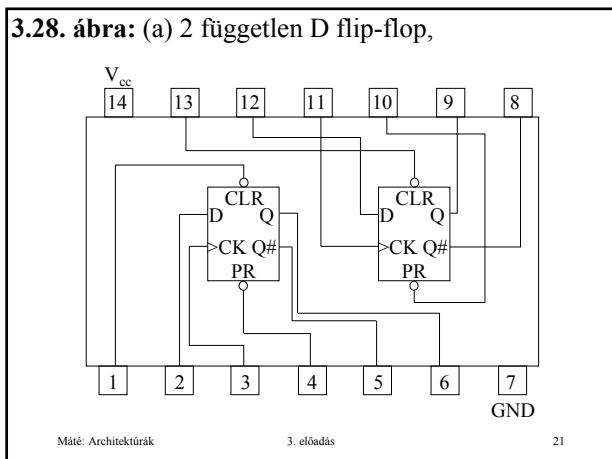
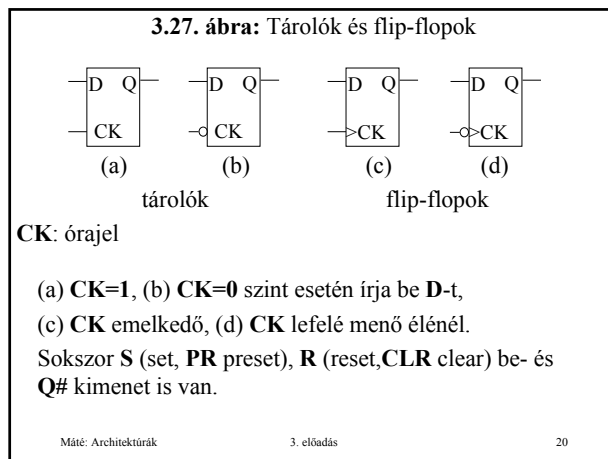
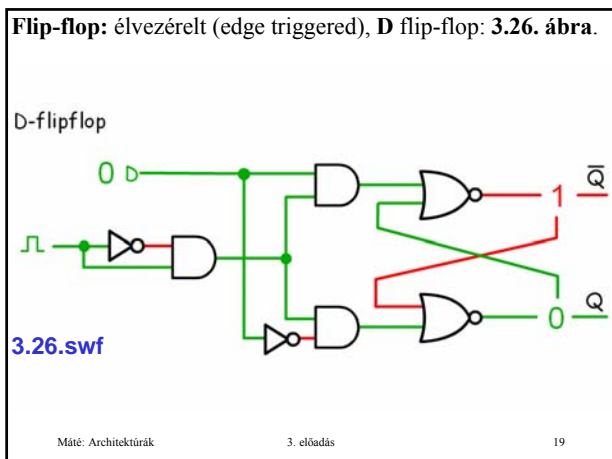
3.25.swf

(a) **d** —
b ÉS c —
c —
b —
a —

(b)

Az inverternek van egy kicsi (1-10 ns) késleltetése (Δ).

Máté: Architektúrák 3. előadás 18



Alapvető digitális logikai áramkörök

Integrált áramkör (IC, Integrated Circuit, chip, lapka)
5x5 mm² szilícium darab kerámia vagy műanyag lapon (tokban), lábakkal (pins). Négy alaptípus:

- SSI (Small Scale Integrated 1-10 kapu),
- MSI (Medium Scale ..., 10-100 kapu),
- LSI (Large Scale..., 100-100 000 kapu),
- VLSI (Very Large Scale ..., > 100 000 kapu).

