BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Rendszerarchitektúrák

Processzorok - bevezetés

2021. tavaszi félév

Szántó Péter
Raikovich Tamás

Áttekintés

- **A későbbi előadásokon**
  - RISC-V processzor
  - Analog Devices Blackfin DSP processzor
  - Tensilica Xtensa konfigurálható processzor
- **A mai előadáson**
  - Alapfogalmak áttekintése

Programozható processzorok

- **Programozható processzor: általános célú műveletvégző**
  - (Általános célú) adatstruktúra + vezérlő
  - Adatstruktúra alap műveletek
    - Adatmozgatás
    - Aritmetikai: összeadás, kivonás (szorzás, osztás)
    - Logikai: bitenkénti AND, OR, XOR
    - Léptetés, forgatás
    - Státusz bitek: zero (Z), carry (C), negative (N), overflow (V)
  - Vezérlő
    - Az utasítások beolvasása, értelmezése és végrehajtása az adatstruktúrán
    - Feltétel nélküli és feltételes ugrás, szubrutinhívás
- **Programozás**
  - Az adott feladat leképzése a processzor utasításkészletére

CISC processzorok

- **Complex Instruction Set Computer (↔ RISC)**
- **Ez a típus volt előbb**
- **Egy utasítás több alacsonyszintű műveletből állhat**
  - Pl. memóriában tárolt érték megnövelése:
    - Memória olvasás, aritmetikai művelet, memória írás
  - Változó órajelciklus igény
  - Magasszintű nyelvi elemek közvetlen támogatása
- **Sokféle címzési mód**
- **Nagyszámú és változó méretű utasításkészlet**
- **Kevés belső regiszter (akkumulátor alapú)**
- **Neumann architektúra: egyetlen közös memória interfész**
- **Mikroprogramozott vezérlő (mikrokód)**
- **Példák: MOS 6502, Intel 8051, Intel x86, Zilog Z80**
  - A mai x86 processzorok belül már RISC felépítésűek!

RISC processzorok

- **Reduced Instruction Set Computer (↔ CISC)**
- **A CISC utasítások helyettesítése egyszerűbb műveletekkel**
  - 1 órajel alatt végrehajthatók
  - Pipeline felépítés és utasítás végrehajtás
- **Kevés címzési mód, ortogonális utasításkészlet**
  - Az utasításoknál használható az összes címzési mód
- **Fix méretű utasításkészlet**
- **Load-store architektúra**
  - Műveletvégzés csak belső regisztereken
  - CISC: memóriában tárolt érték növelése → RISC: 3 utasítás
- **Sok belső regiszter**
- **Harvard architektúra: külön program- és adatmem. interfész**
- **Példák: ARM, RISC-V, IBM PowerPC, MIPS, Xilinx MicroBlaze**

CISC processzorok – MOS 6502

- **1975-ben készült 8 bites CISC processzor**
- **16 bites cím és 8 bites adat interfész**
- **Kb. 3500 tranzisztor**
- **Az olcsó ára miatt ($25) nagyon népszerű volt**
  - Commodore 64, Nintendo Entertainment System
- **Regiszterkészlet**
  - Egy 8 bites akkumulátor (A)
  - Két 8 bites index regiszter (X és Y)
  - Egy 8 bites veremmutató (SP)
  - Egy 16 bites programszámláló (PC)
  - Egy 8 bites státusz regiszter (P)

CISC processzorok – MOS 6502

- **Az memória alsó 256 bájtos tartományának kitüntetett a szerepe**
  - Zero Page (ZP), 256 elemű 8 bites „regisztertömb”
- **Címzési módok**
  - ***Implicit:*** nincs operandus
  - ***Akkumulátor – A:*** a művelet az akkumulátoron hajtódik végre
  - ***Immediate – #imm:*** konstans ALU operandus
  - ***Zero page – d:*** 8 bites ZP memóriacím, ZP[d]
  - ***Abszolút – a:*** 16 bites (teljes) memóriacím
  - ***Relatív – r:*** a PC-hez relatív 8 bites memóriacím PC=PC + r
  - ***Indirekt – (a):*** az operandus beolvasása 16 bites mem. címről
  - ***Zero page indexelt – d, X / d, Y:*** ZP[d + X] vagy ZP[d + Y]
  - ***Abszolút indexelt – a, X / a, Y:*** MEM[a + X] vagy MEM[a + Y]
  - ***Indexelt indirekt – (d, X):*** MEM[(ZP[d + X + 1] << 8) + ZP[d + X]]
  - ***Indirekt indexelt – (d), Y:*** MEM[((ZP[d + 1] << 8) + ZP[d]) + Y]

