



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK



Budapest,
2025.02.11.

SZÁMÍTÓGÉP ARCHITEKTÚRÁK

Bevezetés, Információfeldolgozási modellek

Horváth Gábor, Belső Zoltán

BME Hálózati Rendszerek és Szolgáltatások Tanszék
ghorvath@hit.bme.hu, belso@hit.bme.hu

- Kommunikációs csatornák:
 - Tárgy weblap: <http://www.hit.bme.hu/~ghorvath/szgarch/>
 - **Tanszéki Moodle:** <https://moodle.hit.bme.hu>
 - Teams csoportot nem használjuk
- Oktatási segédanyagok a tárgy weblapján:
 - Előadás fóliák
 - Gyakorlat feladatlapok
 - Jegyzet
 - Videók
 - Online gyakorló app

- **Követelmények**
 - **Számonkérések**
 - ZH: 10. hét, hétfő-kedd 18-20, HSZK + Moodle
 - Lesz próba ZH
 - Feladat típusok:
 - Igaz – hamis – passz
 - “Összekötös”
 - Rövid válaszok
 - Numerikus példák
 - Sikeresség: 40%
 - Egy szint fölött beleszámít a vizsgába
 - **Gyakorlat jelenlét**
 - Min. 70% → 4 jelenlét kötelező
- **IMSc**
 - plusz tananyag

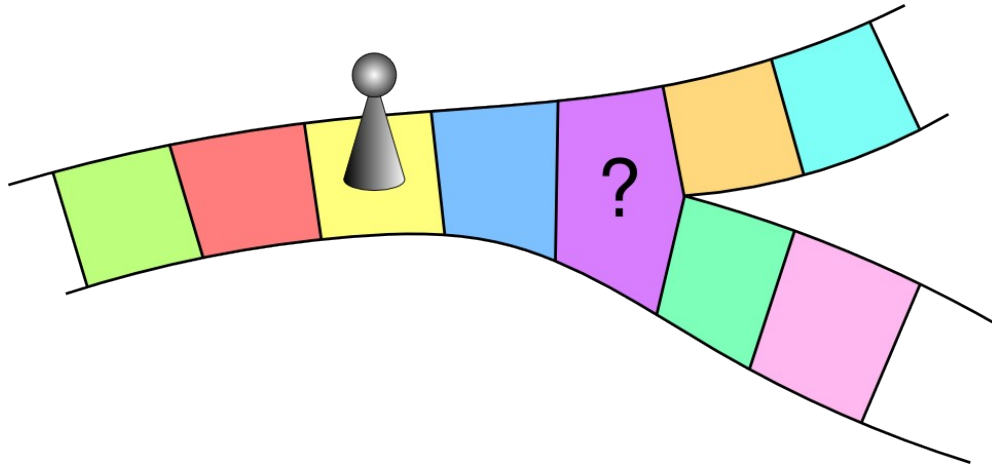
- Megértjük, hogy
 - ...hogyan épül föl egy modern számítógép
 - ...hogyan épül föl egy modern processzor
- Mire jó ez nekünk:
 - Aki ismeri a hardvert,
 - a számítógépet nem fekete doboznak tekinti
 - hatékonyabb programot ír
 - ismeri a kompromisszumokat



Információfeldolgozási modellek

- Hogyan fogalmazzuk meg a problémát egy számítógépnek?
 - Vezérlésáramlásos modell
 - Adatáramlásos modell
 - Igényvezérelt modell

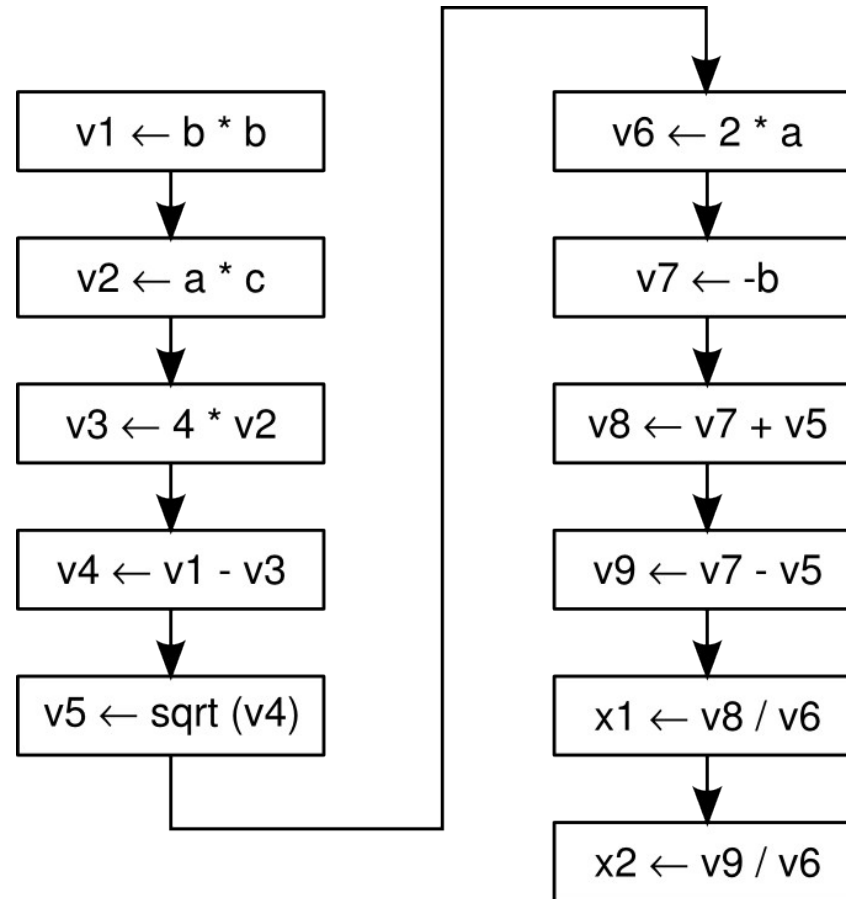
- Utasítások sorrendje: vezérlő token szabja meg



- Vezérlő token mozgása:
 - Implicit (egy lép előre)
 - Explicit (goto, szubrutin, return, stb.)
- Utasítások közötti adatcsere: közös memória
- Program leírása: folyamatábra

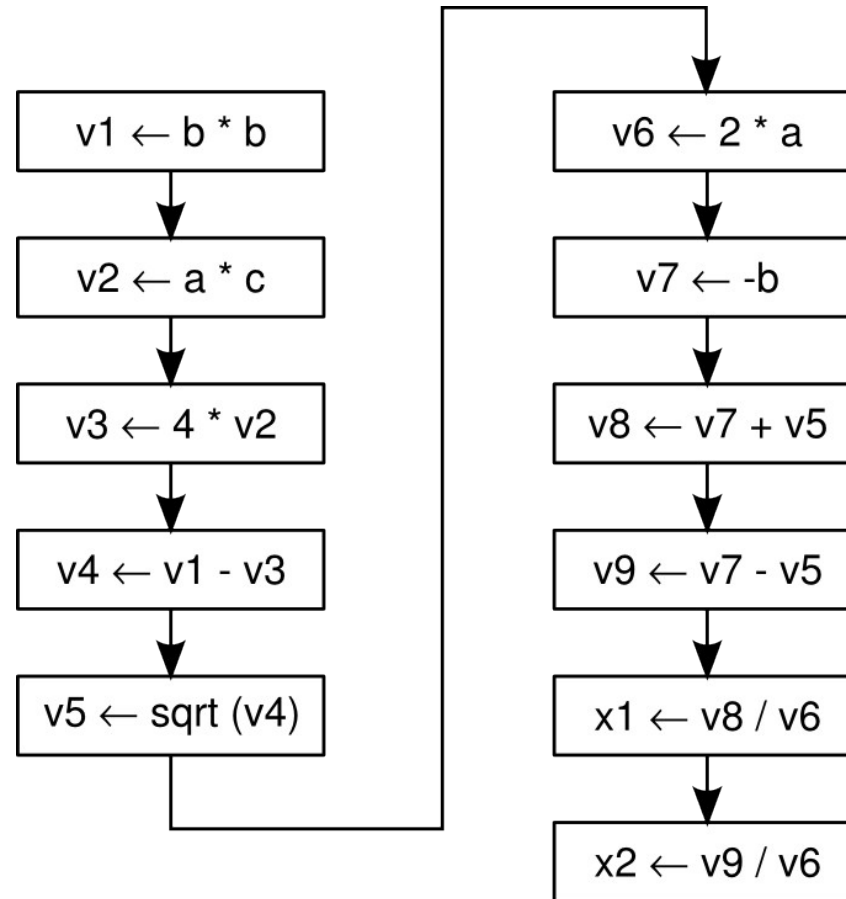
- Példa: másodfokú egyenlet gyökei

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



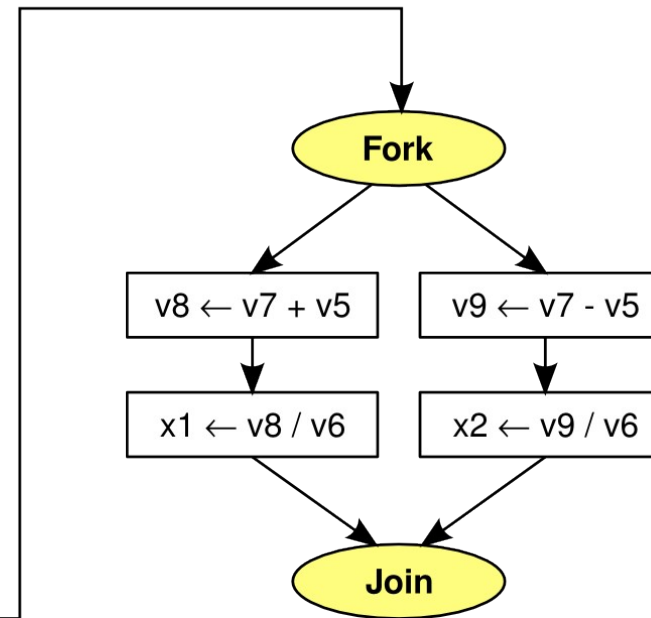
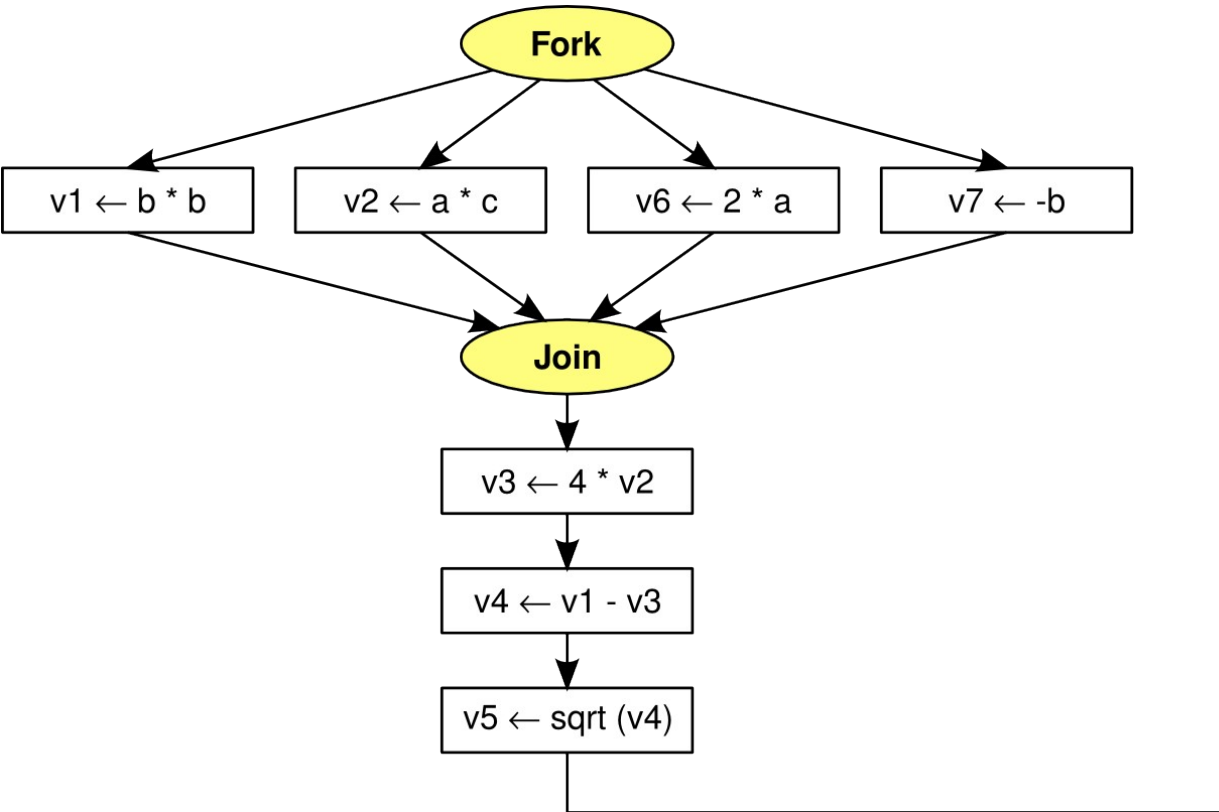
- Párhuzamosság?