Máté: Architektúrák 3. előadás 23

Memória szervezése

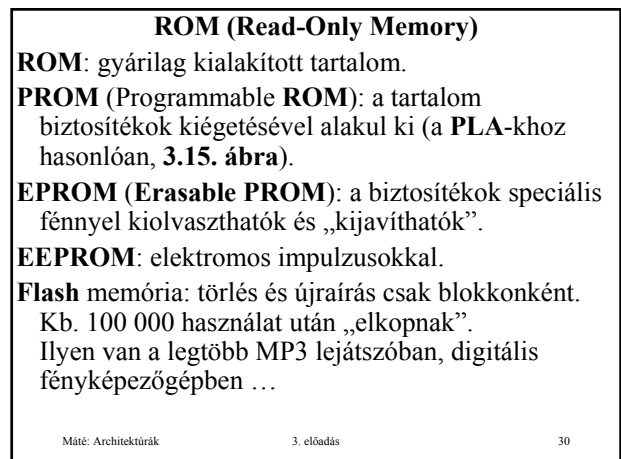
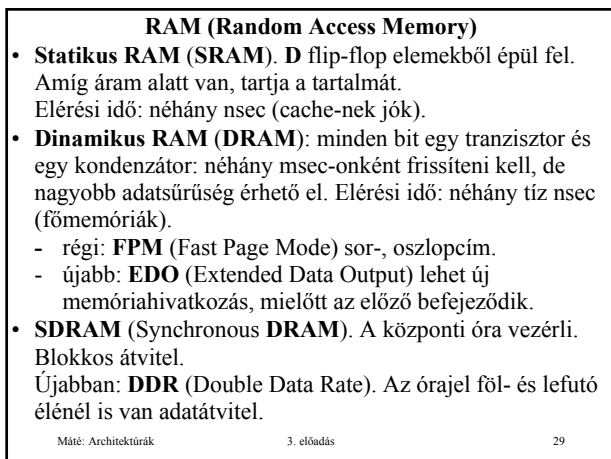
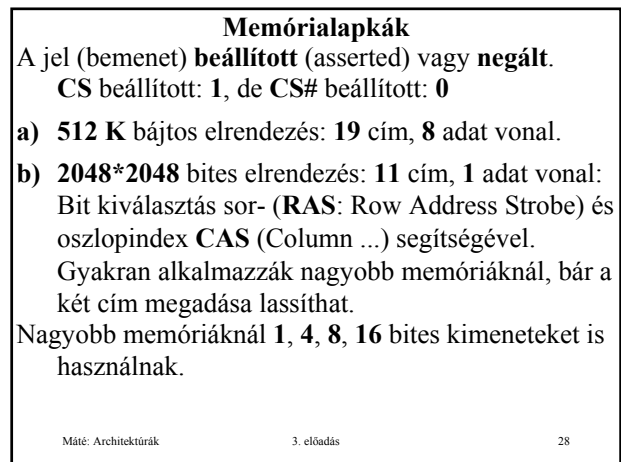
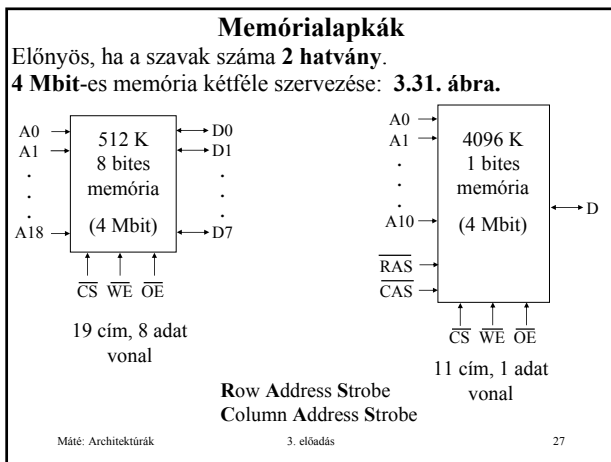
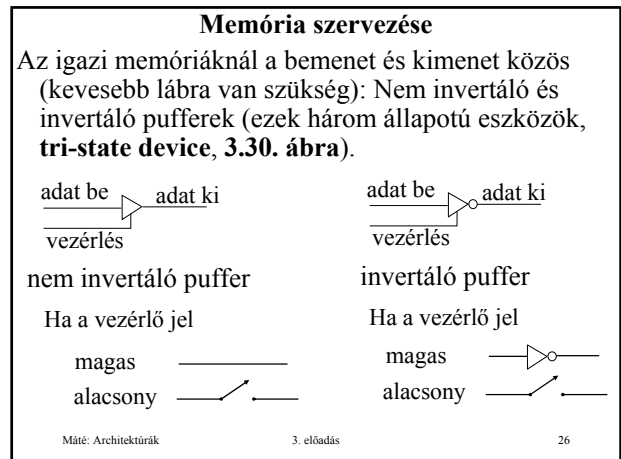
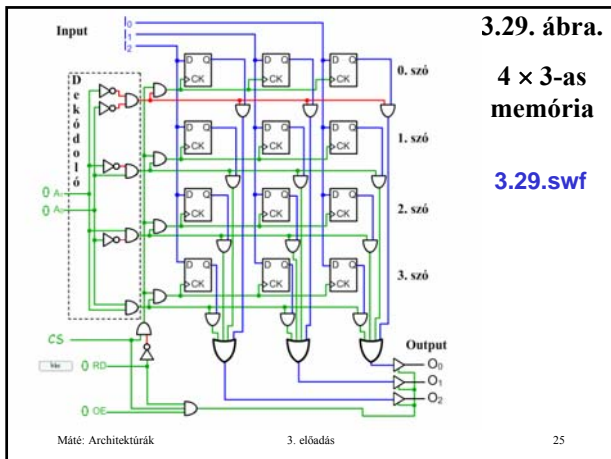
Elvárás: szavak címezhetősége.