CISC processzorok – Intel 80386

- **Regiszterkészlet (32/16 bites) – akkumulátor alapú**
  - Fő regiszterek: **EAX/AX**, EBX/BX, ECX/CX, EDX/DX
  - Index regiszterek: ESI/SI, EDI/DI, EBP/BP, ESP/SP
  - Programszámláló: EIP/IP
  - Szegmens regiszterek (16 bitesek): CS, DS, ES, FS, GS, SS
  - Státusz regiszter
- **Memória címzési módok (változó hosszúságú utasítások!)**
  - 16 vagy 32 bites abszolút
  - Regiszter indirekt
  - Regiszter + offszet indirekt
  - Regiszter + regiszter indirekt
  - Regiszter + regiszter + offszet indirekt
  - Regiszter + (regiszter * skála) + offszet indirekt
    - A skálatényező 1, 2, 4 vagy 8 lehet

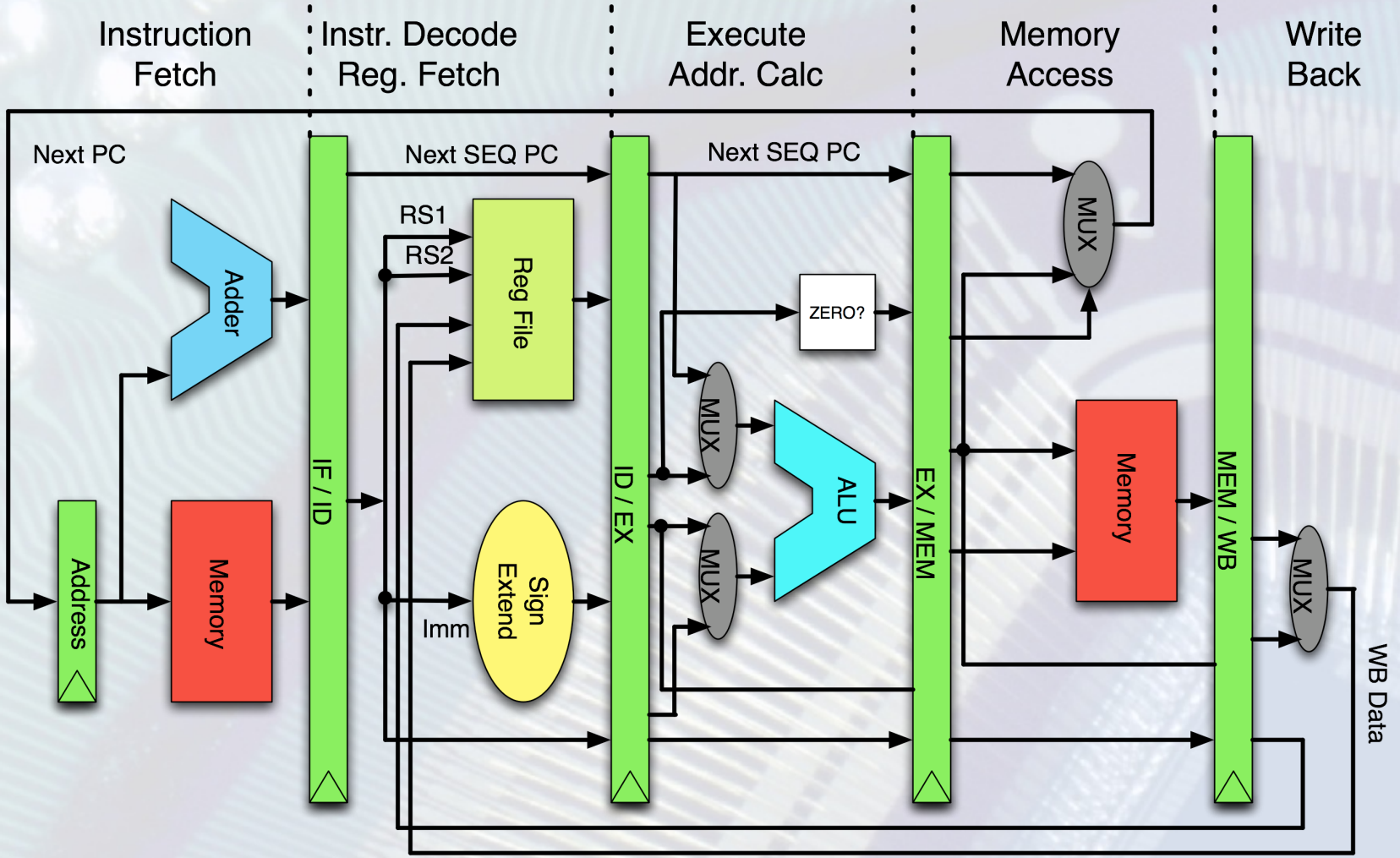
RISC processzorok – Atmel AVR

- **8 bites RISC mikrovezérlő**
- **16 bites cím és 8 bites adat interfész**
- **32 darab 8 bites általános célú regiszter**
  - Indexeléshez: r27:r26 (X), r29:r28 (Y), r31:r30 (Z)
- **Fix 16 bites utasításméret**
  - 2 regisztercímes architektúra
- **Kétfokozatú pipeline**
  - Utasítás lehívás (fetch), végrehajtás (execute)
- **Memória címzési módok**
  - Abszolút
  - Regiszter indirekt (X, Y, Z)
  - Regiszter indirekt poszt inkremens (X+, Y+, Z+)
  - Regiszter indirekt pre dekremens (-X, -Y, -Z)
  - Regiszter indirekt + offszet (Z + offszet)

Pipeline utasítás végrehajtás

- **Az utasítások végrehajtása során az egyes lépésekhez egy-egy pipeline fokozat tartozik**
  - Jó esetben minden órajelciklusban elkezdhető egy újabb utasítás feldolgozása
- **Pl. 3 fokozatú pipeline**
  - Utasítás elővétel (IF, fetch)
    - Programmemória olvasás
    - Beírás az utasítás regiszterbe
    - A programszámláló növelése
  - Utasítás dekódolás (DC, decode)
    - Az elvégzendő művelet meghatározása
    - A műveleti operandusok meghatározása, beolvasása
  - Utasítás végrehajtás (EX, execute)
    - A művelet végrehajtása és az eredmény visszaírása