- $v1, v2, v6, v7$
- $v8, v9$
- $x1, x2$



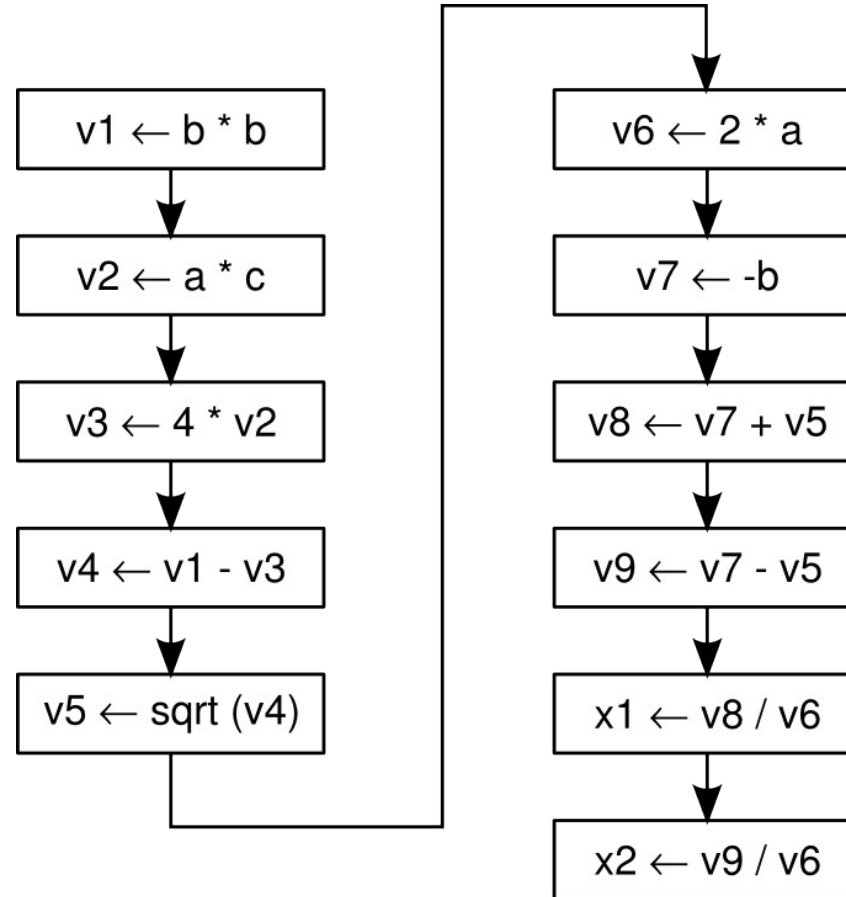
- Párhuzamosság Fork/Join primitívekkel

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

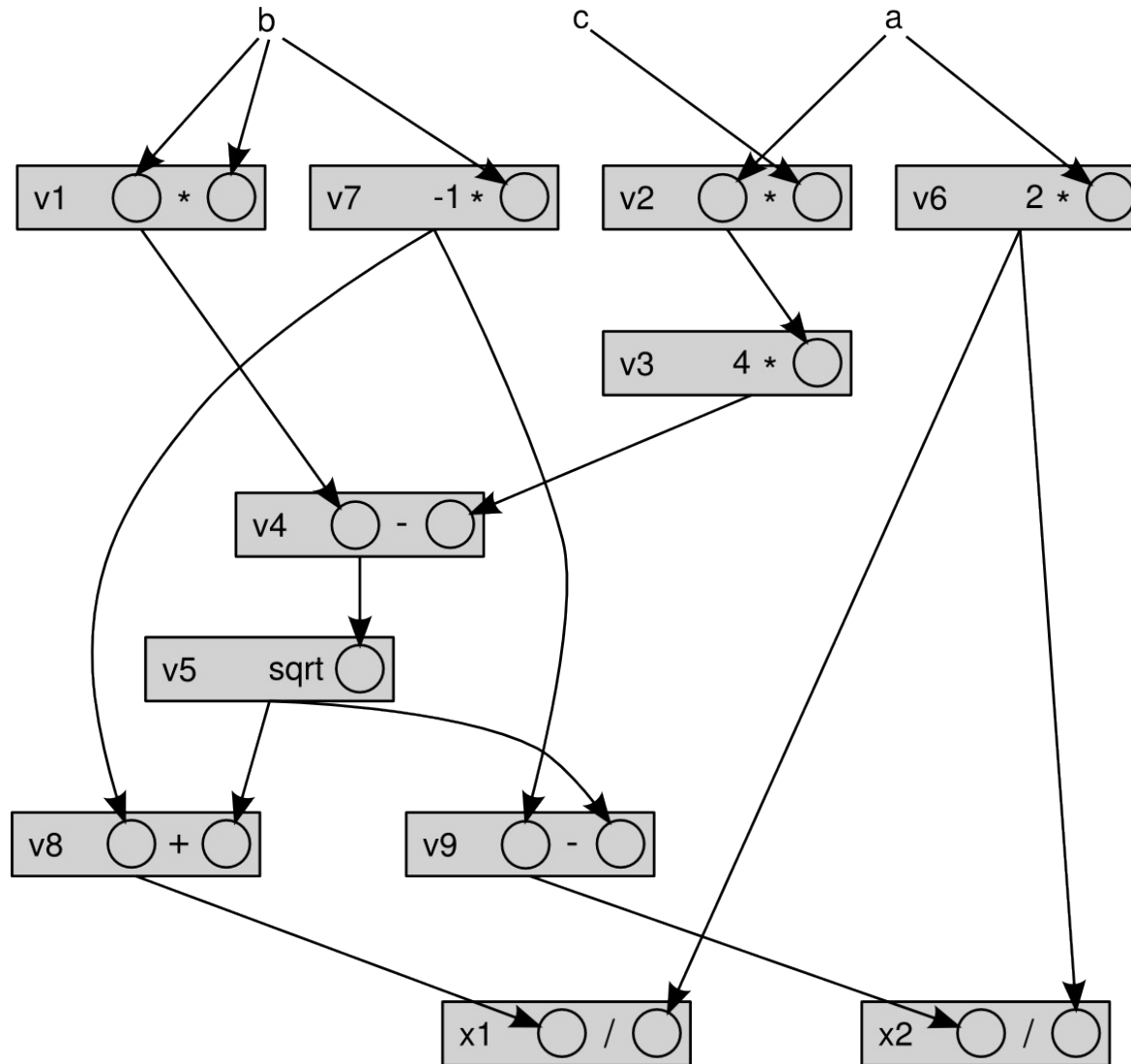


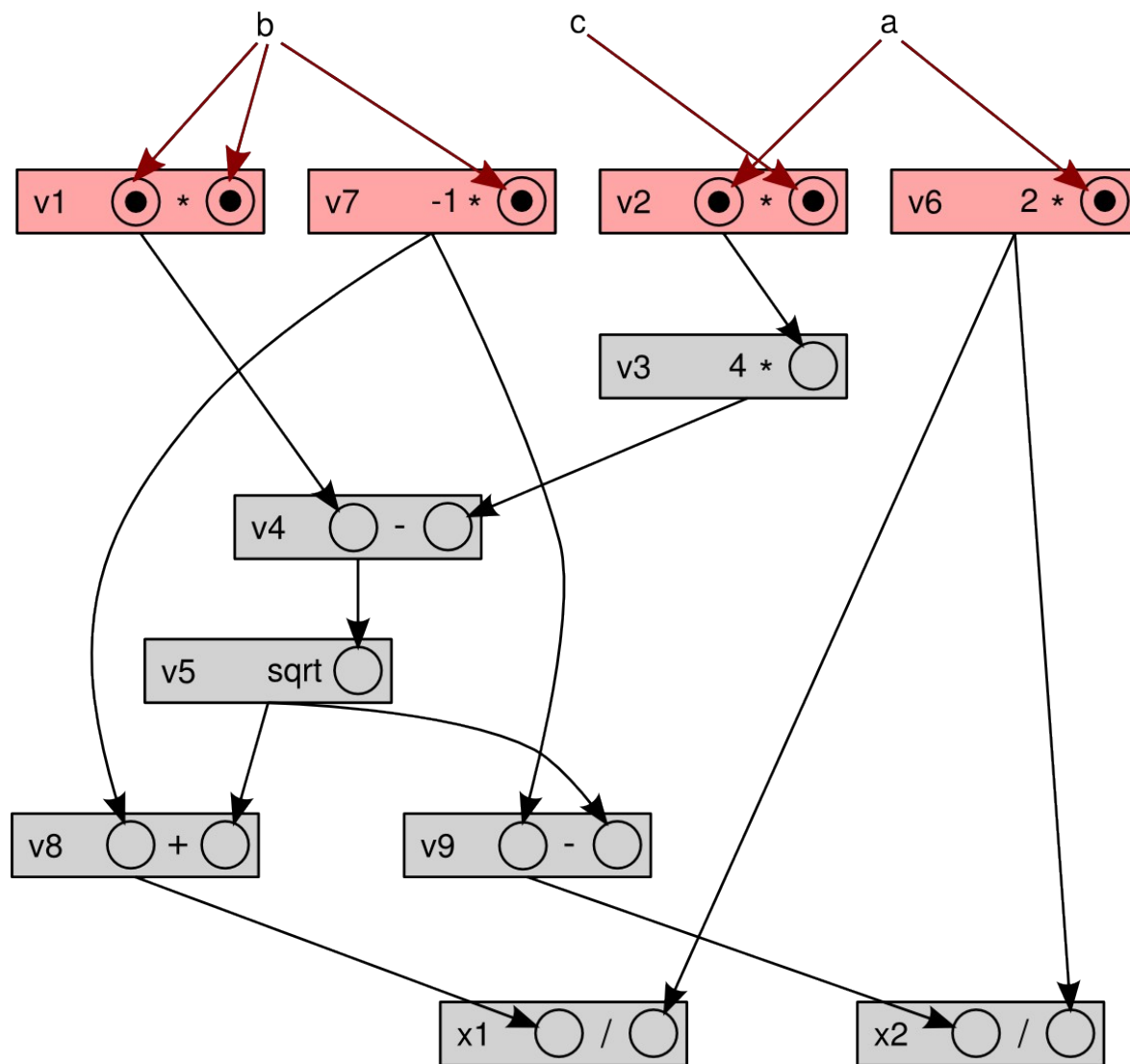
- Értékelés
 - Párhuzamosítási lehetőségeket
 - nem ismeri fel automatikusan,
 - nem képes kiaknázni.
 - Rendkívül elterjedt
 - Példa: Neumann architektúra
 - Vezérlő token: utasításszámláló
 - Közös memória: regiszterek + rendszermemória

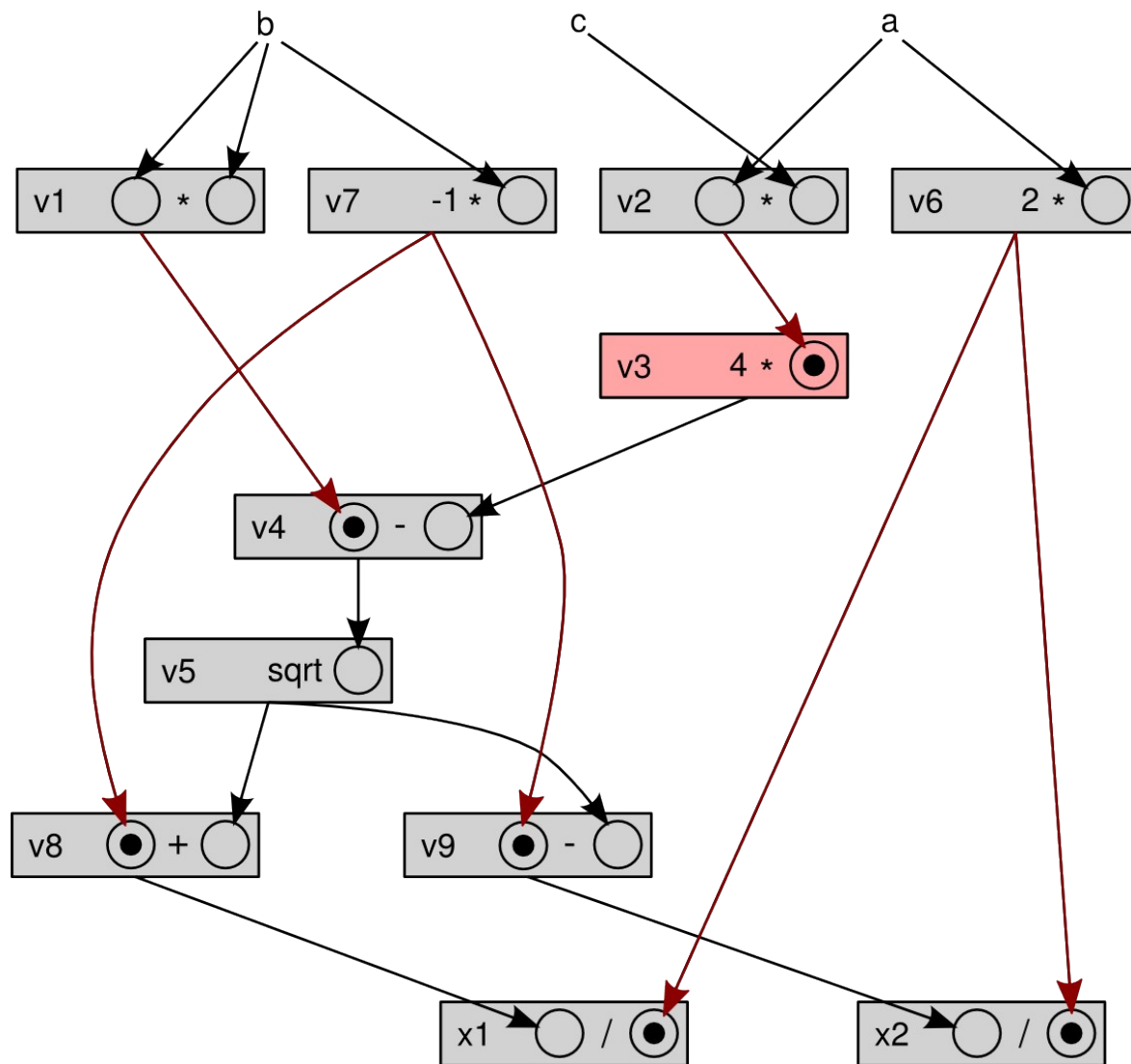
- Honnan tudtuk, hogy ezek párhuzamososíthatók?
 - $v1, v2, v6, v7$
 - $v8, v9$
 - $x1, x2$
- Függőségi analízis!

