

Misák Sándor

SZÁMÍTÓGÉPES ARCHITEKTÚRÁK

Nanoelektronikai és
Nanotechnológiai Részleg

DE TTK

v.0.2 (2007.04.22.)

4. előadás

A DIGITÁLIS LOGIKA SZINTJE I.

A DIGITÁLIS LOGIKA SZINTJE I.

4. előadás

1. Kapuk és Boole-algebra:

- Kapuk;
- Boole-algebra;
- Boole-függvények megvalósítása;
- Áramköri ekvivalencia.

A DIGITÁLIS LOGIKA SZINTJE I.

4. előadás

2. Alapvető digitális logikai áramkörök:

- Integrált áramkörök (IC-k);
- Kombinációs áramkörök (multiplexerek (adatszelektorok), demultiplexerek, kódolók-dekódolók, összehasonlító, programozható logikai tömbök);
- Aritmetikai áramkörök (léptetők, összeadók, aritmetikai-logikai egységek);
- Órák.

A DIGITÁLIS LOGIKA SZINTJE I.

4. előadás

3. Memória:

- Memóriaparaméterek;
- Szekvenciális (sorrendi) áramkörök (tárolók, időzített tárolók, flip-flopok, regiszterek);
- Memóriaszervezés;
- Memórialapkák;
- RAM-ok és ROM-ok.

4. CPU lapkák.

KAPUK ÉS BOOLE- ALGEBRA

LOGIKAI KAPUK

Digitális áramkör egy olyan áramkör, amelyben csak **két logikai érték** van jelen.

Különböző feszültség-(ritkábban áram-) szintek jelentik ezeket a **logikai szinteket**.

Pl.: TTL (Transistor-Transistor Logic, tranzisztor-tranzisztor logika) áramköröknél pozitív logika esetén a logikai **0-nak a $0 \div 0,8$ V**, a logikai **1-nek a $2,4 \div 5$ V** feszültség szint felel meg.

LOGIKAI KAPUK

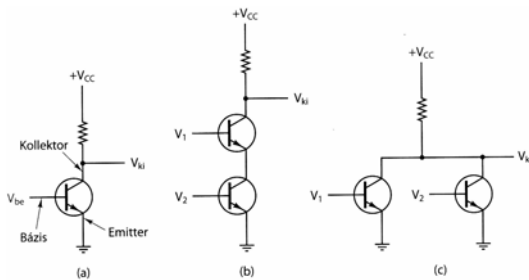
A kapuk **legkisebb elemei a tranzisztorok (bipoláris vagy térvezérlésű)**, melyeknek **három kapcsolata (kivezetése)** van a **külvilággal**.

Egy **bipoláris tranzisztornál** ezek a következők: **bázis, kollektor, emitter**.

Térvezérlésű tranzisztor esetén: **kapu, nyelő, forrás**.

Különböző feszültségekkel vezelve a tranzisztor bázisát, illetve kapuját, a tranzisztor **nyílt (vezeti az áramot)** vagy **zárt (nem vezeti az áramot)** állapotban lesz, ami **a két logikai szintnek** felel meg.

LOGIKAI KAPUK



- a) **Tranzisztoros fordító (inverter, NEM (NOT) kapu);**
 b) **NEM-ÉS (NAND) kapu;**
 c) **NEM-VAGY (NOR) kapu.**

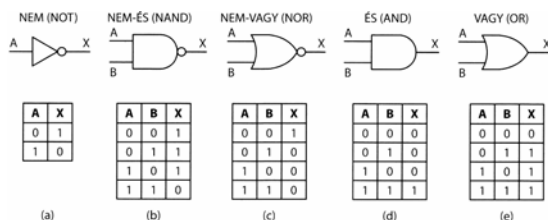
LOGIKAI KAPUK

Különböző módon kapcsolva össze a tranzisztorokat **különböző logikai művelet végző logikai áramkörök** készíthetők: a **legegyszerűbbektől a legbonyolultabbakig**.

A logikai kapuk ezek közül a legegyszerűbbek, mivel **csak a legegyszerűbb logikai műveletek elvégzésére** képesek:

- **logikai tagadás (invertálás, NEM-művelet);**
- **logikai összeadás (VAGY-művelet);**
- **logikai szorzás (ÉS-művelet);**
- **NEM-ÉS-, NEM-VAGY-, kizáró VAGY-műveletek.**

LOGIKAI KAPUK



Az öt alapkapu szimbóluma és működése

LOGIKAI KAPUK

A **kis köröket**, amelyeket a szimbólumok részeként használunk az inverter, a **NEM-ÉS, NEM-VAGY** kapuknál, **inverziós gömböknek (inversion bubbles)** nevezzük.

Eszközszinten a két főbb gyártási technológia: a bipoláris és a MOS (Metal-Oxide-Semiconductor) fém-oxid-félvezetős technológia.

A bipoláris technológia fő típusa:

- a **TTL**, amely évek óta az egyik főhordozója a digitális elektronikának;
- az **ECL (Emitter Coupled Logic) emitter-csatolt logika**, amelyet akkor használnak, ha nagy sebességű műveletek végrehajtására van szükség.

LOGIKAI KAPUK

A **MOS**-kapuk lassabbak, mint **TTL** és az **ECL**, de **kevesebb áramot igényelnek**, és **kevesebb helyet** foglalnak el, és ezért sokkal nagyobb számú elem rakható szorosan egymás mellé.

A **MOS**-nak nagyon sok változata van: **PMOS, NMOS, CMOS**.

Ha különböző technológiájú áramköröket össze akarjuk kapcsolni, gondoskodnunk kell az illető áramkörökről.

BOOLE-ALGEBRA

A kapuk kombinációjából felépíthető **áramkörök leírásához** egy új típusú algebra szükséges, amelynek változói csak a **0** és **1** értéket vehetik fel.

Egy ilyen algebrát **Boole-algebrának** (**switching algebra, kapcsolóalgebrának**) hívják, **George Boole** (1815-1864) angol matematikusról elnevezve.