Pipeline utasítás végrehajtás



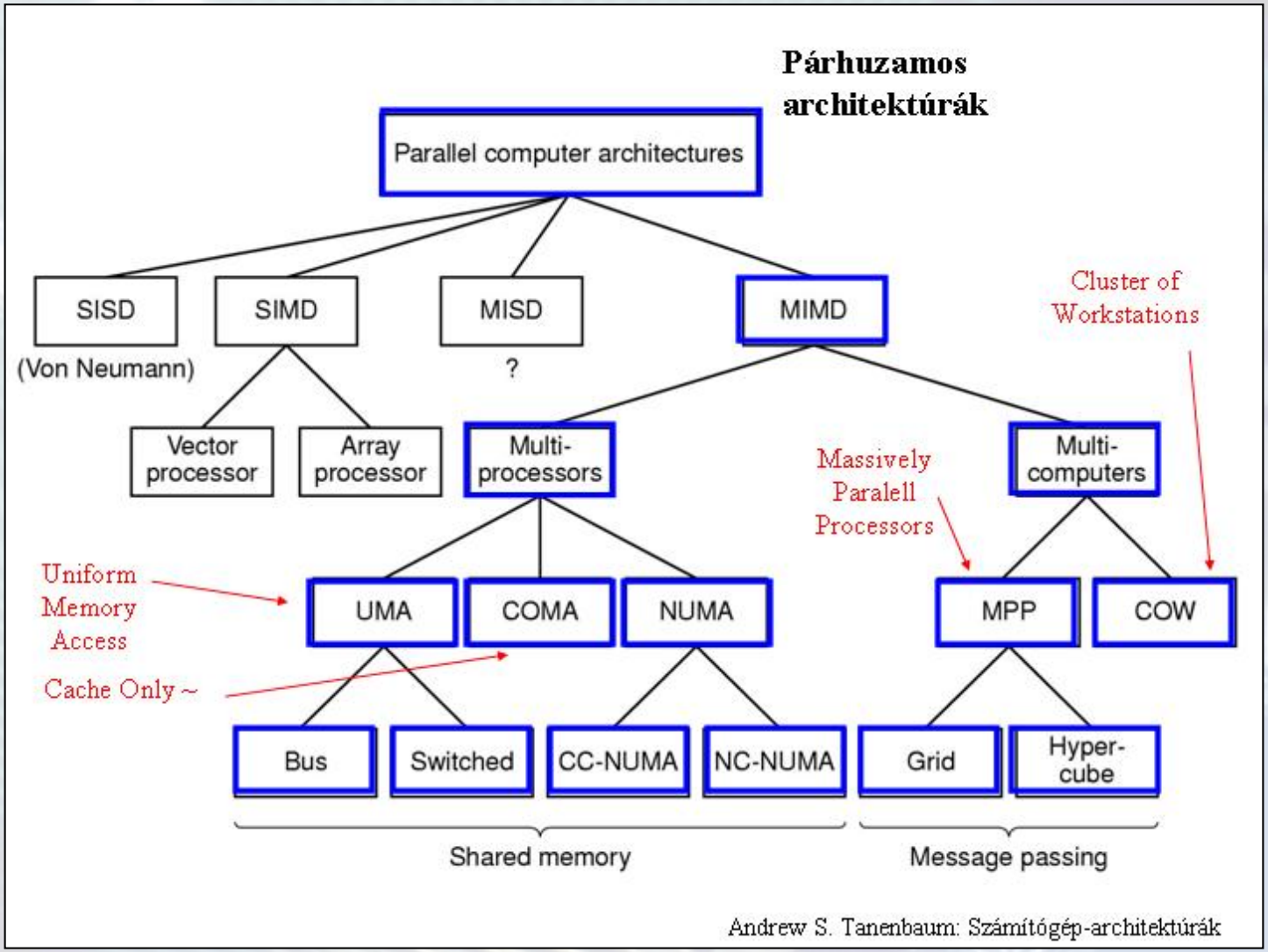
Pipeline utasítás végrehajtás

- **Problémák – külső memória hozzáférés**
  - Az adat rendelkezésre állásáig a pipeline leáll
  - Gyorsítótár (cache) alkalmazása
- **Problémák – adat versenyhelyzet**
  - Az előző eredmény még nincs visszaírva, de operandusa a következő utasításnak
  - Megoldás
    - Az utasítások átrendezése (assembler)
    - A végrehajtás felfüggesztése, amíg az eredmény nem érhető el
    - Adat előrecsatolás (egy pipeline fokozat átugrása)

Pipeline utasítás végrehajtás

- **Problémák – ugrás**
  - A PC új értéket kap, emiatt a már lehívott utasítások érvénytelenek lesznek
  - Megoldás
    - Pipeline kiürítése: kihasználatlan órajel ciklusok
    - Delay slot: a már lehívott utasítás(ok) végrehajtása, programszervezés kérdése (a fordító feladata)
    - Ugrásbecslés: a CPU megpróbálja megjósolni az ugrási címet, helyes becslés esetén nincs veszteség
- **Problémák – megszakítás, kivétel**
  - Nem tudni, hogy mikor történik
  - Pipeline kiürítése az egyetlen lehetőség