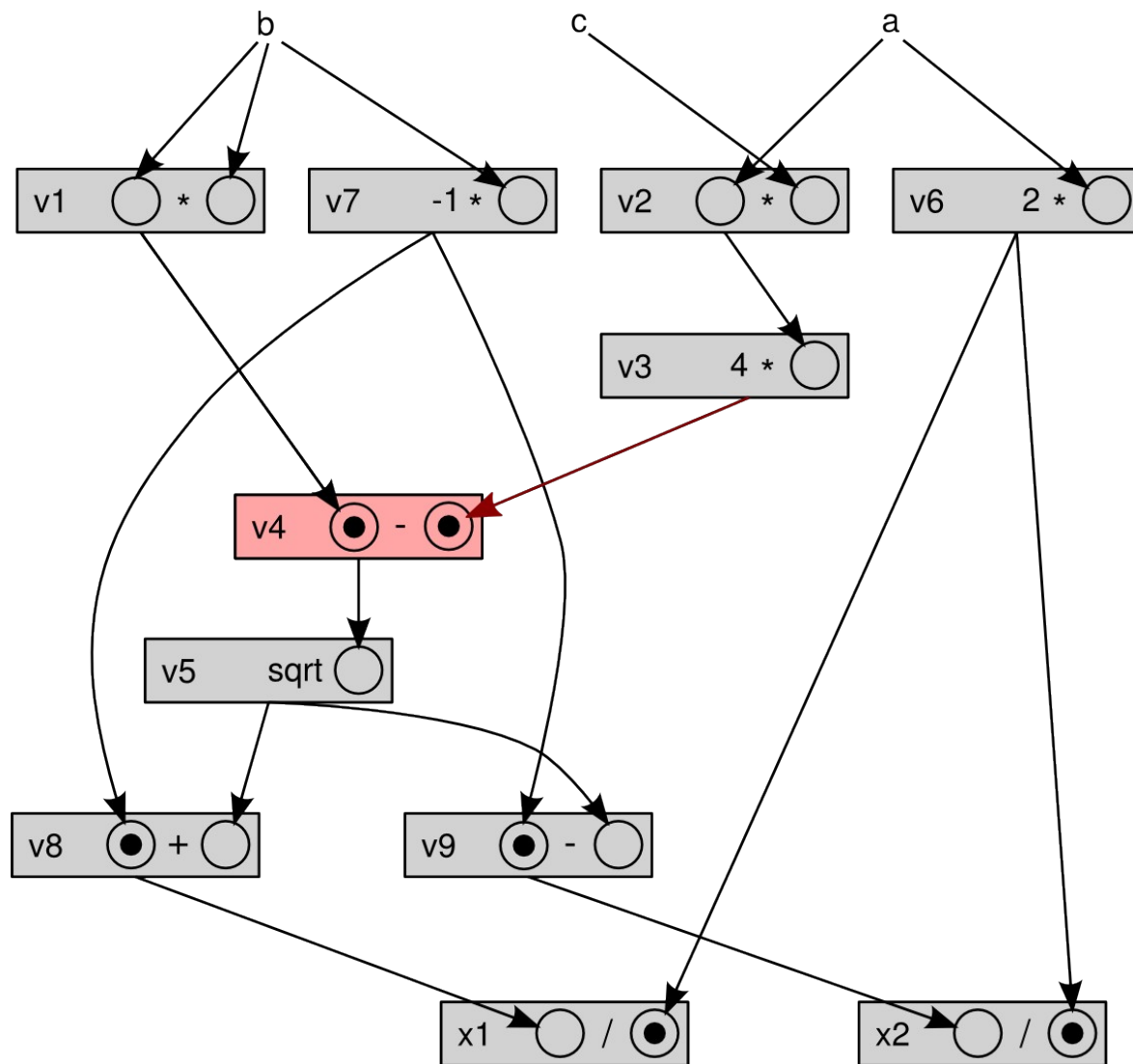


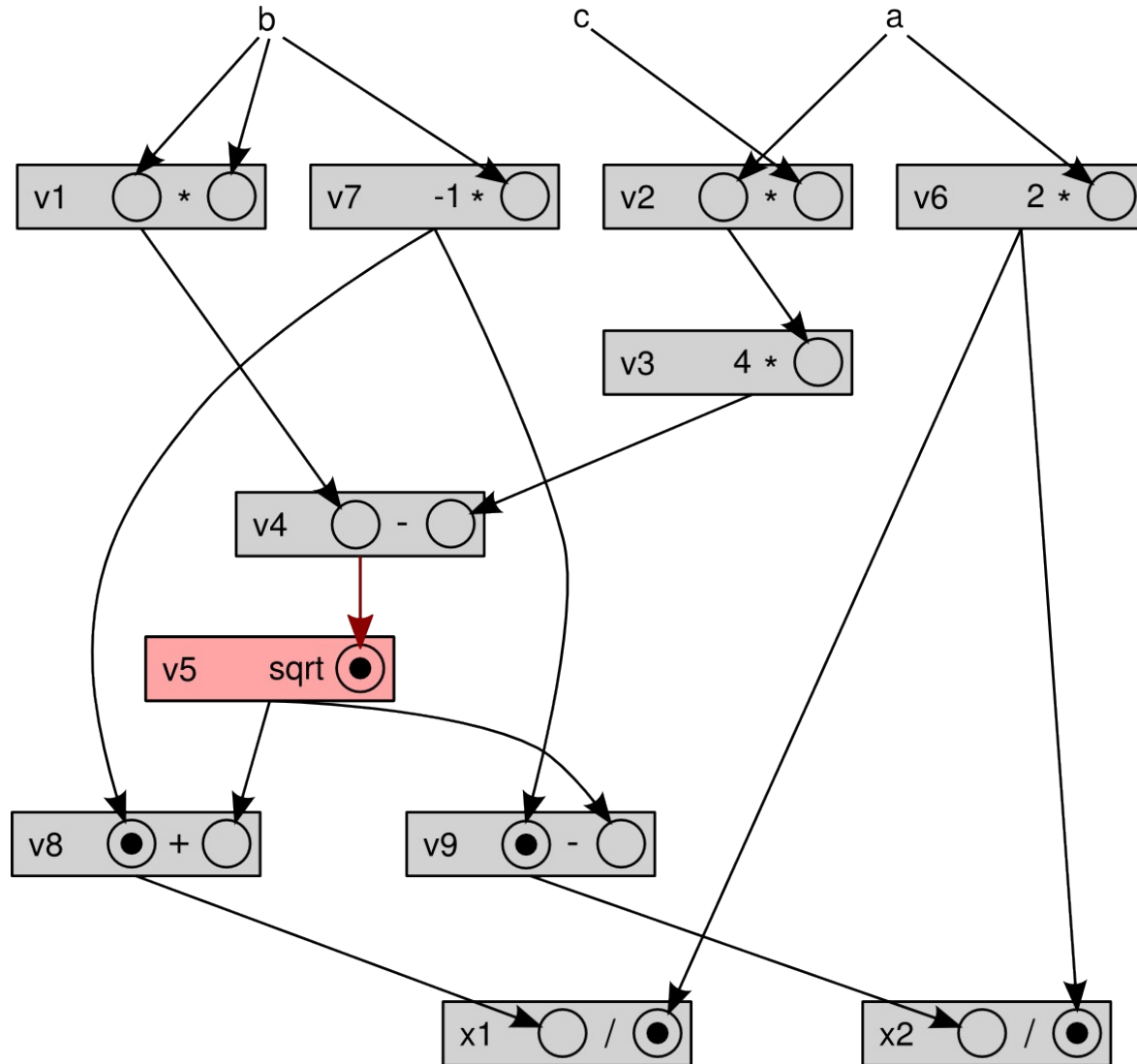
- Nincs vezérlő token
- Nincs közös memória
- Vezérlő token helyett:
 - Utasítások végrehajtása automatikus:
 - ha minden operandus megvan
- Közös memória helyett:
 - Részeredményeket közvetlenül adják át egymásnak az utasítások
- Program leírása: egy precedenciagráf
- Példa: másodfokú egyenlet gyökei