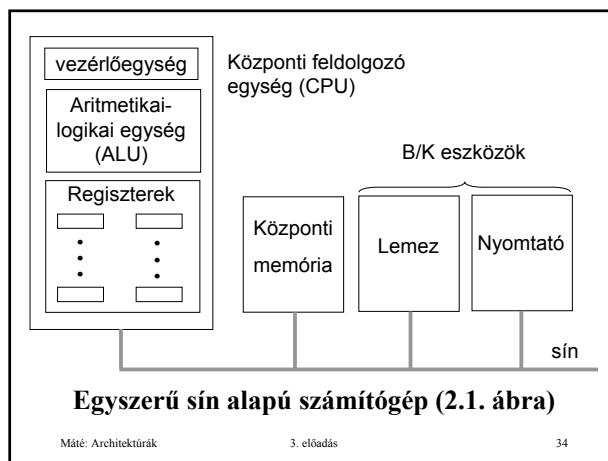
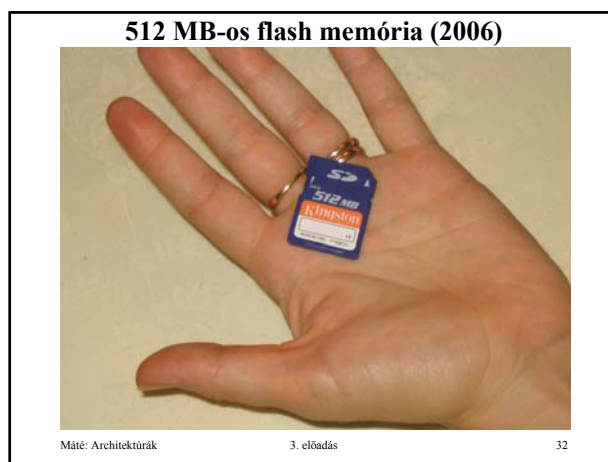
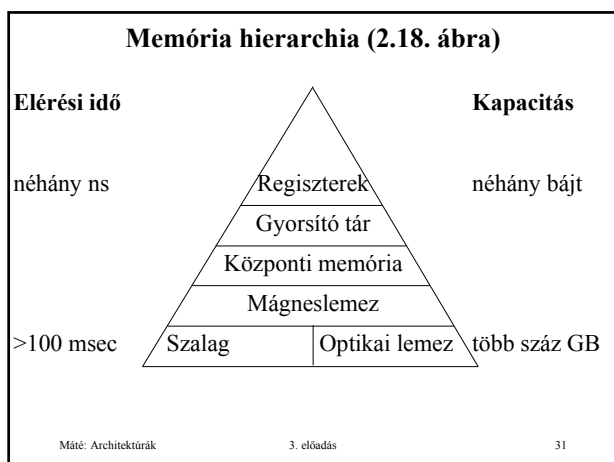
3.29. ábra: Négy db három bites szó. Bemenetek: három a vezérléshez,

- CS (Chip Select): lapka választás,
- RD (Read): 1: olvasás, 0: írás választása,
- OE (Output Enable): kimenet engedélyezése.

kettő a címzéshez (dekódoló),
három a bemenő adatoknak,
három adat kimenet.

Máté: Architektúrák 3. előadás 24





CPU feladata: a memóriában tárolt program végrehajtása.

Részei:

- vezérlőegység, feladata:
 - a program utasításainak beolvasása,
 - az ALU, a regiszterek vezérlése,
- aritmetikai-logikai egység (ALU), feladata: az utasítások végrehajtása,
- regiszter készlet, feladata: részeredmények, vezérlő információk tárolása. A legfontosabb regiszterek:
 - utasításslámláló (Program Counter): **PC**,
 - utasításregiszter (Instruction Register): **IR**,
- adatút (data path, 2.2. ábra).