Párhuzamos architektúrák

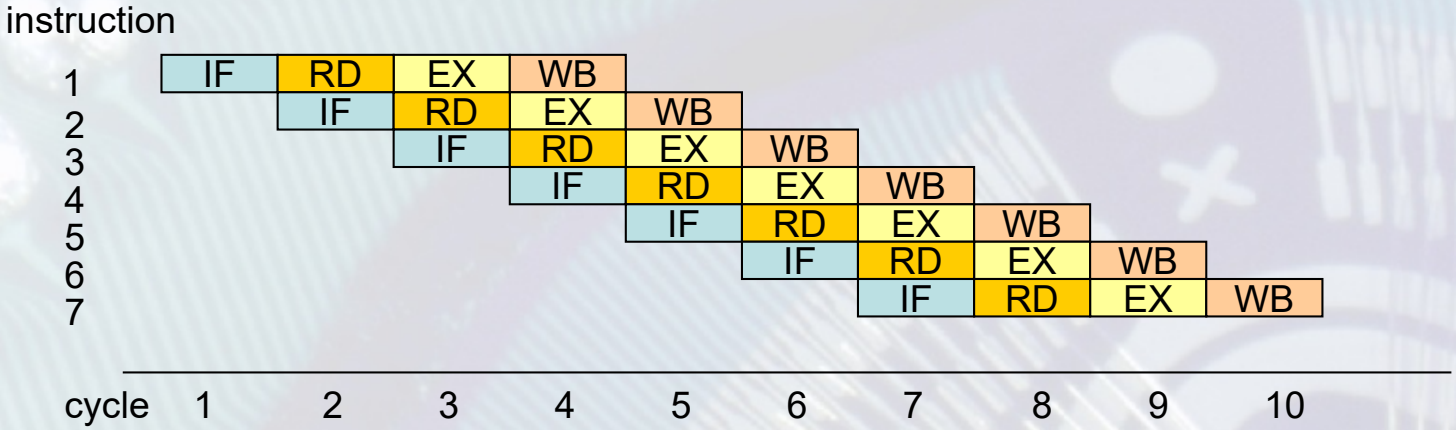


Alapfogalmak

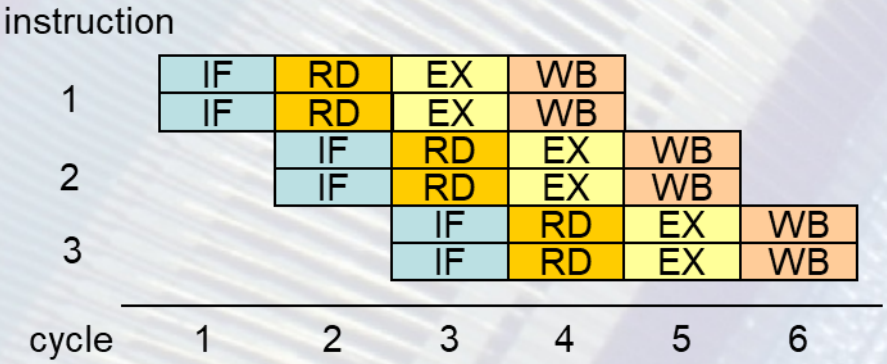
- **Symmetric MultiProcessing**
  - Minden mag látja a teljes memóriát
  - Minden mag teljes hozzáféréssel rendelkezik a memóriához
  - Tipikusan max. ~32 mag
- **Non-Uniform Memory Architecture (NUMA)**
  - Minden processzor saját memóriával rendelkezik, a processzoronkénti sávszélesség jóval nagyobb
  - A memória elérés függ a processzor – memória távolságtól
  - **ccNUMA:** cache coherent NUMA

Alapfogalmak

- **Pipeline végrehajtás**



- **Szuperskalár processzor**
  - Több utasítás lehívása órajelenként (≠ pipeline)



Alapfogalmak

- **In-order utasítás végrehajtás**
  - Az assembly utasítások a kódnak megfelelő sorrendben hajtódnak végre (fetch – wait – execute - write)
- **Out-of-order (OoOE) utasítás végrehajtás**
  - A processzor átrendez(het)i az utasítások sorrendjét
  - Fetch - dispatch (wait) – execute – queue – write
- **Regiszter átnevezés**
  - Több fizikai regiszter van, mint architekturális
  - Megszabadulás a hamis adatfüggőségektől
  - Magasabb utasításszintű párhuzamosság elérése
- **Ugrásbecslés**

Alapfogalmak

- **SISD**
  - Single Instruction Single Data
- **SIMD**
  - Single Instruction Multiple Data
  - Vektor utasításkészletek: Intel MMX, SSEx, AMD 3DNow!, Intel AVX, Power AltiVec, ARM NEON
- **MIMD**
  - Multiple Instruction Multiple Data
  - Pl. GPU
- **VLIW: Very Long Instruction Word**
  - Utasítás szintű párhuzamosítás fordítási időben
  - A fordító próbál optimalizálni és nem a hardver

Alapfogalmak

- **Többszálúság (multi-threading)**
  - ***Temporal MT:*** az egymást követő utasítások származhatnak más szálból
  - ***Simultaneous MT:*** a processzor feldolgozó egységei ugyanazon órajelben más-más szállal foglalkoznak (szuperskalár)
  - ***Multi-core:*** több, teljes processzor mag egy szilíciumon