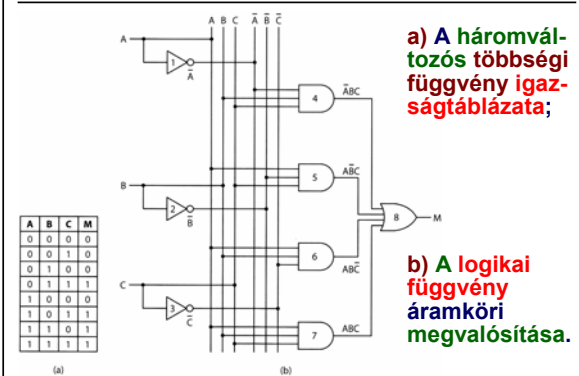
A **Boole-függvényeknek** egy vagy több bemeneti változója van, és egy eredményt szolgáltat, amely csak e változók értékétől függ.

BOOLE-ALGEBRA

Mivel az **n** változós **Boole-függvény** változóinak csak **2^n** lehetséges kombinációja van, a függvényt teljesen leírhatjuk egy **2^n** sorral rendelkező táblázattal, ahol egy-egy sor megmondja a bemeneti értékek adott kombinációja mellett a **függvényértékét**.

Ezt a táblázatot **igazságtáblázatnak** (truth table) nevezzük.

BOOLE-ALGEBRA



BOOLE-FÜGGVÉNYEK MEGVALÓSÍTÁSA

Egy Boole-függvény megvalósítása:

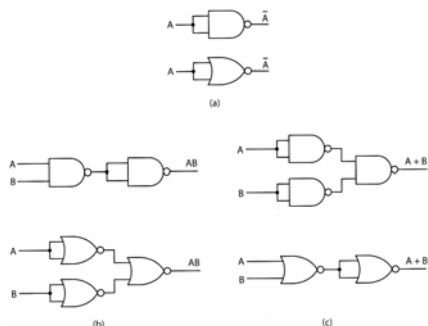
1. Írjuk fel a függvény igazságtáblázatát.
2. Biztosítsunk **NEM** kapukat minden bemenet komplementjének előállításához.
3. Rajzoljunk **ÉS** kaput minden sorhoz, amelynek az eredményoszlopában **1** van.
4. Kapcsoljuk össze az **ÉS** kapukat a megfelelő bemenetekkel.
5. Az összes **ÉS** kapu kimenetét tápláljuk be egy **VAGY** kapuba.

BOOLE-FÜGGVÉNYEK MEGVALÓSÍTÁSA

A **NEM-ÉS** és a **NEM-VAGY** kapukról azt mondjuk, hogy **teljesek (complete)**, mert bármely **Boole-függvény** kiszámítható ezek bármelyikének kizárólagos felhasználásával.

Más kapunak nincs meg ez a tulajdonsága, ami miatt **előnyben részesítik** ezeket az áramköri blokkok tervezésénél.

BOOLE-FÜGGVÉNYEK MEGVALÓSÍTÁSA



a) NEM, b) ÉS, c) VAGY kapuk csak NEM-ÉS és NEM-VAGY kapukkal történő megvalósítása.

ÁRAMKÖRI EKVIVALENCIA

A hálózattervezők gyakran próbálják csökkenteni termékeikben a **kapuk számát**, hogy **csökkentsék az alkatrészek árát**, a **nyomatott áramkört lap nagyságát**, az **áramfogyasztást** és így tovább.

Az áramkör bonyolultságának csökkentéséhez, a tervezőknek találni kell egy olyan áramkört, amely ugyanazt a függvényt (ún. **ekvivalens függvény**) számolja ki, mint az eredeti, de **kevesebb kapuból áll** (vagy **egyszerűbb kapukból**, pl. kétbemenetes kapukból a négybemenetesek helyett).

ÁRAMKÖRI EKVIVALENCIA

Az **ekvivalens áramkörök** keresésében a **Boole-algebra** nagyon értékes eszköz.

Általában az áramkör tervezők **Boole-függvénnyel** kezdenek, és aztán **alkalmazzák a Boole-algebra szabályait**, és próbálnak **egyszerűbb, de ekvivalens függvényt** találni.

A **végleges formából** azután létrehozzák az áramkört.

Ahhoz, hogy ezt a megközelítést használjuk, **szükségünk van a Boole-algebra néhány azonosságára.**

ÁRAMKÖRI EKVIVALENCIA

Név	ÉS forma	OR forma
Identitásszabály	$1A = A$	$0 + A = A$
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	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Disztribúciós szabály	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Abszorpció 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 **Boole-algebra** néhány azonossága

KAPUK ÉS BOOLE-ALGEBRA

INTEGRÁLT ÁRAMKÖRÖK

A kapukat nem egyedileg gyártják és árulják, hanem egységekben, ún. integrált áramkörökben, amelyeket **IC**-knek (**Integrated Circuits**) vagy **lapkának (chips)** hívnak.

Egy **IC kb. egy 5x5 mm-es négyzetes szilíciumdarab**, amelyen néhány kaput helyeznek el.

A kis **IC**-ket szokásosan egy derékszögű műanyag vagy kerámialapon (tokban) helyezik el, amely **5-15 mm széles és 20-50 mm hosszú**.

INTEGRÁLT ÁRAMKÖRÖK

A hosszú élek mentén **5 mm** hosszú láboknak **két párhuzamos sora** van, amely behelyezhető egy **foglalatba**, vagy **nyomtatott áramköri táblára** forrasztható.

Minden láb egy-egy kapunak a **bemenete** vagy **kimenete**, vagy **áram**, vagy pedig **föld bemenet**.

A kívül kétsoros lábazást és a belső integrált áramkört együtt **DIP-nek (Dual Inline Packages, kétlábsoros tokozás)** nevezik, de mindenki **lapkának (chipnek)** hívja, elkenve a különbséget a **szilíciumdarab** és a **tokozása** között.