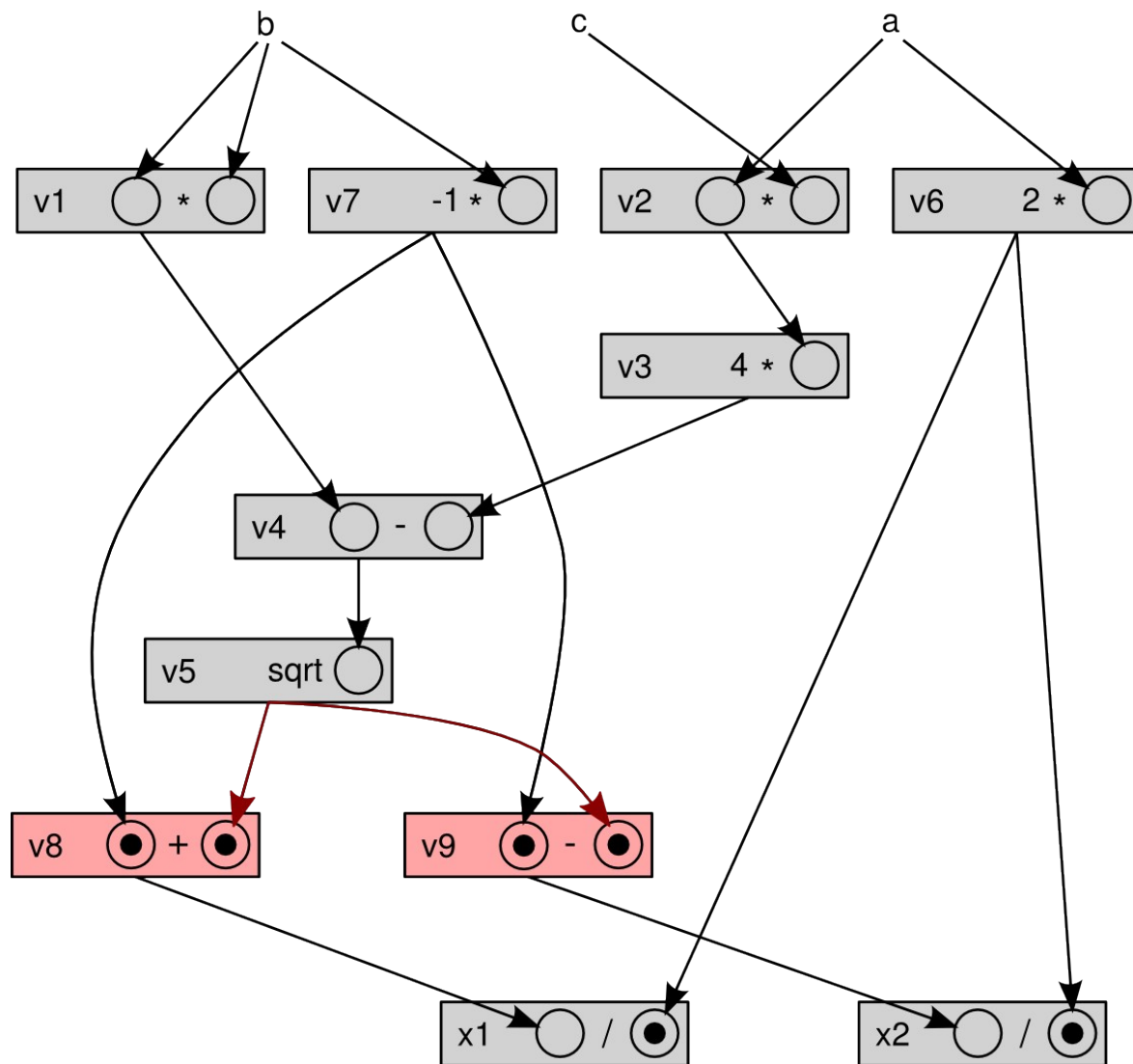


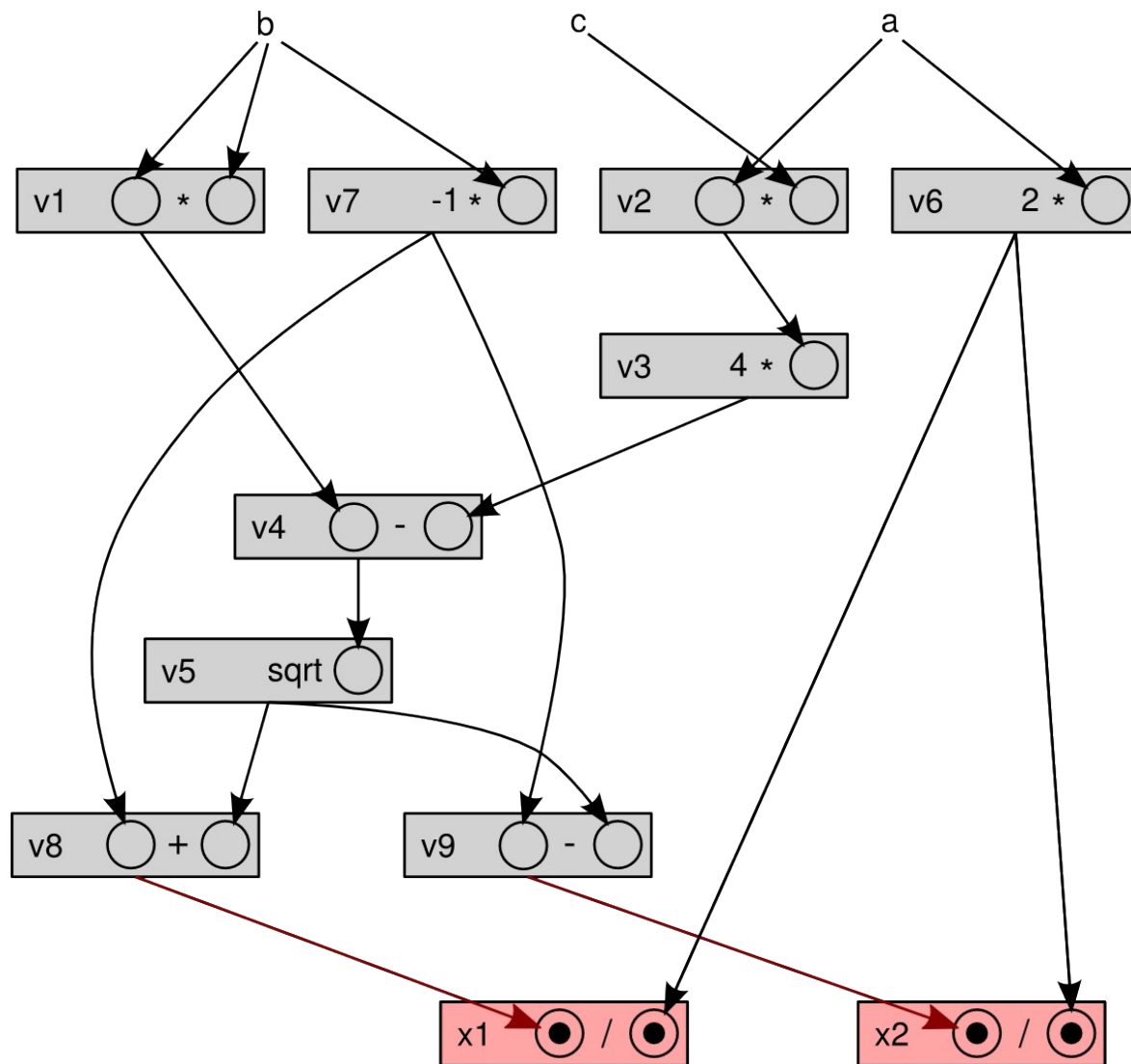






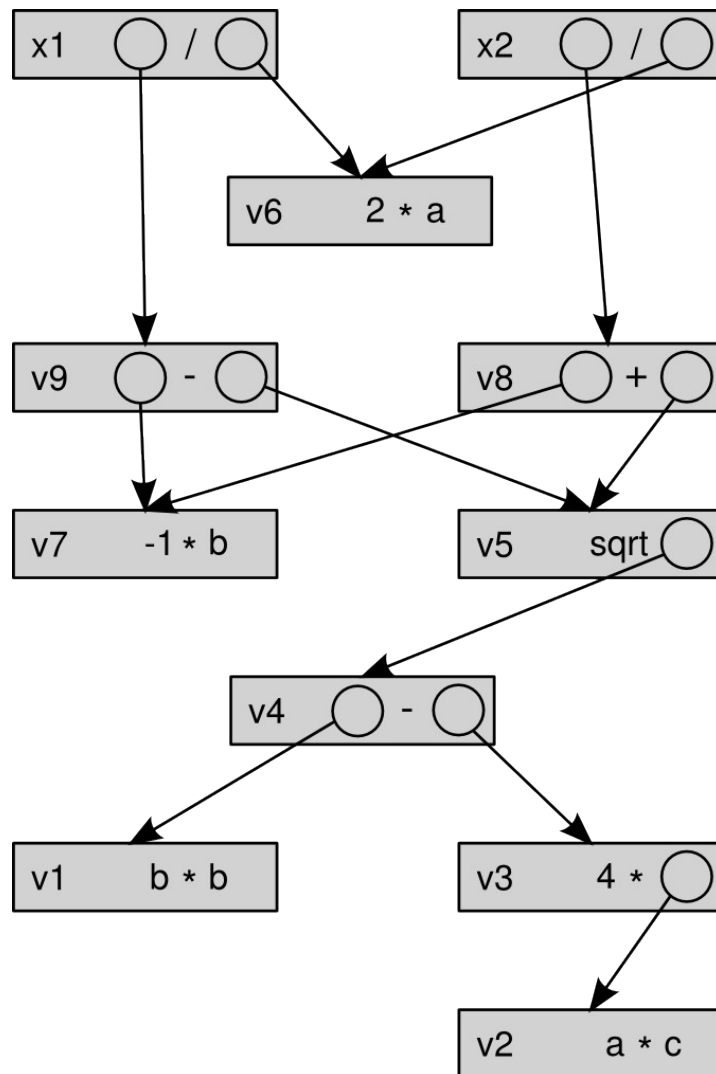






- Értékelés
 - Párhuzamosságot automatikusan felderíti
 - Tisztán adatáramlásos számítógép nem terjedt el
 - De az elv igen:
 - Táblázatkezelők
 - Out-of-order utasításvégrehajtás
 - Adatfeldolgozó rendszerek (pl. Apache Pig)

- Utasítások végrehajtása automatikus:
 - ha szükség van az eredményre
- Összehasonlítva, utasítások végrehajtási ideje:
 - Vezérlésáramlásos: ha odaér a token
szájbarágós
 - Adatáramlásos: ha minden operandus megvan
szorgalmas
 - Igényvezérelt: ha szükség van az eredményre
lusta
- Példa: másodfokú egyenlet gyökei



