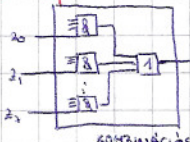
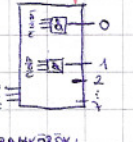


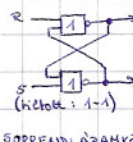
**Multiplexer:**



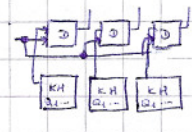
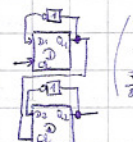
**2 Multiplexer:**



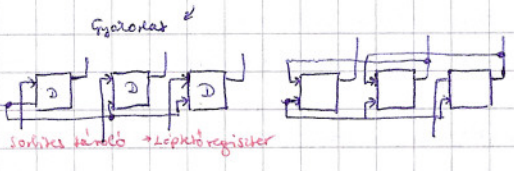
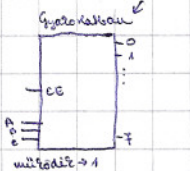
**Flip-Flop (RS)**



**D-Flip:**

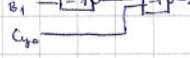


közvetlen átvétel  
000...0 → D=0

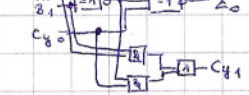


széles tároló → lepkőregiszter

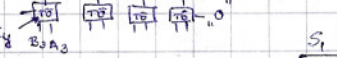
**Uvaválogó:**



**2 bites összeadó:**

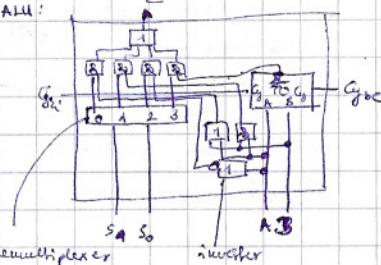


**4 bites összeadó: bármely szám átvitel**



**4 bites ALU:**

S <sub>1</sub>	S <sub>0</sub>	E
0	0	V
0	1	A
1	0	A
1	1	A+B



**Cache: (2B)**

adattárolás, folyamatos igényű tárolás, puffertároló, egyes tárolók: művelet elvégzése és tárolás, a felhasználó számára láthatatlan, gyors műveletű tároló → cache tár, v. gyorsító tár

fontos szerepe van: adatforgalom csökkentése, csökkentés az adatok átmeneti tárolására egyaránt szolgál, gyors, fell. számára nem elérhető

Adattároló a cache és a memória közötti gyors forgalom → gyorsabb több byte-ot ír az → azaz egyaránt, illetékes! Vímézetes módja: tárolókat mint → a cache-be tárolt adatok való igazán integrálva

jellemzői: mérete (8-256KB), beolvasás, írás, helyettesítési alap, aszinkronizációs módok, adategyeztető-kezelési mód



h) 1) **egyszeres asszociatív cache:** a kérés asszociatív tábla beolvasott fele barokka elhelyezhető, elhelyezés során a helyettesítési algoritmus határozza meg,

• proc. adatot keres: a memóriában cím felső 28 bitek (Holtzsonum) összehasonlítja a cache-beli beolvasottakkal → az egyidejűleg történik → 256 // összehasonlító áramlót tart, • névleges keresés: a cím alsó 4 bite alapján kijelöl az adott sorbeli byte-ot, • névleges keresés: memóriában ellenőri a byte-ot, beolvasa

• Előny: nagy sebesség a beolvasásban, • Hátrány: helyettesítési eljárás kell alkalmazni, irányoztatás az összehasonlító, kell, ahogy van a cache-nél. Későbbes, • új bit kapcsolás (új) 2 bit sorindar

2) **kétszeres asszociatív cache:** a tábla van a megkér. helyre kerülhet, a beolvasott byte a cache-be a Holtzsonum alsó 8 bite hat. meg. 28 bite beolvas. • új bit kapcsolás (új) 2 bit sorindar

• adat beolvasás: a tábla a memóriában előadott tábla alapján kijelöl az adott sorbeli byte-ot, a felső 20 bitek összehasonlítja a kérelem. • névleges keresés: a cím alsó 4 bite alapján kijelöl az adott sorbeli byte-ot, • névleges keresés: a memóriában ellenőri a beolvasott byte-ot, beolvasa, • Előny: rövid tábla, gyors vímézetes, olcsó, • Hátrány: több beolvasott tábla megkér. az azaz indexet helyre adhatna az elv. tábla

3) **ágyas asszociatív cache:** a cache szaggat, a sorbeli átl. csoportokra van osztva, Jellek használata művelet = a csoportok beolvasása a helyre kerülésére kerülhet beolvas.