Máté: Architektúrák 3. előadás 35

Adatút (data path, 2.2. ábra).

- A regiszter készletből feltöltődik az ALU két bemenő regisztere
- ALU**
- Az eredmény az ALU kimenő regiszterébe kerül
- Az **ALU** kimenő regiszteréből a kijelölt regiszterbe kerül az eredmény

Nem biztos, hogy az ALU be- és kimenő regiszterei tényleges regiszterként vannak kialakítva.

Máté: Architektúrák 3. előadás 36

CPU (Central Processing Unit) feladatai

- a végrehajtható utasítás betöltése,
- a betöltött utasítás típusának megállapítása,
- az ezt követő utasítás címének megállapítása,
- ha kell, az operandus(ok) helyének megállapítása,
- ha kell, az operandus(ok) betöltése,
- az utasítás végrehajtása,
- ha kell, az eredmény helyének megállapítása,
- ha kell, az eredmény tárolása,
- az egész ciklus újra kezdése.

Máté: Architektúrák

3. előadás

37

Feladatok

Mit nevezünk kombinációs áramkörnek?
Milyen kombinációs áramköröket ismer?
Milyen be- és kimenetei vannak a multiplexernek, a demultiplexernek, a dekódolóknak?
Mire használható a multiplexer, és hogyan?
Mire használható a **PLA**, és hogyan?
Milyen aritmetikai áramköröket ismer?
Hogy működik a léptető?
Hogy működik a „fél összeadó”?
Mi indokolja a „fél összeadó” elnevezést?

Máté: Architektúrák

3. előadás

38

Feladatok

Hogy épül fel a teljes összeadó?
Milyen részei vannak az **ALU**-nak?
Milyen be- és kimenetei vannak az 1 bites **ALU**-nak?
Milyen műveletek végezhetők el az **ALU**-val?
Milyen vezérlő bemenetek esetén lesz **1** az eredmény?
Milyen eredményt szolgáltat az **F₀=0, F₁=1, ENA=0, ENB=0, INVA=1, INC=1** vezérlő bemenet?
Hogy működik az átvitel továbbterjesztő/kiválasztó összeadó?

Máté: Architektúrák

3. előadás

39

Feladatok

Hogy érhetünk el az órajelnél finomabb időzítést?
Milyen nem kombinációs áramköröket ismer?
Kombinációs áramkör-e az **ALU**?
Hogyan csökkenthető az összeadásnál az átvitelekből származó idő?
Hány stabil állapota van az **SR** tárolónak?
Mi a különbség az **SR** és az időzített **SR** tároló között?
Mi a különbség az **SR** és **D** tároló között?
Mi a pulzusgenerátor, és mi a működési elve?
Mi a különbség a tároló és a flip-flop között?

Máté: Architektúrák

3. előadás

40

Feladatok

Hogy működik az invertáló és a nem invertáló puffer?
Miért használnak a memóriáknál invertáló vagy nem invertáló puffert?
Hogy címezhető meg **n** címlábon **2ⁿ**-nél nagyobb memória?
Mit jelent, hogy a **CS/CS#** bemenet beállított/negált?
Mi a **RAM**?
Milyen elemekből épül fel a **SRAM**?
Milyen elemekből épül fel a **DRAM**?
Hogy működik a **DRAM**?

Máté: Architektúrák

3. előadás

41

Feladatok

Hogy működik az **SDRAM**?
Mit jelent az **FPM** rövidítés?
Mit jelent az **EDO** rövidítés?
Hogy működik a **DDR**?
Mit jelent a **ROM** rövidítés?
Hogy működik az **EPROM**?
Hogy működik az **EEPROM**?
Milyen memória van a legtöbb fényképezőgépben?

Máté: Architektúrák

3. előadás

42

Feladatok

Mi a CPU?
Mi az ALU?
Mi az adatút?
Milyen részei vannak a CPU-nak?
Mi a regiszter?
Mit jelent az implicit operandus kifejezés?

Máté: Architektúrák

3. előadás

43

Az előadáshoz kapcsolódó

Fontosabb tételek

Programozható logikai tömbök
Aritmetikai áramkörök. Léptető, fél és teljes összeadó,
ALU, az ALU-val végezhető műveletek, átvitel
továbbterjesztő és kiválasztó összeadás
Nem kombinációs áramkörök.
Óra, tárolók, flip-flop-ok
A CPU részei, feladatai, adatút

Máté: Architektúrák

3. előadás

44