

VEZÉRLŐK

© Benesóczky Zoltán 2004

A jegyzetet a szerzői jog védi. Azt a BME hallgatói használhatják, nyomtathatják tanulás céljából. Minden egyéb felhasználáshoz a szerző beleegyezése szükséges.

A feladatot két részre osztjuk: adatstruktúrára és vezérlőre

Adatstruktúra

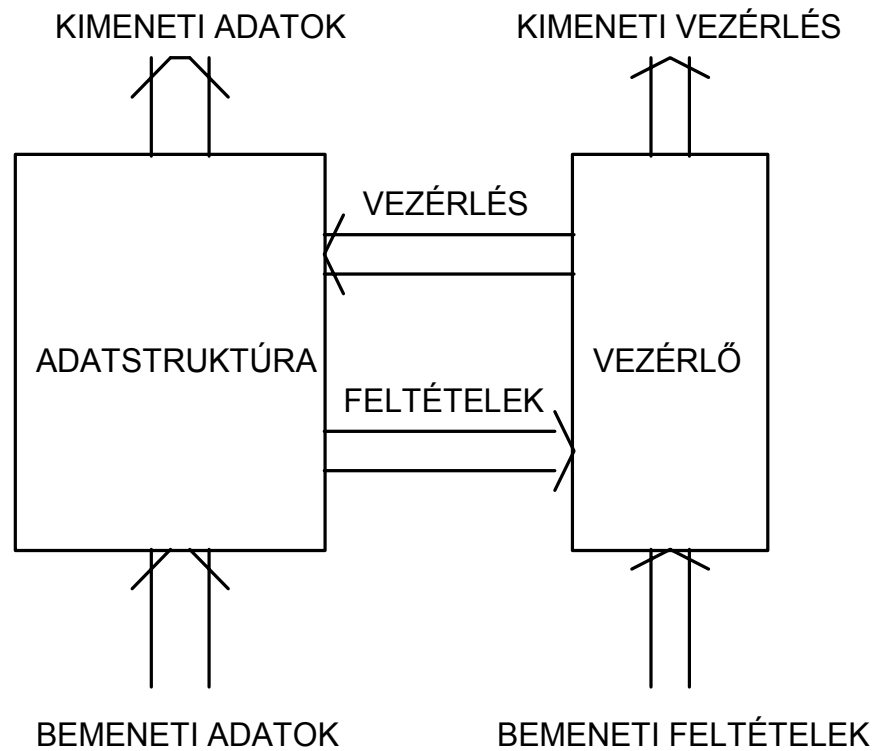
Ha egy feladatot *a bemeneti és kimeneti adatok között elvégzendő*, valamely algoritmussal megadott *transzformációnak* tekintünk, akkor *az adatstruktúra a transzformáció egyes lépéseinek megvalósításához szükséges funkcionális elemek, és az azokat összekapcsoló adatutak összessége.*

Az adatstruktúra tervezéséhez “intuíció” kell.

Vezérlő

A vezérlő feladata, hogy az adatstruktúrából és a külvilágból jövő feltételjelek figyelembe vételével a *vezérlőjelek segítségével megvalósítsa az adatstruktúrán az algoritmust.*

A vezérlő szisztematikusan tervezhető.



- a feladat megoldás algoritmusának megtervezése
- adatstruktúra megtervezése,
- az adatstruktúrát vezérlő algoritmus megtervezése (folyamatábra vagy állapotgráf)
- szisztematikus vezérlő kiválasztása és megtervezése (és mikroprogramozott vezérlő esetén a mikroprogram megírása), vagy programozható logika esetén a vezérlő állapotgráfjának megadása HW leíró nyelven.

A vezérlő:

Hagyományos szinkron sorrendi hálózat.

A vezérlő állapotgráfjával vagy folyamatábrával megadott állapotgép megvalósítása.