INTEGRÁLT ÁRAMKÖRÖK

A legtöbb ismert tokozásnak **14, 16, 18, 20, 22, 24, 28, 40, 64** vagy **68** lába van.

A **nagy lapkánál** gyakran használatos a **négyzetes tokozás**, ahol mind a négy oldalon, vagy pedig az alsó részen találhatóak lábak.

Egy kaput **ideálisnak** nevezünk, ha a kimenet azonnal előáll, amint a bemenetet alkalmaztuk.

A valóságban a lapkáknek véges **kapukésleltetésük (gate delay)** van, amely tartalmazza mind a **jel terjedését** a lapkán keresztül, mind a **kapcsolási időt**.

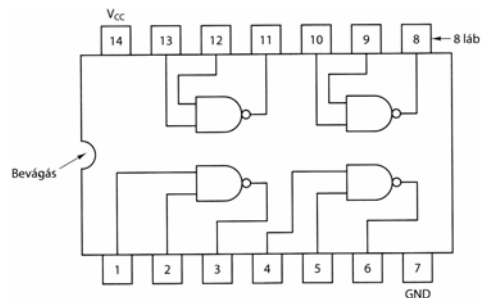
A **tipikus késleltetés 1-10 ns** között van.

INTEGRÁLT ÁRAMKÖRÖK

A lapkákat (nem pontos) **osztályokba** sorolhatjuk a **kapuk száma** alapján:

- **SSI (Small Scale Integrated – kis integráltságú)** áramkör: **1-10** kapu;
- **MSI (Medium Scale Integrated – közepes integráltságú)** áramkör: **10-100** kapu;
- **LSI (Large Scale Integrated – nagy integráltságú)** áramkör: **100-100000** kapu;
- **VLSI (Very Large Scale Integrated – nagyon nagy integráltságú)** áramkör: **> 100000** kapu.

INTEGRÁLT ÁRAMKÖRÖK



SSI lapka négy kapuval

KOMBINÁCIÓS ÁRAMKÖRÖK

A digitális logika nagyon sok alkalmazása megkívánja, hogy egy áramkör többszörös bemenettel és többszörös kimenettel rendelkezzen, és a **kimeneteit a pillanatnyi bemenetei határozzák meg**.

Az ilyen áramkört **kombinációs áramkörnek (combinational circuit)** hívjuk.

A memóriaelemeket tartalmazó áramkör tud **olyan kimeneteket generálni**, amelyek függenek a **tárolt értékektől** és természetesen a **bemenő változóktól** is.

Az ilyen tulajdonságú áramköröket **szekvenciális (sorrendi, sequential) áramköröknek** nevezzük.

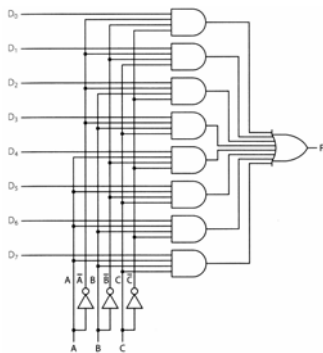
KOMBINÁCIÓS ÁRAMKÖRÖK

A digitális logika szintjén a **multiplexer (adatszelektor)** olyan áramkör, amely **2ⁿ adatbemenettel**, **1 adatkimenettel** és **n vezérlőbemenettel** rendelkezik, mely utóbbiak egy adatbemenetet választanak ki.

A kiválasztott adatbemenetre azt mondjuk, hogy a kimenetre **irányított** vagy „**kapuzott**” (gated).

A multiplexer fordítottja a **demultiplexer**, amely egy egyedi bemenő jelet irányít a **2ⁿ kimenet** valamelyikére az **n vezérlővonal értékétől** függően. Ha a vezérlő vonalak bináris értéke **k**, a **k-adik kimenet** a kiválasztott kimenet.

KOMBINÁCIÓS ÁRAMKÖRÖK



Nyolcbemenetes multiplexer áramkör

KOMBINÁCIÓS ÁRAMKÖRÖK

Egy olyan áramkört, amely n bites számot használ bemenetként, és pontosan egyet kiválaszt a 2^n kimenet közül (1-re állítja).

Az ilyen áramkör **dekódolónak** nevezzük.

Egy másik hasznos áramkör az **összehasonlító (comparator)**, amely két bemeneti szót hasonlít össze.

Tetszőleges függvények (igazságtáblázatok) megszerkeszthetők azáltal, hogy **ÉS** kapukkal logikai szorzatokat számolunk ki, és azután a szorzatokat **VAGY**-oljuk.

Egy nagyon általános lapka, a **programozható logikai tömb** vagy **PLA (Programmable Logic Array)** szolgál a logikai szorzat-összeg képzésére.

ARITMETIKAI ÁRAMKÖRÖK

Egy **léptető áramkör** a bemeneti jelek 1 bittel való eltolását végzi. Az eltolás irányát 1 darab vezérlővonal állapota határozza meg. Az áramkör bemeneteinek száma megegyezik kimeneteinek számával.

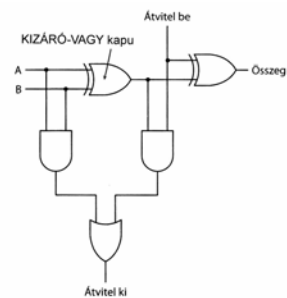
A **CPU** egyik leglényegesebb része egy **összeadást** végző áramkör.

Ezeknek két fajtája van az ún. **fél** és a **teljes összeadó áramkör**. Mindkettő képi az aritmetikai összeget és az átvitelt a következő helyiértékre.

A különbség közöttük az, hogy a **félösszeadó** nem veszi figyelembe az előző helyiértéken keletkezett átvitelt, a **teljes összeadó** viszont igen.

ARITMETIKAI ÁRAMKÖRÖK

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) A teljes összeadó igazságtáblázata;
b) Teljes összeadó áramkör.