x1 $\bigcirc / \bigcirc$

x2 $\bigcirc / \bigcirc$

v6 $2 * a$

v9 $\bigcirc - \bigcirc$

v8 $\bigcirc + \bigcirc$

v7 $-1 * b$

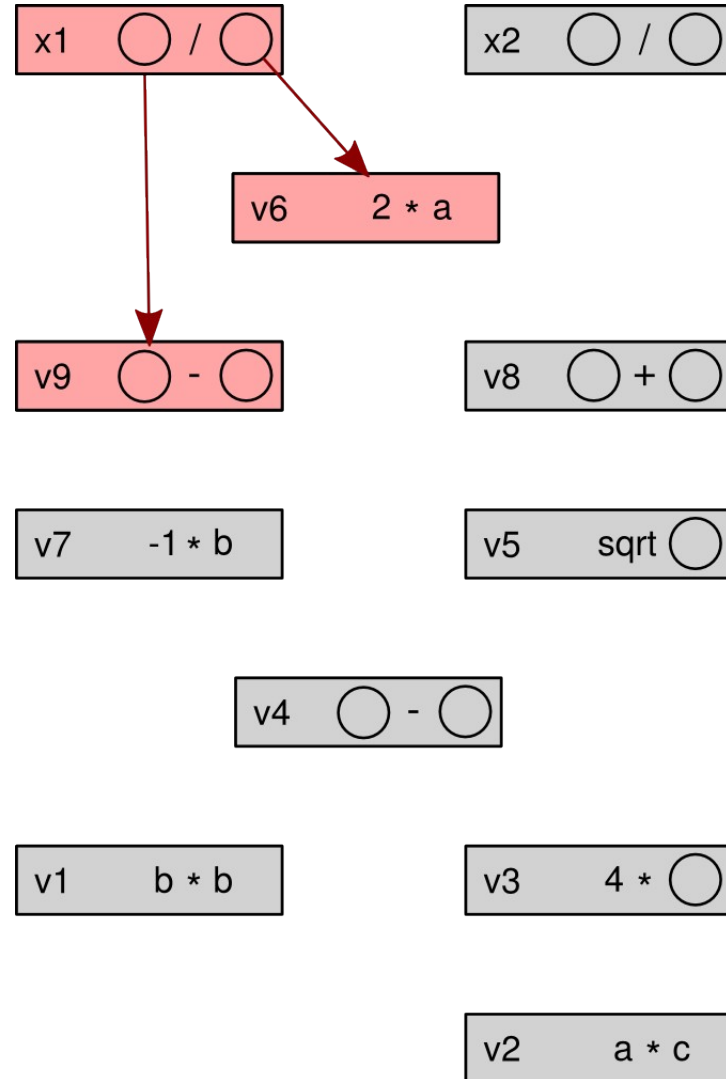
v5 $\text{sqrt } \bigcirc$

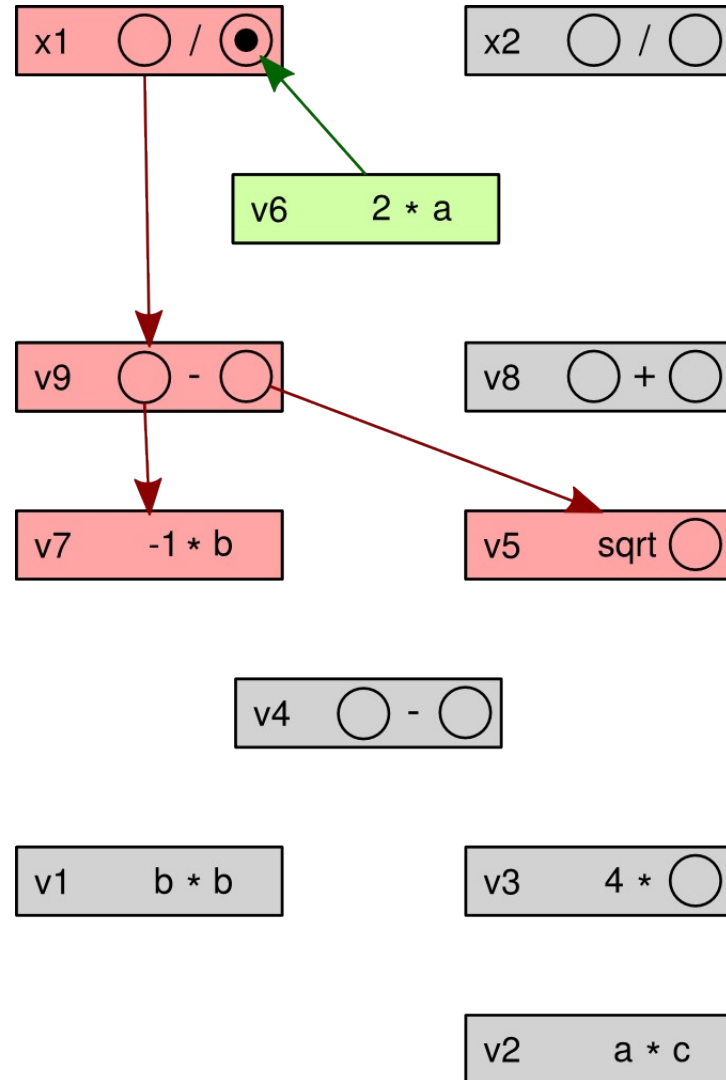
v4 $\bigcirc - \bigcirc$

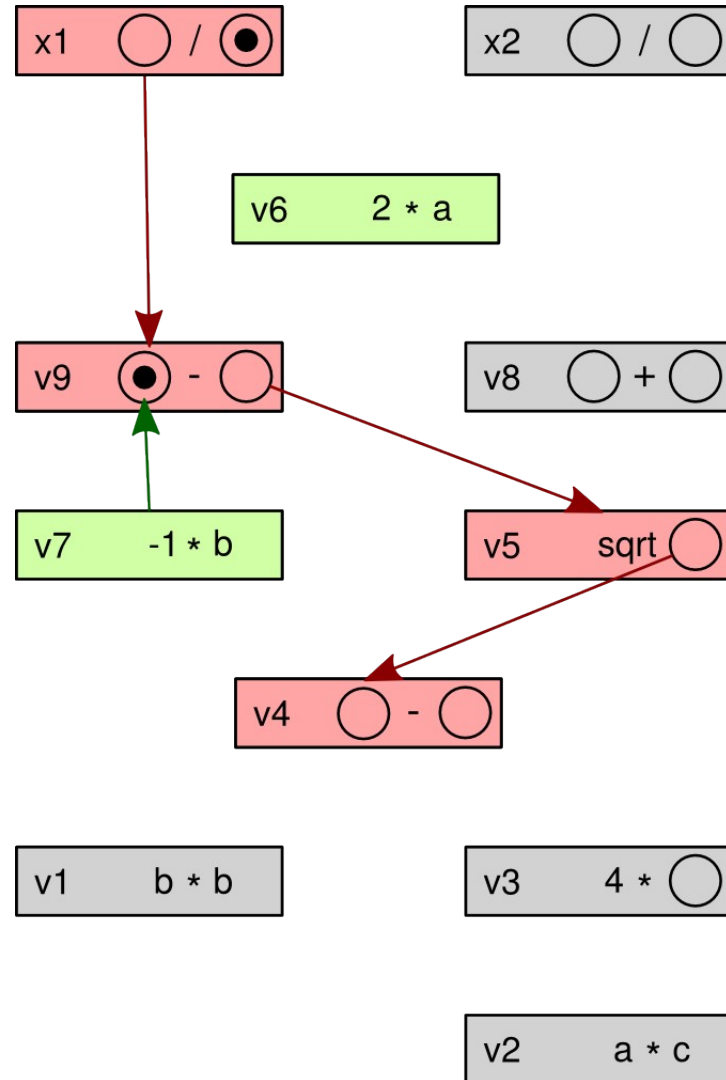
v1 $b * b$

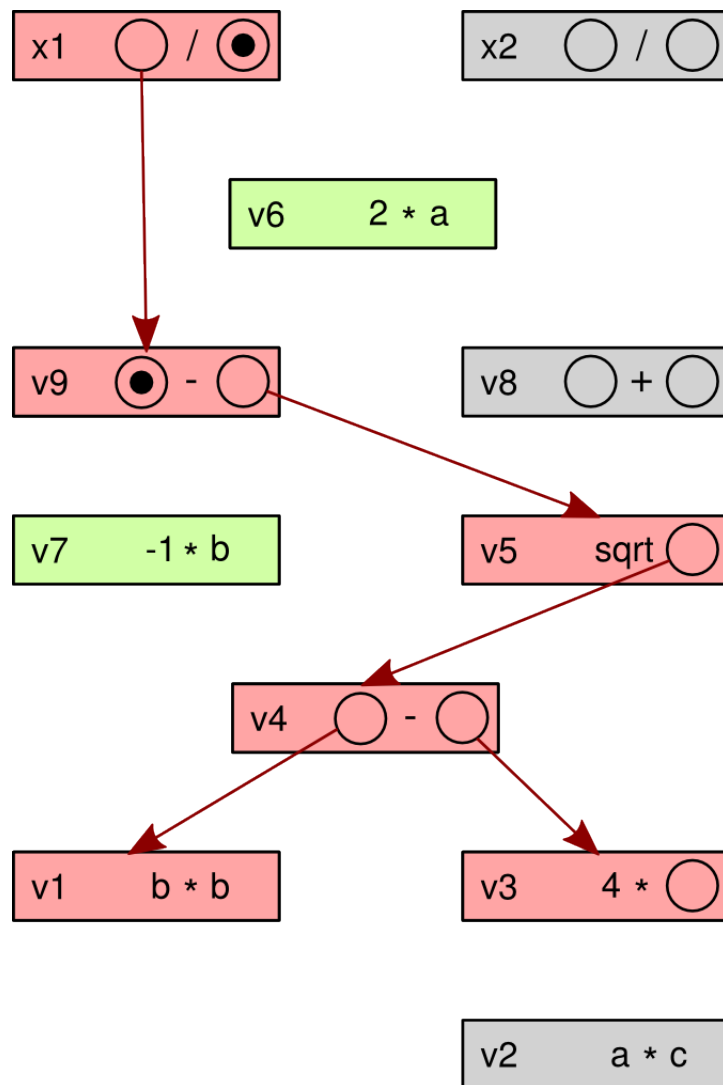
v3 $4 * \bigcirc$

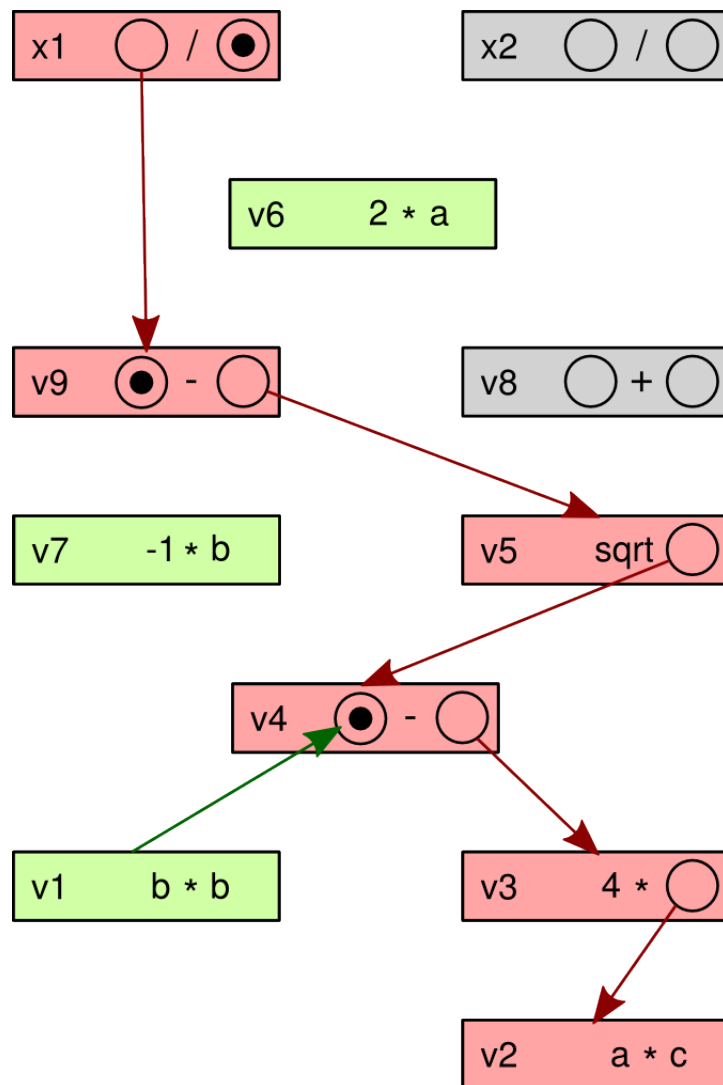
v2 $a * c$