Alapfogalmak – Cache

- **Miért kell?**
  - DRAM hozzáférés tulajdonságai
  - Kis késleltetés
- **Cache-ben:**
  - Tárolt memória cím (TAG) – cache találat megállapítás
  - Adat – cache line
- **Memória írás**
  - Write-through: minden cache írás hatására frissül a RAM-ban az adat
  - Write-back: adat kiírás a cache frissítése esetén
- **Cache koherencia**
  - Több CPU mag, DMA képes HW (cache invalidálás)

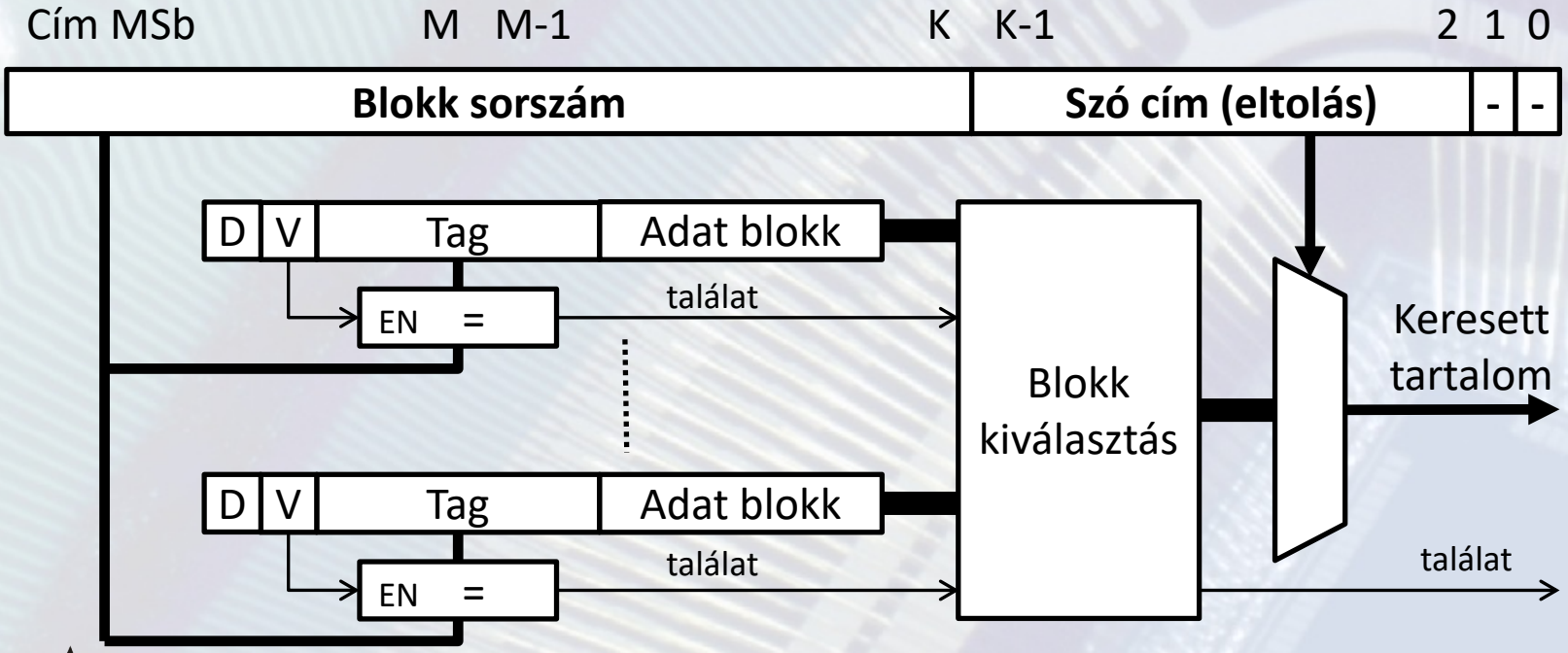
Alapfogalmak – Cache

- **Az adatok tárolása 2^K byte méretű blokkokban (cachce line) történik, a cache mérete 2^M byte (M>K)**
  - A cím [K-1:0] bitjei adják a blokkon belüli eltolást
  - A cím felső bitjei adják a blokk sorszámát
- **Minden blokkban az adatok mellett tárolni kell még**
  - Tag: a memória hányadik blokkja tartozik a tárolt adathoz (a cím felső bitjei)
  - Valid bit: a blokk érvényességének jelzése
  - Dirty bit: történt-e írás erre a blokkra