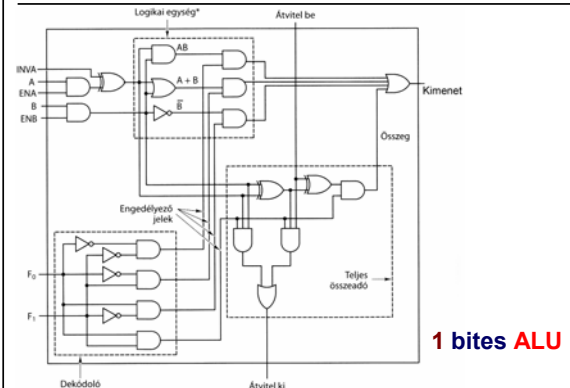
ARITMETIKAI ÁRAMKÖRÖK

A teljes összeadók között megkülönböztetik az **átvitelt tovább terjesztő összeadókat (ripple carry adder)**, illetve az **átvitelt kiválasztó összeadókat (carry select adder)**.

A legtöbb számítógép egyetlen áramkört tartalmaz az **ÉS**, **VAGY** végrehajtására és **két gépi szó összeadására**. Ez az áramkör tipikusan n bites szavakra készül, és n azonos áramkört tartalmaz az egyes bit pozíciókra.

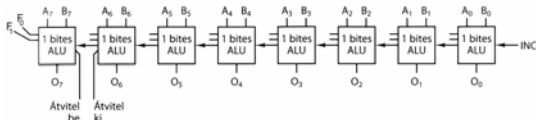
Az ilyen áramköröket **aritmetikai-logikai egységnek (Arithmetic Logic Unit)** nevezzük.

ARITMETIKAI ÁRAMKÖRÖK



1 bites ALU

ARITMETIKAI ÁRAMKÖRÖK



Nyolc 1 bites ALU-szelet összekapcsolása 8 bites ALU-vá. Az engedélyező és invertáló jelek nincsenek feltüntetve

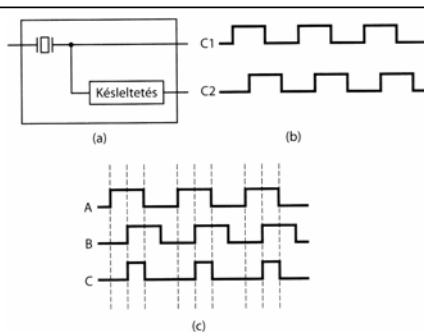
ÓRÁK

Az **óra (clock)** ebben az értelemben egy áramkör, amely pontosan meghatározott szélességű impulzusok sorozatát bocsátja ki, és nagyon precízen meghatározott a két egymás utáni impulzus közötti intervallum is.

A két egymást követő impulzus élei közötti intervallumot az **óra ciklusidejének (clock cycle time)** nevezzük.

A nagy pontosság eléréséhez az óra frekvenciáját általában egy **kristály-oszcillátor** vezérli.

ÓRÁK



- a) Óra;
- b) Időzítési diagram;
- c) Aszimmetrikus óra generálása.

MEMÓRIA

MEMÓRIAPARAMÉTEREK

A memória legkisebb tárolási egysége az egy bináris jel (0, vagy 1) tárolására szolgáló elemi rész. A tárolók ezen legkisebb egységét **bitnek (binary digit)** nevezik.

A processzor által fizikailag egy egységként kezelhető legkisebb memória terület ennél azonban nagyobb.

A fizikailag legkisebb egységként kezelt tárolóterületet **rekesznek (location, cell, cella)**. A mai gépeken ez az érték egységesen 1 byte (8 bit).

MEMÓRIAPARAMÉTEREK

Minden fizikailag önállóan kezelhető tárolóhely (rekesz) **címmel** rendelkezik, amely alapján a tárolóhelyet a processzor ki tudja választani és abban adatot tud elhelyezni, vagy adatot tud onnan kiolvasni.

A rekeszek címét 0-val kezdődő szigorúan növekvő sorszámok alkotják.

A címzés szempontjából, ha nem byte-onként, hanem pl. 4 byte-os szavanként kell elérni a tárolót, nagyon lényeges, hogy a szóhoz tartozó byte-ok milyen sorrendben kerülnek tárolásra. Két változat használatos: a „**big endian**” és a „**little endian**” tárolási forma.

MEMÓRIAPARAMÉTEREK

- A „**big endian**” tárolási formánál, a szó **legmagasabb** helyiértékű byte-ja kerül a **legalacsonyabb** című tárolóhelyre és a szó címzése a **legalacsonyabb** memóriacímmel történik (**SPARC**, **IBM** gépek).
- A „**little endian**” tárolási formánál, a szó **legalacsonyabb** helyiértékű byte-ja kerül a **legalacsonyabb** című tárolóhelyre és a szó címzése a **legalacsonyabb** memóriacímmel történik (**Intel** processzorok, **DEC VAX** gépek).

MEMÓRIAPARAMÉTEREK

Lényeges jellemző a **cím lehetséges mérete**, azaz az, hogy hány bináris helyiértéket lehet felhasználni a cím értékének leírására.

Ha ez a szám **16** bit, akkor a maximális tárolóhelyek száma: $2^{16} - 1 = 65535$.

A processzor ennél több tárolóhelyet közvetlen módon nem tud lekezelni, ez alkotja a **tárolóhelyek címezhető tartományát**.

Az ily módon értelmezett címtartomány alkotja egyúttal a **fizikai címek tartományát** is és a mögötte lévő, létező (installált), vagy nem létező tárolóeszköz a **fizikai tároló**, amely többnyire a főtár.

MEMÓRIAPARAMÉTEREK

Az aritmetikai műveletvégzés során, egy-egy számadat leírására nem elegendő **1** byte, ezért egy egységként **2-4-8** byte-ot használ a processzor. Ezt a feldolgozásoknál, adatátvitelknél használt méretet **szónak** (**word**) nevezzük.

A központi táruk működésének fontos jellemzője még az **elérési idő** (**access time**) és a **ciklusidő** (**cycle time**).