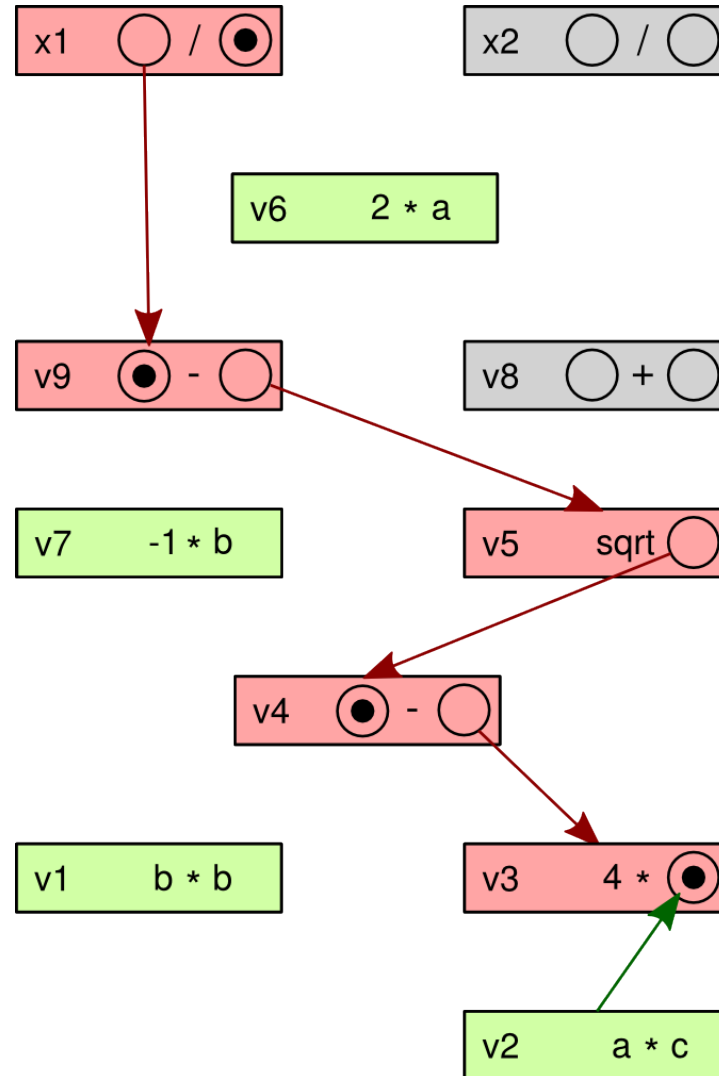


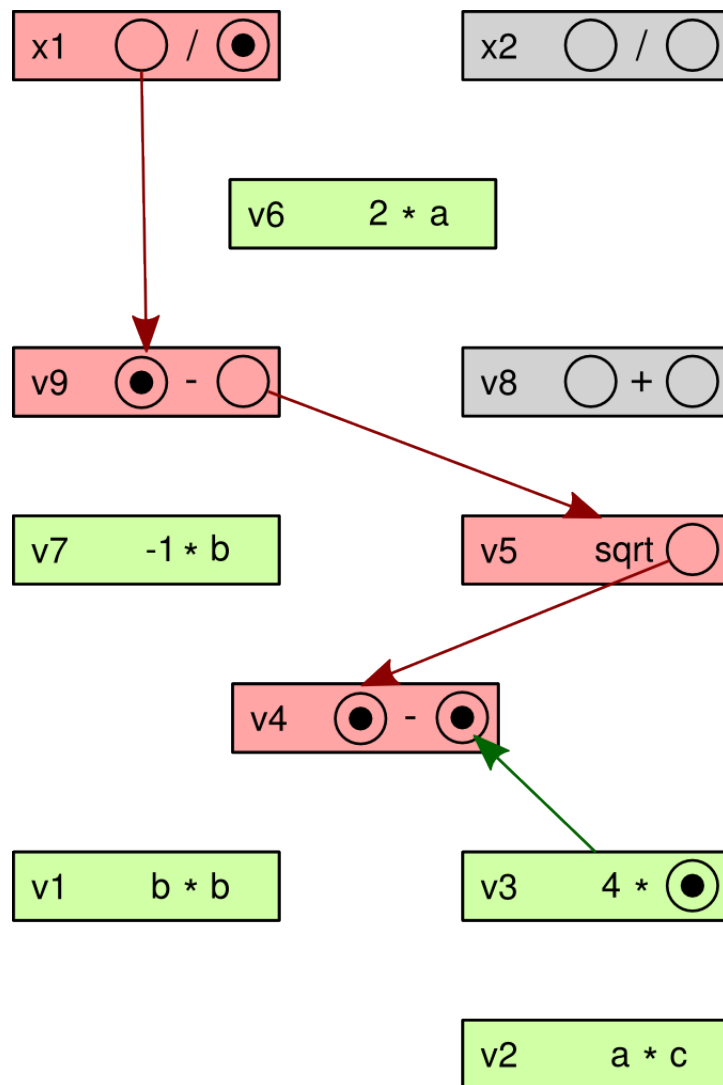


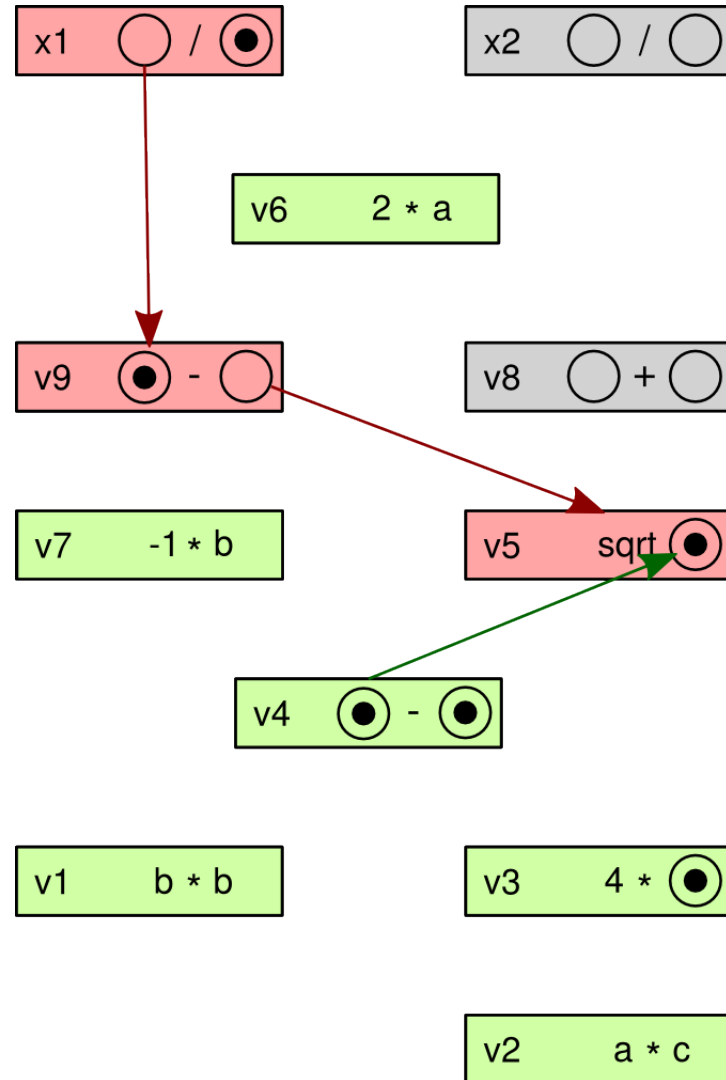


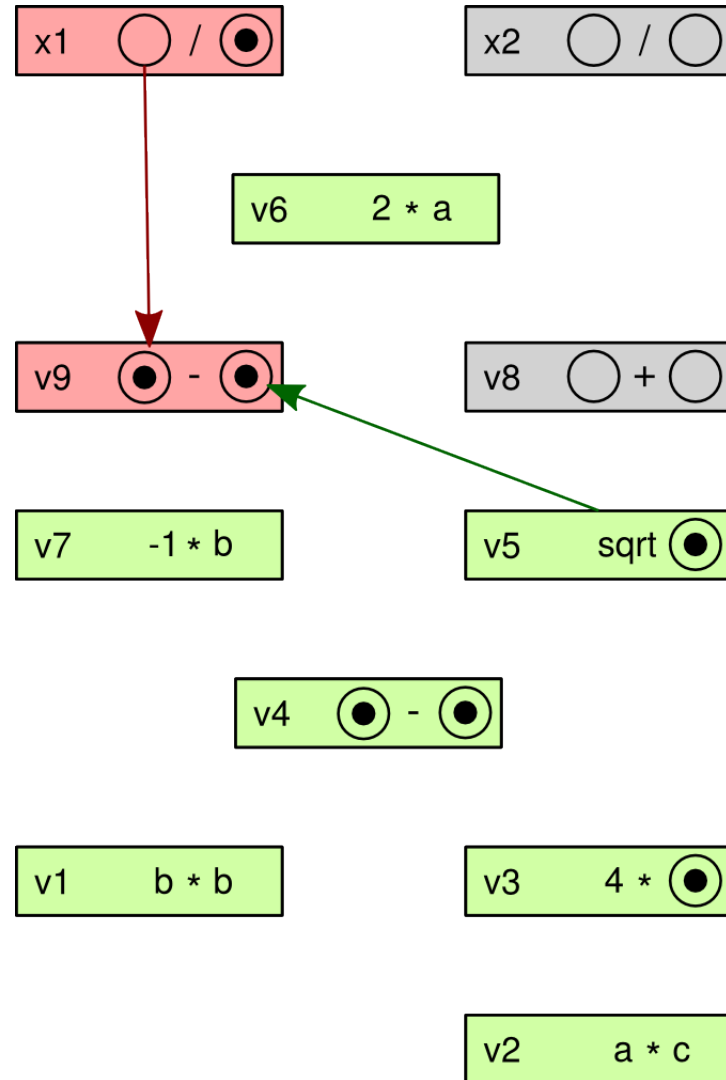


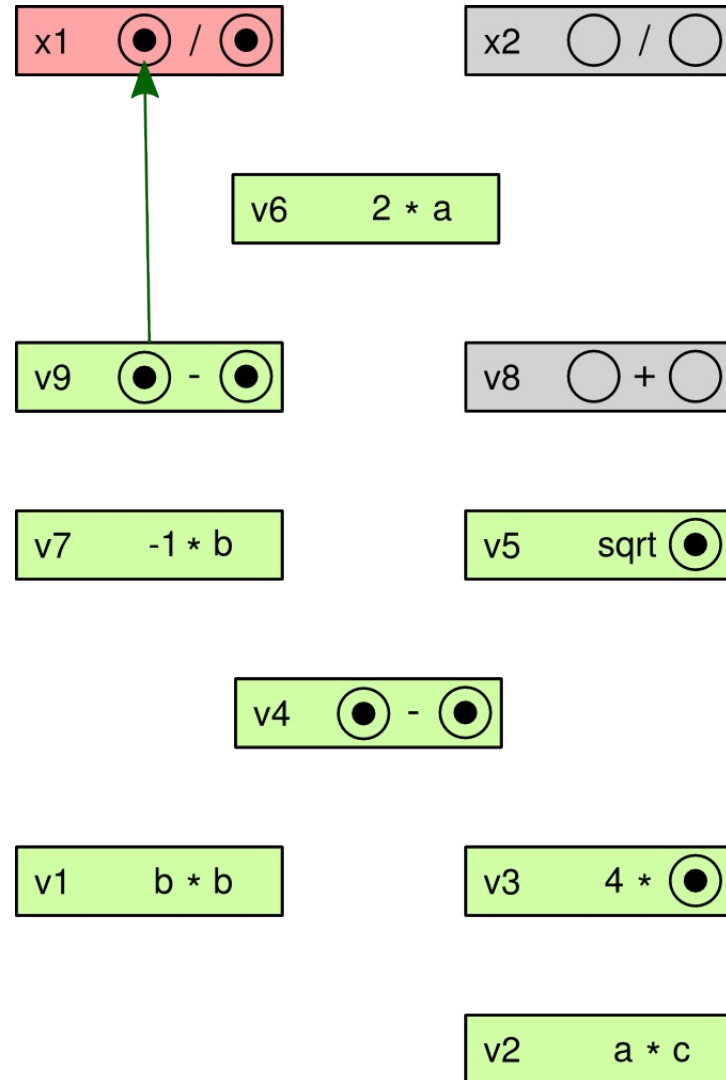


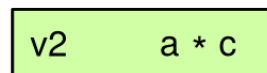
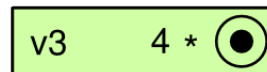
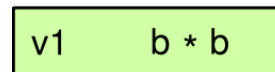
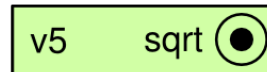
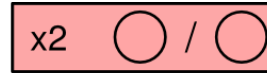