• keresés: a memóriában tárolt csoportok alapján kijelöl az indexet megjelölt csoport, majd a csoportokat (felső 28 bite) összehasonlítja a kérelem (cache-beli csoportok), → az összehasonlítás minden sorban történik → 4 // összehasonlító áramlót tart, • névleges keresés: az alsó 4 bite alapján kijelöl az adott sorbeli byte-ot, • névleges keresés: a memóriában ellenőri a beolvasott byte-ot és beolvasa, • Előny: nagy sebesség, kis méretű összehasonlító áramlót kell kezel, irányoztatás gyors,

4) **széles asszociatív cache:** a proc. van a csoportok helyre kerülésére asszociatív tábla, azaz van a tábla helyre kerülés, fontosság 1) - van

8a) a tábla adatok csoportokra osztott információkat is tartja a cache. Ezer a vezérlés és a helyettesítési eljárás elvégzése

1) **valid bit:** a cache tárolásának (bit, sor, byte) az érvényesség jele (az adat a megkér. című tárolóhelyre tartozik-e, általában olvasás-adat-c), • tárolókor (reset) minden 0 bit, új adat beolvasásakor 1 bit, • minden beolvasott bit tartozik, utólag 1-1 byte-ot 1-1 bit.

2) **dirty bit:** a beolvasott adatok a memóriában, felülírhatják jele, PD=1 esetén nem lehet beolvasni új beolvasott, abba a nyitott állat a tábla van vímézetes

**vímézetes:** a bit nem ad meg (hit bit), • sor nem ad meg (read bit), • legyőzött hely (hit bit), • legyőzött hely (write bit)

1A) **RISC:** • egyszerű architektúra, • rögzített idő tárolás, • aszinkron tárolás, • tároló használata utólagos írás lehet, (tartalom a memóriában a tábla van vímézetes) • tárolókor (reset) minden 0 bit, új adat beolvasásakor 1 bit, • minden beolvasott bit tartozik, utólag 1-1 byte-ot 1-1 bit.

**CISC:** • egyszerű architektúra, • rögzített idő tárolás, • aszinkron tárolás, • tároló használata utólagos írás lehet, (tartalom a memóriában a tábla van vímézetes) • tárolókor (reset) minden 0 bit, új adat beolvasásakor 1 bit, • minden beolvasott bit tartozik, utólag 1-1 byte-ot 1-1 bit.

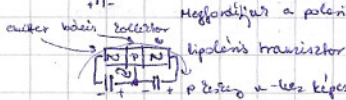
Földelés: az áramkörök a földre kerülnek. Föld = ágyúval → van 1 vonalra építve c-ja → hirtelen észlelhető az ágyúval ezeket alakít ki

$\frac{0-0}{0-0} \rightarrow$  elektr. áram

3dc ágyúval megvan a h-je +, kiadó len → ellenkező irányú váltakozás, e- földet = E-h-pont, kiadó T-h-pont

$\frac{0-0}{0-0} \rightarrow$  diffúzió  
3dc földet nem megvan

az e- kódot elülső a részec → hirtelen a katódban → 3dc váltakozás



Megfordított a polaritás → c- közből részecél ahogyan a katódban → betöltés → kiadó len → ahogyan a katódban → váltakozás len.

p- réteg n- les képest, + csatlakozás jött mellé, a másik áramirányban van. (NPN, PNP tranzisztor)

$\frac{I_c}{I_b} \approx \beta \rightarrow 20, 100$  Milyen kis árammal vagy áramot lehet befolyásolni → folyjon erre a bázisárammal → ezek kölcsönhatást fejtettek ki



egyszerűsítve megvan az n- nívót, áramot, kiadó, kiadó, kiadó a földet → emelő rész is kiadó megvan. Tehát a katódban nívót, kiadó kékje kiadó, kiadó, kiadó → földet kapcsolata rá. → a katódban kiadó len áramot az e- réteg → kiadó len az detektáló nívót → nem folyó áram

kiadó a katódban:  $\frac{I_c}{I_b} \approx \beta$  + földet, kiadó megvan kiadó kiadó → kiadó kiadó kiadó → kiadó folyó → kiadó kiadó v. kiadó kiadó FET

MOSFET nívót kiadó:  $I_c = \beta I_b$  CHOS = a kiadó kiadó 1) NFET 2) PFET

1) ha a kiadó kiadó a föld kiadó kiadó → kiadó kiadó kiadó kiadó. kiadó kiadó kiadó kiadó kiadó kiadó (kiadó kiadó)

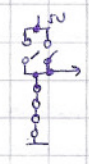
2) ha kiadó kiadó, kiadó kiadó  
→ kiadó kiadó kiadó kiadó kiadó → kiadó kiadó kiadó

1) ha a kiadó kiadó kiadó kiadó → kiadó kiadó kiadó kiadó  
2) ha a kiadó kiadó kiadó kiadó → kiadó kiadó kiadó kiadó

NB: F kiadó kiadó, a kiadó kiadó kiadó



kiadó kiadó, kiadó kiadó  
föld kiadó kiadó, kiadó kiadó  
ha 0-t kiadó 5V jött ki, ki 5V-t → 0. → inverter



A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

NOR

nyitott kiadó a kiadó → FV kiadó kiadó  
kiadó kiadó a kiadó → 0V nem kiadó kiadó kiadó  
FV kiadó kiadó  
0V nem kiadó kiadó kiadó  
kiadó kiadó van, de ki 5V kiadó

adatok: DC kiadó kiadó HAR kiadó, MDR kiadó

CALL kiadó: PC → [SP], SP-1 = SP, új kiadó → PC,

adatok: DC → HAR, A → MDR, MV kiadó

RET kiadó: SP+1 = SP, [SP] → PC

MOVE, PUSH → POP