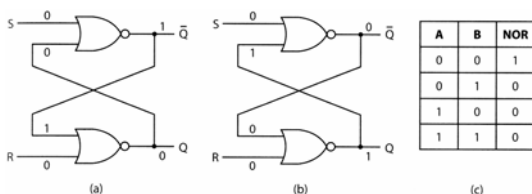
Elérési idő az időtartam, amely a kiolvasás megkezdése és az adatnak a tároló kimenetén való megjelenése között eltelik.

MEMÓRIAPARAMÉTEREK

A **ciklusidő** ennél valamivel hosszabb időtartam, mert magában foglalja a kiolvasás utáni **feléledési időt** (**recovery time**) is, amelyre egyes memóriáknak szüksége van a következő memóriához fordulást megelőzően.

A **statikus RAM**-ok **elérési ideje** és **ciklusideje** közel azonos értékű, míg a **dinamikus RAM**-ok **ciklusideje** kb. **kétszerese** az elérési időnek.

TÁROLÓK



- NEM-VAGY** kapukból álló **SR-tároló 0-s** állapotban;
- NEM-ÉS** tároló **1-s** állapotban;
- NEM-VAGY** igazságtáblázata.

TÁROLÓK

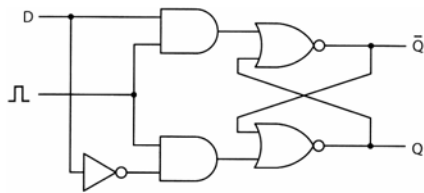
Az előző ábra áramkörét **SR-tárolónak** (**Set Reset Latch**) hívjuk.

Gyakran fontos, hogy a tároló állapotváltozásai csak bizonyos meghatározott pillanatban történjenek.

Ezeket az áramköröket egy újabb bemenettel bővítették, melyeket **érvényes** (**enable**), illetve a **kapuzójel** (**strobe**) bemenetnek neveztek el. Amikor ezen bemenet értéke **1**, akkor az áramkör érzékeny az **R**, **S** bemenetek állapotváltozásaira.

Ezeket az áramköröket **időzített** (**clocked**) **tárolóknak** (**reteszeknek**) nevezik.

TÁROLÓK



Időzített D-tároló

FLIP-FLOPOK

Sok áramkörtél szükséges lehet, hogy meghatározott időpontban vegyen mintát bizonyos vonalon levő értékről, és tárolja azt az értéket.

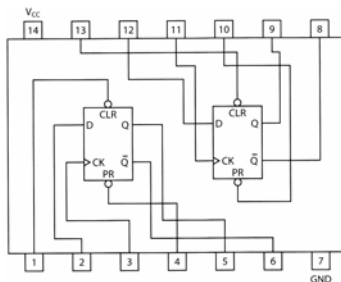
Ezt a változatot **flip-flopnak** (flip-flop, billenőkör) nevezzük.

Flip-flop esetén az órajel 1-es állásánál nem fordul elő az állapotváltozás, hanem csak akkor, amikor az órajel átmegy 0-ból 1-be (felfutó él) vagy 1-ből a 0-ba (lefutó él). Így az órajel hossza nem lényeges, ha elég gyors az átmenet.

A flip-flop **élvezérelt** (edge triggered), míg a tároló **szintvezérelt** (level triggered).

Több flip-flopból alakítanak ki **regisztereket**, melyek alkalmasak több bit tárolására is.

FLIP-FLOPOK



Duális D-flip-flop

MEMÓRIASZERVEZÉS

Nagyobb memóriák építéséhez **másfajta szervezés szükséges**, mint egyszerűen a flip-flopok számának növelése.

A memóriaszervezés itt olyan, hogy **egyedi szavakat tudunk megcímezni**.

Az ilyen szervezésű **4x3** bites szóból álló memóriának a kapacitása **12** bit, ami alig több, mint egy **8-as** flip-flopé, viszont kevesebb lábát igényel és a tervezés könnyen kiterjeszhető nagy memóriákra.

Elvileg ezt a memóriát egy **14** lábás tokozásba el lehet helyezni, beleértve a **tápfeszültséget** és a **földet**, szemben a **20** lábás oktális flip-floppal.

MEMÓRIASZERVEZÉS

Ennek a **4x3** szavas memóriának **8** bemenő és **3** kimenő vonala van.

Ezek:

- 3 adatbemenet;
- 2 cím bemenet;
- 3 vezérlő bemenet;
 - **CS (Chip Select)** a memóriaplakka kiválasztásához;
 - **RD (Read)** az olvasás és írás megkülönböztetésére;
 - **OE (Output Enable)** a kimenet engedélyezésére;
- 3 adatkimenet.

MEMÓRIASZERVEZÉS

Az előző memóriának a **bemeneti** és a **kimeneti adatvonalak különbözőek**, ezzel ellentétben a **valódi memóriáknak azonosak**.

Ahhoz, hogy **kiküszöböljük a kimeneti és a bemeneti adatok ütközését**, szükségünk van egy speciális kapcsolóra, az ún. **nem-invertáló pufferre** (noninverting buffer). Ez egy **vezérelt kapcsoló**: ha a vezérlő bemenetére magas jel érkezik, a puffer **huzalként** viselkedik; ha alacsony – akkor **szakadásként**.

Az ilyen puffer áramkörök háromállapotú eszközök, mert ki tudnak bocsátani **0-t**, **1-t** vagy **semmit** (nyitott áramkör, **szakadás**, nagyimpedanciájú állapot).

MEMÓRIASZERVEZÉS

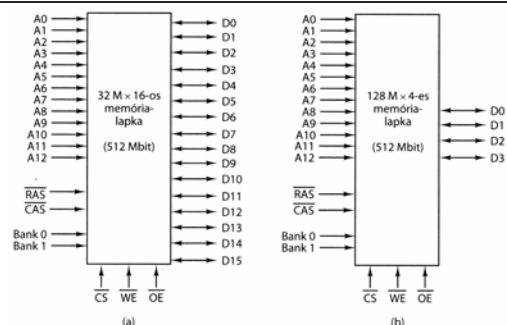
A pufferek a jeleket is erősítik, így sok bemenetet tudnak egyszerre meghajtani.