Alapfogalmak – Cache

Teljesen asszociatív gyorsítótár (az egyik véglet)

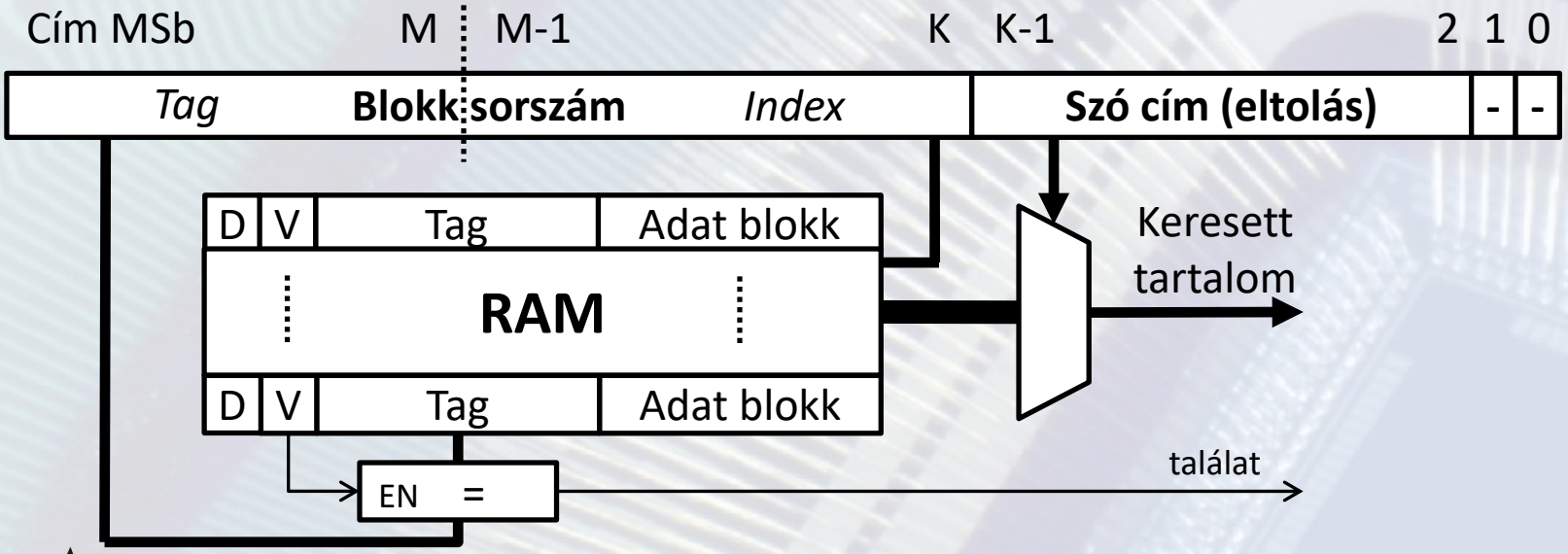
- **A memória blokkok bárhová elhelyezhetők a cache-ben**
- **Tartalom szerint címezhető memória (CAM)**
- **Nagy erőforrás- és energiaigény: sok széles komparátor**



Alapfogalmak – Cache

Direkt leképzésű gyorsítótár (a másik egyik véglet)

- **A memória blokkok csak egy helyre kerülhetnek**
  - Pl. a blokk sorszám alsó bitjei (index) döntik ezt el
- **Jóval kisebb erőforrásigény: memória és 1 komparátor**
- **Hátrány: gyakori cache miss nem megfelelő program vagy adatpuffer szervezésnél a működés jellege miatt**



Alapfogalmak – Cache

N-utas asszociatív gyorsítótár (a kettő között)

- **A memória blokkok N helyre kerülhetnek (tipikusan N=2-16)**
  - Az index bitek egy N elemű halmazt jelölnek ki
- **Erőforrásigény: N db memória és N db komparátor**
- **Ritkább cache miss, mérsékelt komplexitás és fogyasztás**