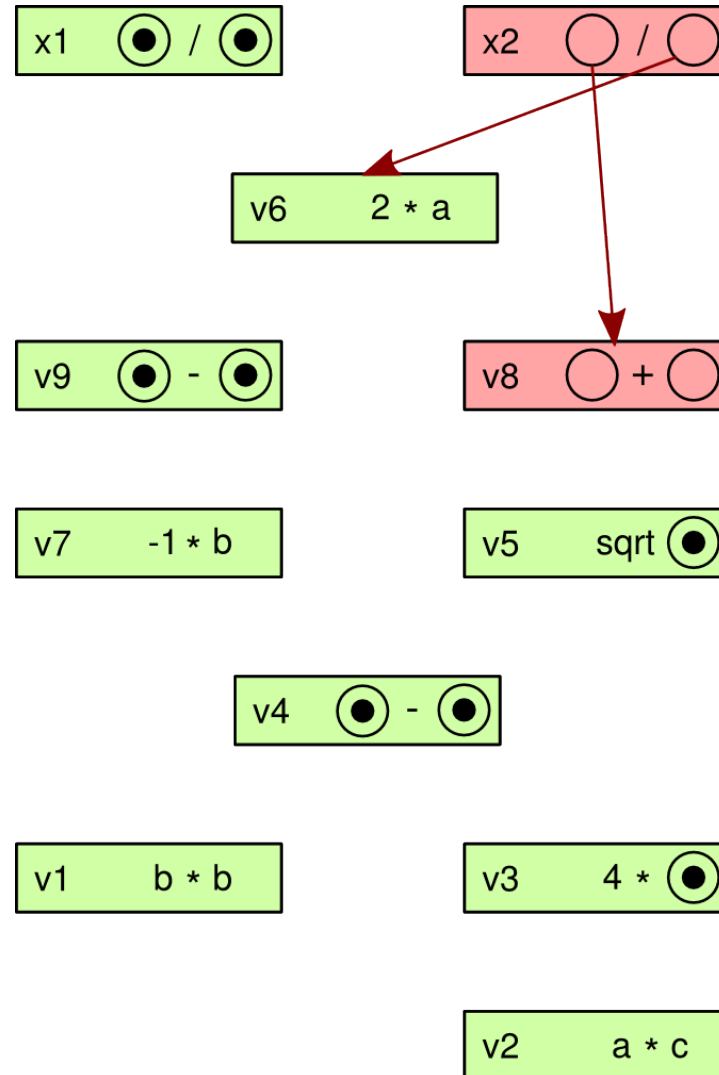


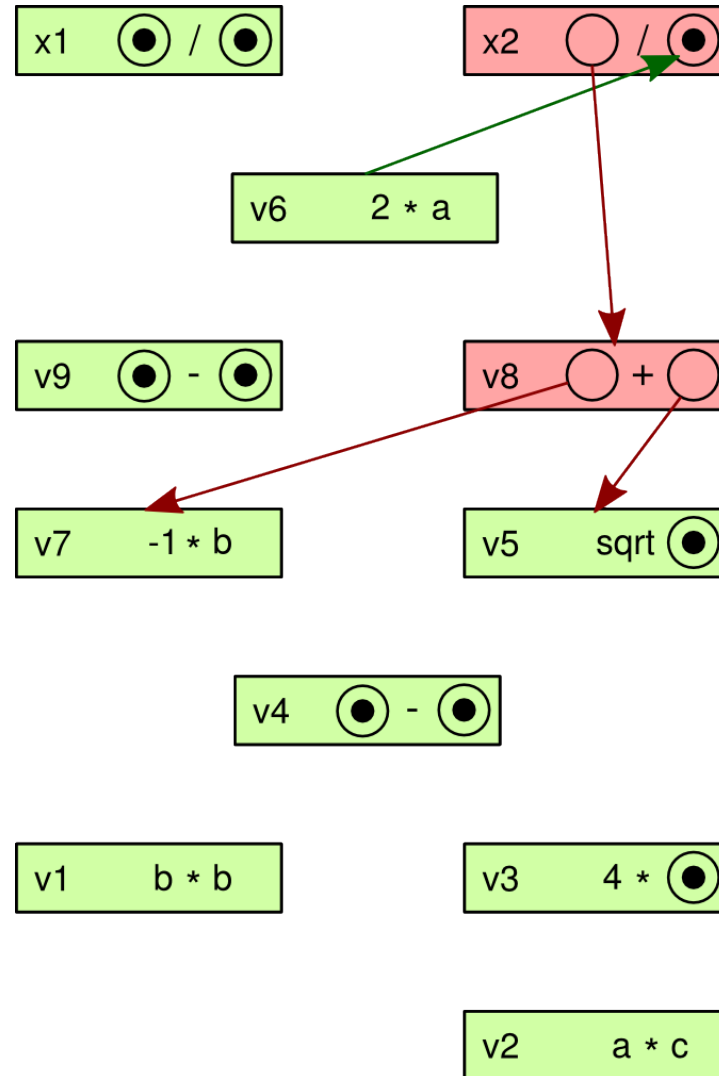


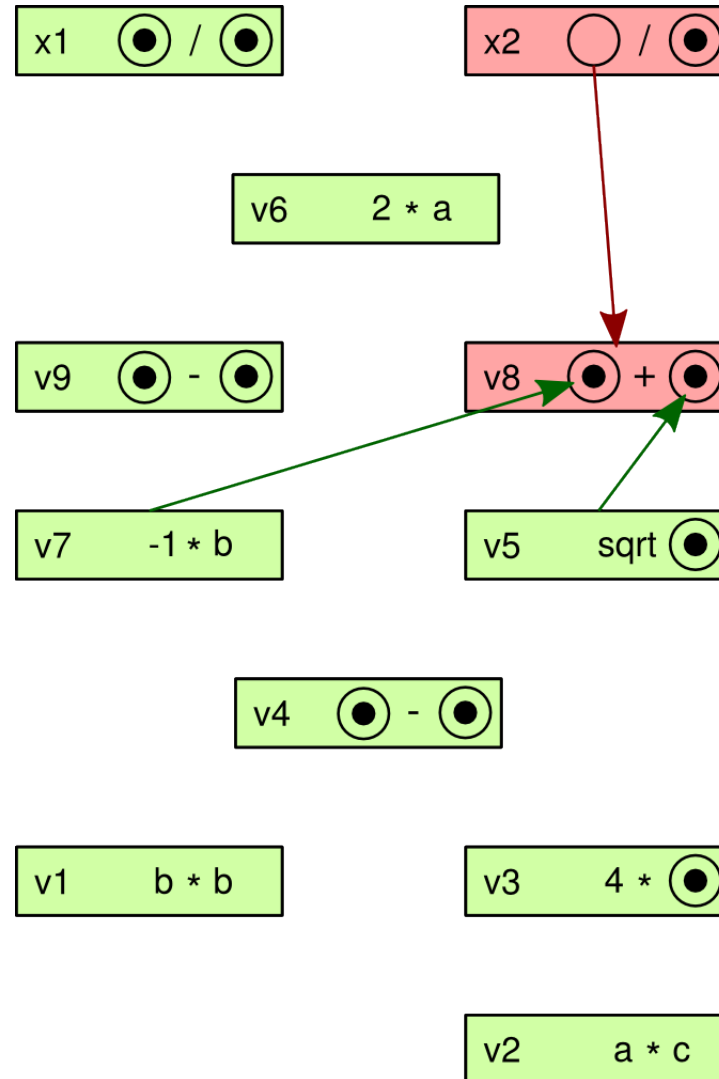


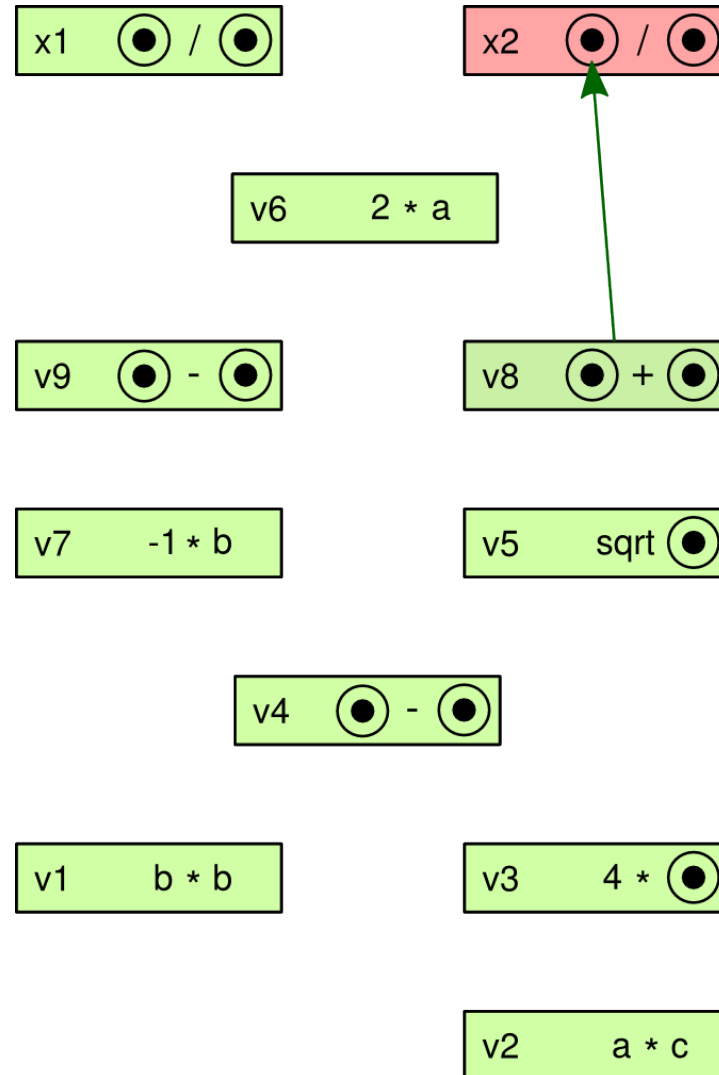












x1 $\odot / \odot$

x2 $\odot / \odot$

v6 $2 * a$

v9 $\odot - \odot$

v8 $\odot + \odot$

v7 $-1 * b$

v5 $\text{sqrt}(\odot)$

v4 $\odot - \odot$

v1 $b * b$

v3 $4 * \odot$

v2 $a * c$

- Értékelés
 - Párhuzamosságot automatikusan felderíti
 - Tisztán igényvezérelt számítógép nem terjedt el
 - De az elv igen:
 - Funkcionális programnyelvek
 - Google map-reduce API
 - MAP művelet: részfeladatokra bontás
 - REDUCE művelet: részeredmények összegzése
 - Működés:
 - Elosztott rendszer sok számítógéppel
 - Részfeladatokra bont + részfeladatokat delegálja + eredményeket összegzi és visszaadja → mindezt rekurzívan
 - Igényvezérelt modell!

- Összegzés:
 - CPU-k programozói interfésze vezérlésáramlásos
 - E fölött lehet igényvezérelt elven működő funkcionális nyelvet használni, vagy map-reduce-t
 - Ez alatt lehet adatáramlásos utasítás-végrehajtást megvalósítani



Főbb vezérlésáramlásos architektúrák

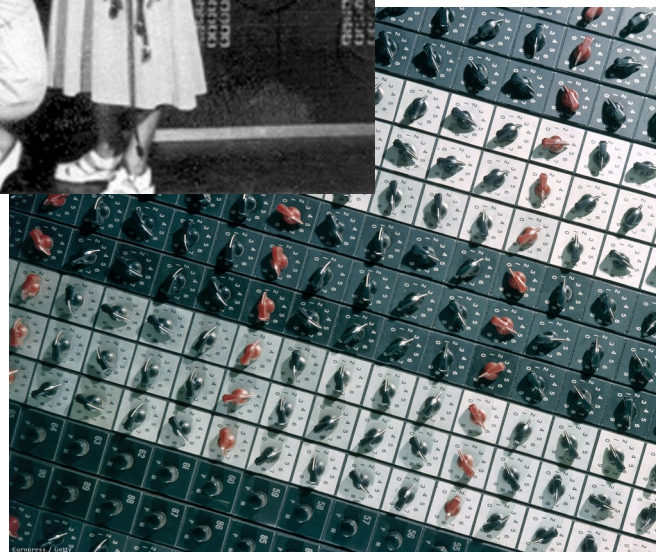
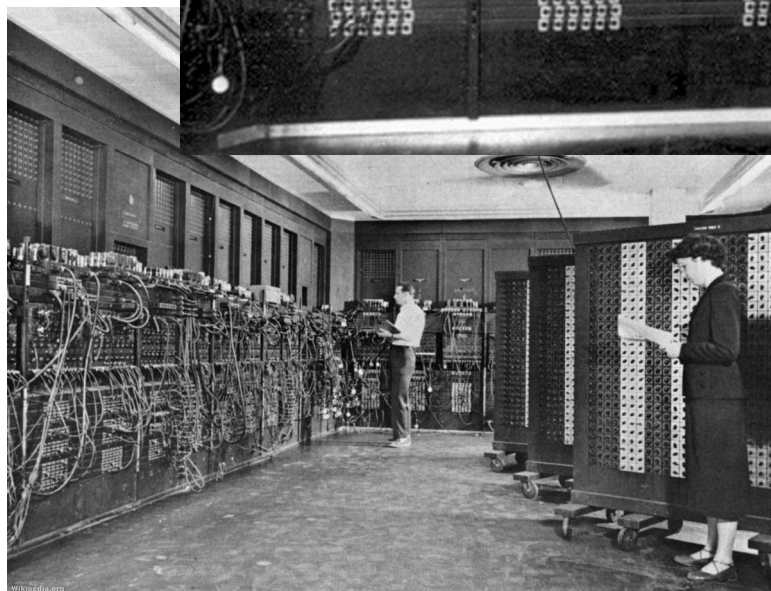
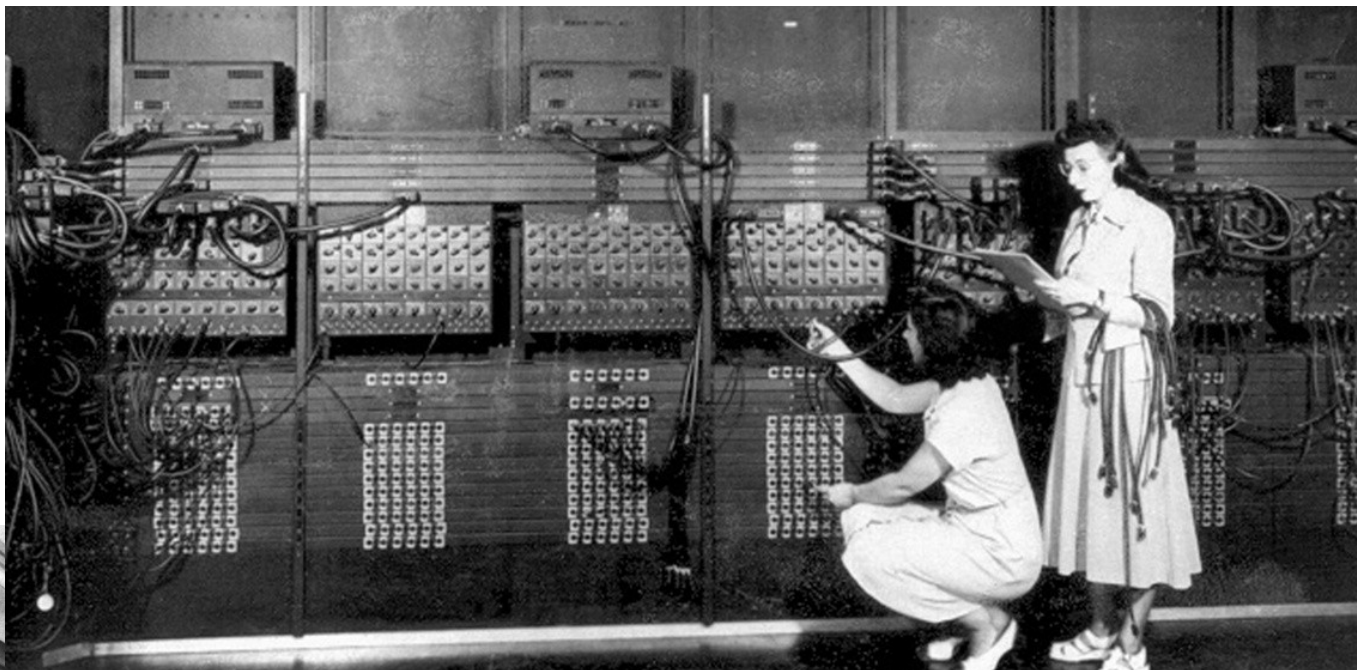
- 1945, "First Draft of a Report on the EDVAC" c. tanulmány
- Első Neumann elvű számítógép: Institute for Advanced Study, Princeton Egyetem (1945-1951)
- Igazából:
 - Turing – 1936
 - Eckert & Mauchly – 1943



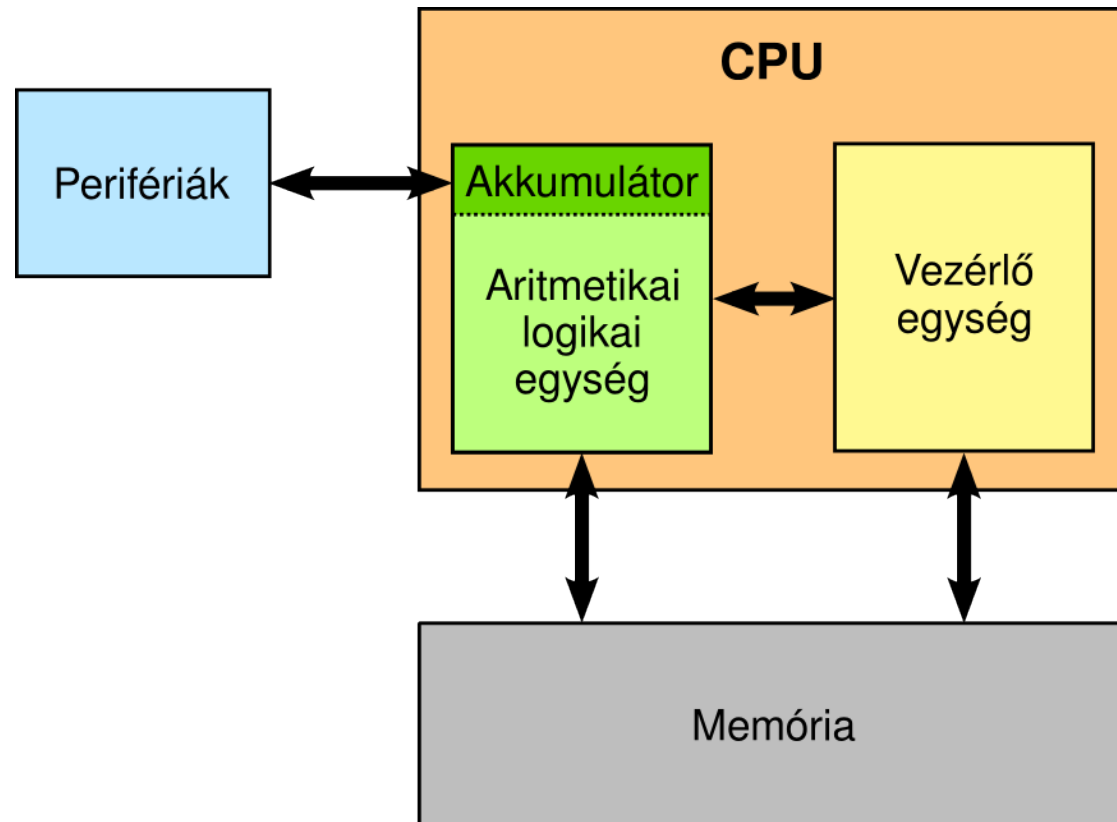
- Újdonság:

Az utasítások az adatokkal együtt vannak eltárolva a memóriában

- Korábban: programozás = kapcsolósor beállítás, újrarahuzalozás
- Motiváció:
 - A programot így könnyebb megváltoztatni
(Új program az ENIAC-ra = 3 hét munka!!)
 - Gyorsabban lehet több programot egymás után futtatni
 - Programot generáló programot lehet írni
 - Fordítóprogramok születése