A pufferek szerepe az említett memóriánál, hogy engedélyezzék a kimenetet, vagy lekapcsolják azokat az áramkör további részeseiről, a vezérlő bemenetek függvényében.

Az ilyen szervezésű memória könnyen kiterjeszhető nagyobbra: növeljük a memóriaszavak számát és a szóhosszt.

Az ilyen típusú struktúránál, a maximális hatékonyság érdekében a memóriában lévő szavak számának kétfelváltó hatványának kell lennie, de a szavakban lévő bitek száma bármennyi lehet.

MEMÓRIALAPKÁK



512 Mbit-es memórialapka két lehetséges szervezése

MEMÓRIALAPKÁK

Az előző diáról a memórialapka magyarázata:

- \overline{CS} – Chip Select, memórialapka kiválasztása;
- \overline{OE} – Output Enable, kimeneti jelek engedélyezése;
- \overline{WE} – Write Enable, írás engedélyezése (az írás/olvasás megkülönböztetése);
- \overline{RAS} – Row Address Strobe, sorcím kapuzójele;
- \overline{CAS} – Column Address Strobe, oszlop kapuzójele;
- Bank i – Modul kiválasztás;
- A_i – Cím bemenetek;
- D_i – Adat bemenetek;

Hatalmas memórialapkákat gyakran $n \times n$ -es mátrixként építenek föl, amelyek címezése sor és oszlop megadásával történik.

MEMÓRIALAPKÁK

Az előző dia memórialapka két eltérő és egymástól független tervezési elvet mutatnak be.

Az első a kimeneti szélesség (bitekben): 1, 2, 4, 8, 16 vagy hány bitet továbbít egyszerre a lapka.

A második, hogy az összes címbit megjelenik-e egyszerre a lábakon, vagy a sorok és oszlopok egymás után jelennek meg.

A tervezőknek mindkét kérdésre válaszolniuk kell, míg mielőtt elkezdik a lapka tervezését.

RAM-ok és ROM-ok

Az eddig megismert memóriák mindegyikét lehetett írni és olvasni.

Ezeket a memóriákat RAM-nak (Random Access Memory, véletlen elérésű memória) hívjuk.

Két változata van a RAM-oknak, a statikus és dinamikus.

A statikus RAM-ok (SRAM) belső áramkörei hasonló felépítésűek, mint a D-flip-flop. Ezeknek a memóriáknak az a tulajdonsága, hogy tartalmuk addig marad meg, amíg a memória áramellátása biztosított.

RAM-ok és ROM-ok

A statikus RAM-ok nagyon gyorsak. A tipikus elérési idő néhány ns. Emiatt a statikus RAM-ok nagyon népszerűek második szintű gyorsítótárak (memóriák) építésére.

A legjobb statikus RAM-ban is egy 1 bites memóriacella kiépítéséhez 6 tranzisztor szükséges.

A dinamikus RAM-ok (DRAM) nem flip-flopot használnak.

RAM-ok és ROM-ok

A dinamikus **RAM** egy cellája **1 tranzisztort** és egy kicsi **kondenzátort** tartalmaz. A kondenzátor **feltöltött** vagy **kisült**, annak megfelelően, hogy **0-t** vagy **1-t** tárol.

Mivel az elektromos töltés hajlamos a szivárgásra, a **dinamikus RAM**-ban minden bitet **frissíteni** kell (**refresh**, **reload**) néhány ezredmásodpercenként, hogy az adat megmaradjon.

Külső áramkörök (logikák) vigyáznak a frissítésre, ezért a **DRAM**-ok összetettebb kapcsolódást igényelnek, mint a statikus **RAM**-ok.

RAM-ok és ROM-ok

A **DRAM**-ok lassabbak (néhány **tíz ns**), mint az **SRAM**-ok, viszont sokkal könnyebben integrálhatók kisebb helyre, ezért nagyobb a tárcapacitásuk.

A valós számítógép-rendszerekben a **statikus RAM gyorsítótárak** és a **dinamikus RAM főmemóriák** kombinációja egyesíti mindegyikük jó tulajdonságait.

Dinamikus **RAM** lapkák típusai:

- **FPM** (**F**ast **P**age **M**ode, gyors lapkezelésű);
- **EDO** (**E**xtended **D**ata **O**utput, kiterjesztett adatkimenetű);
- **SDRAM** (**S**ynchronous **D**RAM, szinkron **DRAM**);
- **DDR** (**D**ouble **D**ata **R**ate, dupla adatsebességű).

RAM-ok és ROM-ok

Az **FPM DRAM** belseje egy bitmátrix és úgy dolgozik, hogy a hardverben megjelenik egy sorcím, a következő lépésben egy oszlopcím. Külön jel szólítja fel a memóriát, amikor dolgoznia kell, így a memória a fő rendszerórától eltérő szinkronizációval működik.

Az **EDO DRAM** megenged egy második memóriahivatkozás megkezdését, mielőtt az előző befejeződött volna. Ennek az egyszerű szállítószalagelvnek az alkalmazása az egyedi memóriahasználatot nem teszi gyorsabbá, de javítja a memória sávszélességét azzal, hogy több szót ad másodpercenként.

Az **FPM** és **EDO DRAM** ciklusideje **>12 ns**-nál.

RAM-ok és ROM-ok

Az **SDRAM** a statikus és dinamikus **RAM**-ok keveréke, és a fő rendszeróra vezérli.

Az **SDRAM** előnye, hogy az óra kiküszöböli azokat a vezérlőjeleket, amelyek megmondják a memórialapknak, hogy mikor válaszoljon.

Ehelyett a **CPU** megmondja a memóriának, hogy hány ciklusban kell működnie, azután elindítja.

Minden egymást követő ciklusra a memória **4**, **8**, vagy **16** bitet bocsát ki attól függően, hogy hány kimeneti vonala van. A vezérlőjelek elhagyása növeli az adatátviteli sebességet a **CPU** és a memória között.

RAM-ok és ROM-ok

A **DDR SDRAM** memórialapkája adatot bocsát ki mind az órajel felfutó élénél, mind a lefutó élénél, megduplázva az adatsebességet.

Egy **8 bites** széles **DDR** lapka **200 MHz**-es környezetben másodpercenként **200 milliószor** két alkalommal **8 bites** adatot bocsát ki (egy rövid intervallumban természetesen), teljesítve az elméleti **3,2 Gbps**-os csoportos (**burst**) **átviteli sebességet**.

RAM-ok és ROM-ok

A memórialapkák másik nagy csoportja a nem felejtő lapkák, melyek tartalma megmarad, még akkor is, ha az áramot kikapcsoljuk.

A **ROM** (**R**ead **O**nly **M**emory, csak olvasható memória) tartalma nem változtatható meg, nem törölhető sem belsőleg, sem más módon.

A **ROM**-ban tárolt adatok betöltése a gyártás során történik.

Egy fényérzékeny maszkon át történő megvilágításával kezdődik.

RAM-ok és ROM-ok

A maszk tartalmazza a kívánt bitmintát, azután a megvilágított (vagy nem megvilágított) felület **kimaratása** következik.

Csak egyetlen módon lehet megváltoztatni a **ROM**-ban tárolt programot, mégpedig úgy, hogy a teljes lapkát kicseréljük.

A **ROM**-ok jóval olcsóbbak, mint a **RAM**-ok.

A **PROM** (**Programmable ROM**, programozható **ROM**) hasonlít a **ROM**-hoz, de **egyszer mező-programozható**.

Sok **PROM** egy **apró biztosítékból** álló tömböt tartalmaz.

RAM-ok és ROM-ok

A speciális olvadó biztosítékot úgy égetjük ki, hogy a megfelelő sorokat és oszlopokat kiválasztjuk, és ezután nagy feszültséget kapcsolunk a lapka megfelelő lábára.

Az **EPROM** (**Erasable PROM**, törölhető **PROM**) nemcsak mező-programozható, hanem mező-törölhető is.

Azzal, hogy az **EPROM**-ot **kvarcüveg** ablakon át erős **ultraviola fénnel 15 percig** megvilágítjuk, az összes bitet **1-re** állítjuk. Ha sok változás szükséges a tervezési ciklus alatt, az **EPROM**-ok sokkal gazdaságosabbak, mint a **PROM**-ok, mert újra felhasználhatók.

RAM-ok és ROM-ok

Az **EPROM**-ok ugyanazt a memóriaszervezést tartalmazzák, mint a statikus **RAM**-ok.

Az **EPROM**-oknál még jobbak az **EEPROM**-ok (**Electrically Erasable PROM**, elektromosan törölhető **PROM**), amelyek elektromos impulzusokkal törölhetők ahelyett, hogy egy speciális kamrában ultraviola fénnel világítanánk meg őket.

Ráadásul az **EEPROM** a helyén programozható újra, míg az **EPROM**-ot ehhez be kell tenni egy speciális **EPROM**-programozó berendezésbe.

RAM-ok és ROM-ok

A negatívum, hogy a legnagyobb **EEPROM**-ok kapacitása tipikusan csak **1/64** része a közönséges **EPROM**-okénak, és csak fele olyan gyorsak. Az **EEPROM**-ok nem tudják felvenni a versenyt a **DRAM**-okkal vagy az **SDRAM**-okkal, mert **tízszer** lassabbak, **százszor** kisebb a kapacitásuk és jóval drágábbak. Ezeket csak akkor alkalmazzák, amikor „nem felejtő” képességük a fő szempont.

A legújabb **EEPROM** típus a **flash memória** (**flash memory**). Az **EEPROM**-hoz hasonlóan ugyancsak **elektromosan törölhető**.

RAM-ok és ROM-ok

Viszont szemben az **EEPROM** **bájtonkénti** írásával/olvasásával, a **flash memória** lehetővé teszi a **blokkonkénti** törlést és újraírást.

Jól integrálhatók, viszonylag olcsók. Gyorsak, az elérési idejük **50 ns**.

Az **írási/olvasási ciklusok** száma eléri a **100000**, majd elhasználódnak.

Amíg a mérnökök meg nem oldják ezt a problémát, addig még **mágneses tárolókat** használunk **háttértárolóként**.

Megtalálhatók digitális kamerákban, **mp3-lejátszóknak**, pendrive-okban, stb.

RAM-ok és ROM-ok

Típus	Kategória	Törlés	Bájt-változtatás	Felejtés	Tipikus használat
SRAM	Olvasás/írás	Elektromos	Igen	Igen	2-es szintű gyorsítótár
DRAM	Olvasás/írás	Elektromos	Igen	Igen	Fő memória (régli)
SDRAM	Olvasás/írás	Elektromos	Igen	Igen	Fő memória (új)
ROM	Csak olvasás	Nem lehetséges	Nem	Nem	Nagy berendezések
PROM	Csak olvasás	Nem lehetséges	Nem	Nem	Apró berendezések
EPROM	Főleg olvasás	UV-fény	Nem	Nem	Prototípusok
EEPROM	Főleg olvasás	Elektromos	Igen	Nem	Prototípusok
Flash	Olvasás/írás	Elektromos	Nem	Nem	Filmként digitális kamerához

A különböző típusú memóriák összehasonlítása.

CPU lapkák

Minden modern **CPU** egyetlen lapkán helyezkedik el. A kapcsolatuk a rendszer többi részével jól definiált. Minden egyes **CPU** lapkának van **lábkészlete**, ezen keresztül tartja a kapcsolatot a külvilággal. A **CPU lábkiosztásából** (**pinout**) megtudhatjuk minden egyes láb funkcióját.