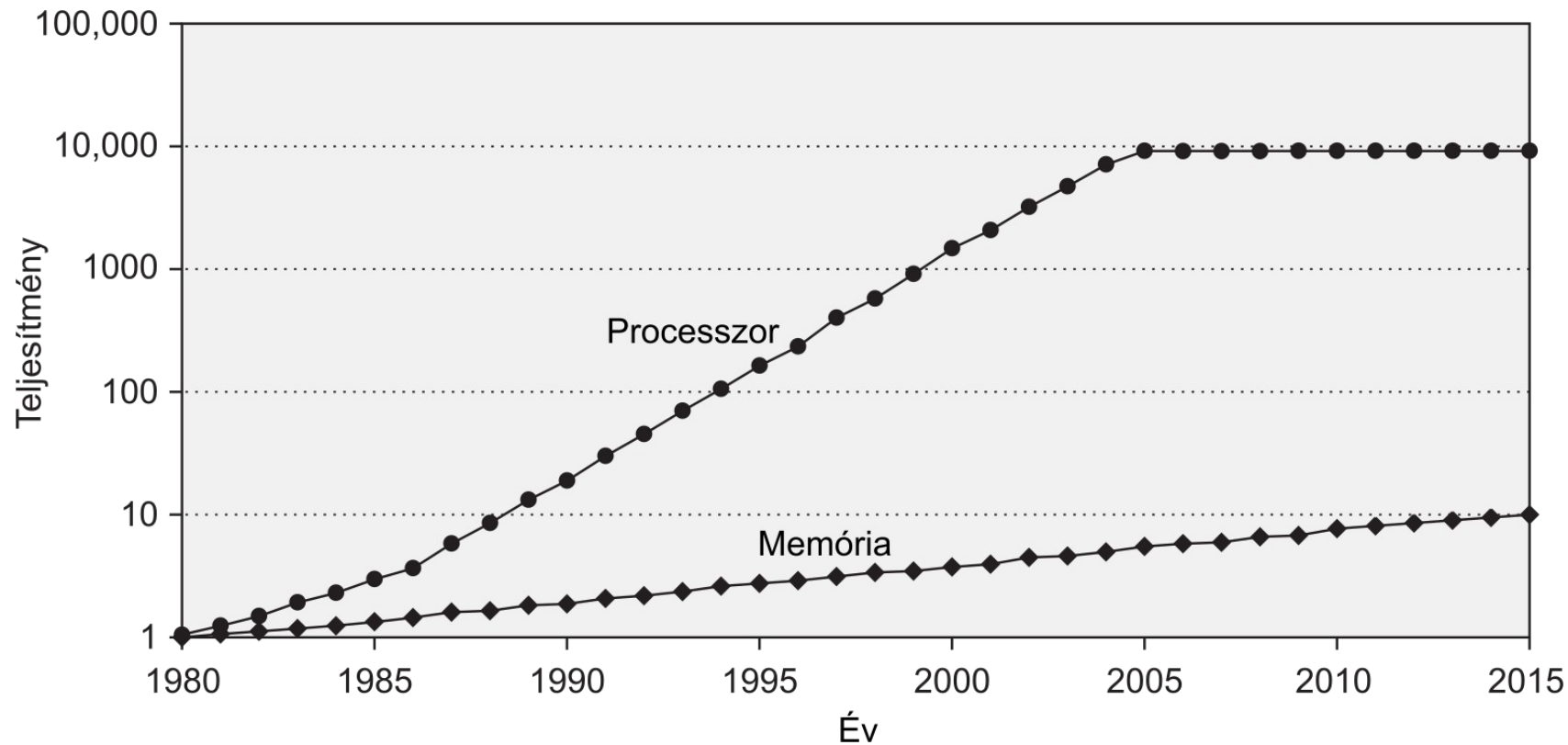


https://index.hu/tech/2016/02/14/70_eves_az_eniac/



- Memória tartalma:
 - Fix bitszélességű adataegységek
 - Lehet
 - Utasítás
 - Egész szám
 - Karakter
 - Lebegőpontos
 - ...
 - Megkülönböztetés: **NINCS!**
 - Attól függ, milyen utasítás nyúl hozzá

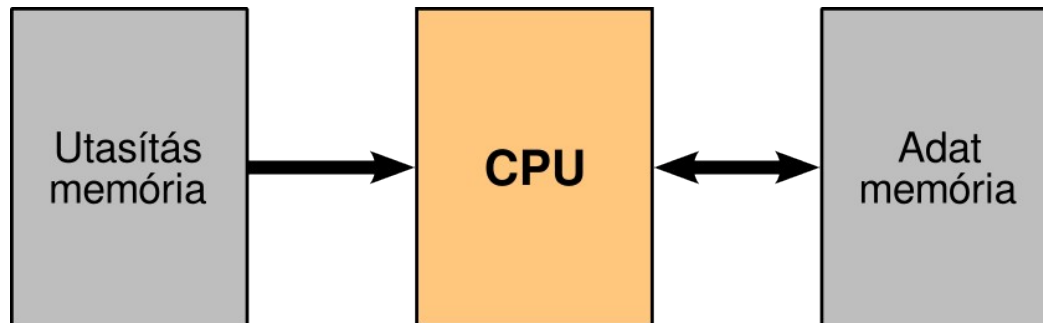
- Szűk keresztmetszet:
 - **Memória-sávszélesség!**



- Önmódosító programok:
 - Memóriaműveletekkel átírják saját utasításait
 - Pár bájttal spórolás vs. olvashatatlan kód
 - Soha!
- Példa: Fibonacci sorozat N. eleme (lásd: jegyzet)

- Harvard Mark I:
 - Lyukkártyáról olvassa az utasításokat
 - Relékapcsolókból álló memória tárolja az adatokat
- Eltérés a Neumann architektúrához képest:

Az utasítások és az adatok tárolása fizikailag elkülönül egymástól

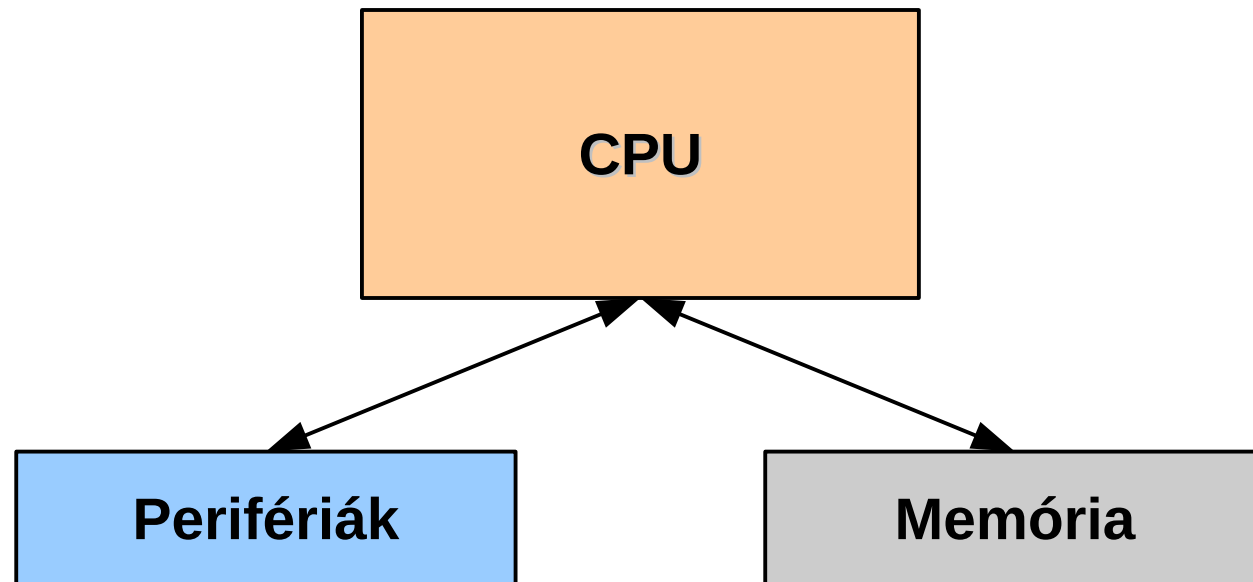


- Klasszikus Harvard architektúra:
 - Utasítások: kizárólag az utasításmemóriában
 - Adatok: kizárólag az adatmemóriában
- A kétféle memóriaművelet átlapolható:
 - Amíg lehívja az i . utasítás operandusát
 - Azzal egy időben lehívhatja az $i+1$. utasítást
- Módosított Harvard architektúra:
 - Utasítás memóriából is lehet adatot olvasni
- Alkalmazás:
 - Mikrokontrollerek, jelfeldolgozó processzorok (PIC, Atmel, ...)



Tematika

- Információfeldolgozási modellek (1. előadás)
 - **Vezérlésáramlásos**
 - Adatáramlásos
 - Igényvezérelt



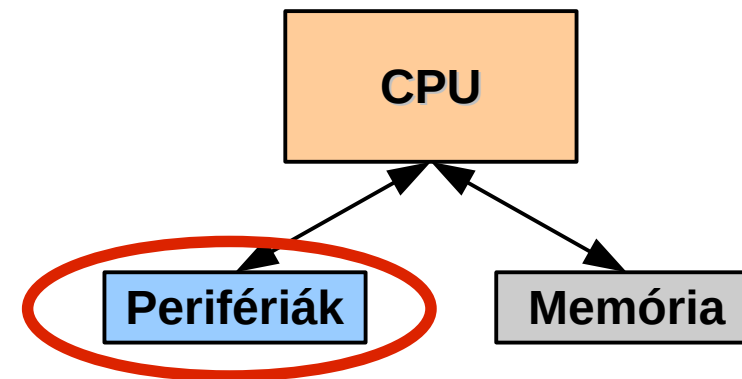
- **Periféria kezelés**

- Előadás

- Forgalom szabályozás, CPU tehermentesítése, buszok, déli híd – északi híd, stb...
 - A legfontosabb periféria: HDD/SSD
 - Csatolófelületek: PCI, PCI Express, USB
 - RAID

- Gyakorlat

- Konkrét feladat: számkódos ajtónyitó
 - Megvalósítás alapelemekből
 - Megvalósítás Arduino-val
 - Egyszerű periféria kezelés gyakorlása
 - Periféria okozta terhelés számítása
 - HDD késleltetés, átviteli sebesség, stb.
 - SSD blokk-kezelés, szemétygyűjtés, stb.



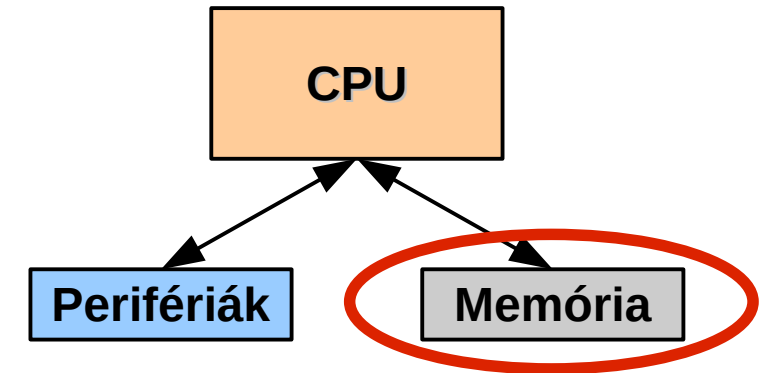
- **Memóriakezelés**

- Előadás

- Memóriatechnológiák:
SRAM, DRAM, SDRAM, DDR, DDR2, DDR3, DDR4, DDR5,
GDDR3, GDDR5, stb., memória időzítések
 - Virtuális tárkezelés
 - Cache memória
 - Lokalitástudatos programozás

- Gyakorlat

- Numerikus példák
 - Memóriaparancsok ütemezése
 - Laptábla menedzsment
 - Cache menedzsment



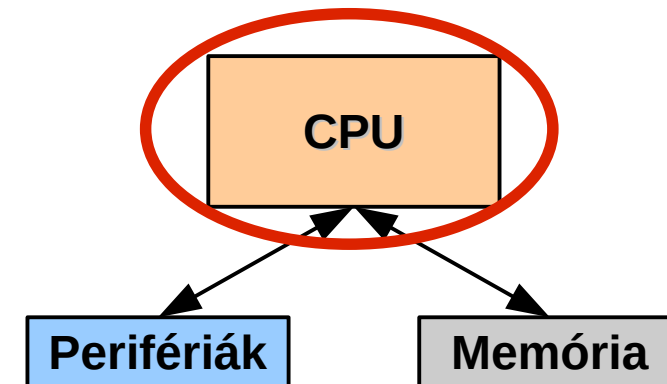
- **CPU**

- Előadás

- Pipeline utasításfeldolgozás
- Soron kívüli utasításvégrehajtás
- Szuperskalár processzorok
- Egyéb széles pipeline-ok
- Elágazásbecslés

- Gyakorlat

- Numerikus példák:
- Utasításütemezés, egymásrahatások azonosítása
- Utasítássorrend-optimalizálás



- **Párhuzamos feldolgozás**
 - SIMD feldolgozás
 - Vektorprocesszorok
 - Vektor utasításkészletek
 - Vektorizálás
 - Mutiprocesszoros rendszerek
 - Szorosan csatolt rendszerek, többmagos processzorok, többszálúságot támogató processzorok, ...
 - Osztott memóriakezelés
- **Támadás a spekulatív végrehajtás ellen**
 - Meltdown, Spectre



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