Néhány láb **kimeneti jeleket** kap a **CPU**-tól; mások **jeleket fogadnak** a külvilágból; **néhány mindkettőre képes**. Az **összes láb funkciójának megértésével** megtanulhatjuk, hogyan **kapcsolódik** a **CPU** a **memóriához**, a **bemeneti/kimeneti (B/K, output/input, I/O) berendezésekhez** a **digitális logika szintjén**.

CPU lapkák

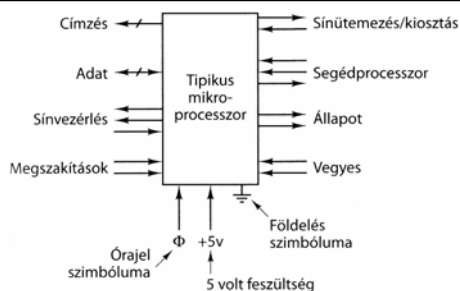
A **CPU** lapka lábait három csoportba osztjuk:

- **Cím**;
- **Adat**;
- **Vezérlés**.

Ezek a lábak **párhuzamos huzalokon** keresztül kapcsolatban vannak a **memória** és **B/K** lapkák hasonló lábaival, ezeket a **párhuzamos huzalokat síneknek** hívjuk.

Egy utasítás betöltéséhez a **CPU** először **beállítja az utasítás memóriacímét** a címlábain.

CPU lapkák



Általános CPU logikai lábkiosztása.

A nyilak jelzik a **bemeneti** és **kimeneti jeleket**. A rövid átlós vonalak jelzik a többszörös lábak használatát, egy-egy konkrét **CPU**-nál **egy szám** jelzi a többszörözés számát.

CPU lapkák

Ezután **beállít egy vagy több vezérlővonalat**, informálva a **memóriát**, hogy **olvasni szeretne (pl.) egy szót**.

A **memória** a **kért szónak a CPU adatlábaikra helyezésével** válaszol, és **beállít egy jelet**, amely azt mondja, hogy ez megtörtént. Amikor a **CPU** látja ezt a jelet, **elfogadja a szót és végrehajtja az utasítást**.

Egy **utasításnak** szüksége lehet **adatszavak olvasására és írására**, ebben az esetben a teljes folyamat minden további szó esetében megismétlődik.

CPU lapkák

A **CPU** a **memóriával** és a **B/K berendezésekkel** úgy tartja a kapcsolatot, hogy **jeleket küld és fogad** a lábain. Másféle kapcsolat nem lehetséges.

Két lényeges paraméter határozza meg a **CPU teljesítményét**: az egyik a **címlábak**, a másik az **adatlábak száma**.

Egy lapka **m** címlábbal **2^m** memória helyet tud megcímezni (**m = 8, 16, 32, 64**).

Egy lapka **n** adatlábbal **n bites** szavakat tud írni és olvasni egyetlen művelet során (**n = 8, 16, 32, 64**).

CPU lapkák

A cím- és adatlábakon kívül minden **CPU**-nak van néhány **vezérlő jele**.

A **vezérlő jelek szabályozzák** a folyamatot, a **CPU**-ból és a **CPU**-ba mozgó adatok időzítését, és további **vegyes funkciók** is vannak.

Minden **CPU**-nak vannak lábai az **áramellátásra** (általában **+3,3** vagy **+5 V**), a **földelésre** és **órajelre** (pontosan meghatározott **négyszögjel**), de a többi láb lapkáról lapkára erősen változik.

CPU lapkák

A **vezérlő lábakat** a következő főbb kategóriákba csoportosítják:

- **Sínvezérlés**;
- **Megszakítások**;
- **Sínütemezés/kiosztás**;
- **Segédprocesszor jelei**;
- **Állapot**;
- **Vegyes**.

A **sínvezérlő lábak** főként a **CPU-ból** a sínre történő kimenetként szolgálnak (ezek egyben bemenetek a **memória-** és a **B/K** lapkák számára).

CPU lapkák

Továbbá megmondják, hogy a **CPU olvasni** vagy **írni** akar, vagy valami mást akar tenni. Ezeket a lábakat a **CPU** a rendszer többi részének a **vezérlésére használja**, és megmondja, hogy mit szeretne tenni.

A **megszakítási lábak** a **B/K** berendezésektől a **CPU-ba** tartó bemenetek. A legtöbb rendszerben a **CPU** megmondja egy **B/K berendezésnek (eszköznek)**, hogyan kezdje el a műveletet, és azután magára hagyja, hogy valami hasznosat csináljon azalatt, amíg a lassú **B/K** eszköz elvégzi feladatát.

CPU lapkák

Amikor a **B/K** eszköz befejezte, **B/K** vezérlő lapka e lábak egyikén jelet küld, hogy **megszakítsa a CPU-t**, és az **kiszolgálja a B/K** berendezést, pl. ellenőrizze, hogy volt-e **B/K** hiba. Néhány **CPU-nak** **kimeneti lába** is van; ezen **nyugtázza a megszakító jelet**.

A **sínütemezés lábak** a sínen a **forgalom irányításához** szükségesek: **megakadályozzák, hogy két berendezés egyszerre használja a sítet**. A sínütemezés tekintetében a **CPU** egy berendezésnek számít a sínen, és akárcsak a többi berendezésnek **kérnie kell a sín használatát**.

CPU lapkák

Néhány **CPU** lapkát úgy terveztek meg, hogy **együttműködjön a segédprocesszorral (coprocessor)**, pl. lebegőpontos, grafikus vagy más lapkával. A **CPU** és a **segédprocesszor** közötti **kommunikáció** biztosítására **speciális lábak** szolgálnak, amelyek küldik és fogadják a különböző kéréseket.

Ezekon kívül néhány **CPU-nak** más célú lábai is vannak. Ezek némelyike az **állapotra vonatkozó információt szolgáltat** vagy **fogad**, míg mások a számítógép **alaphelyzetbe állításánál** használatosak, és megint mások biztosítják a **kompatibilitást** a korábbi **B/K** lapkákkal.