Régen, SSI elemekkel a nagy állapotszámú SSH-k megtervezése és megvalósítása problémás volt, a bonyolult vezérlőfüggények miatt.

Ezért az MSI elemek megjelenése után elterjedtek el a ***szisztematikus vezérlők.***

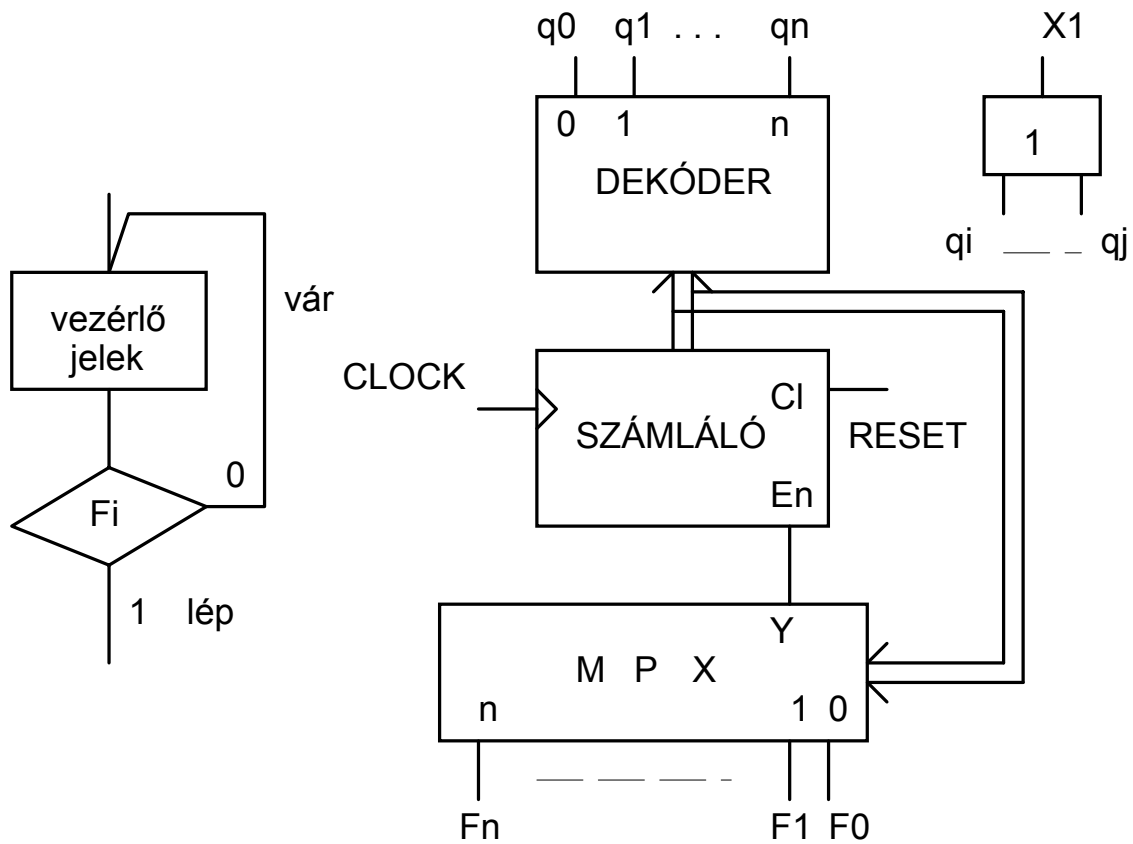
A fázisregiszteres (pl. számláló típusú) vezérlőkkel már nagyobb állapotszám esetén is viszonylag egyszerű volt a tervezés, de ***minden módosítás új HW tervezését vonta maga után.***

Erre megoldást a ***mikroprogramozható vezérlők*** hoztak, melyekkel még nagy állapotszám esetén is könnyen tervezhető vezérlő és viszonylag ***nagyobb változtatások is csak a ROM tartalom módosítását*** igénylik.

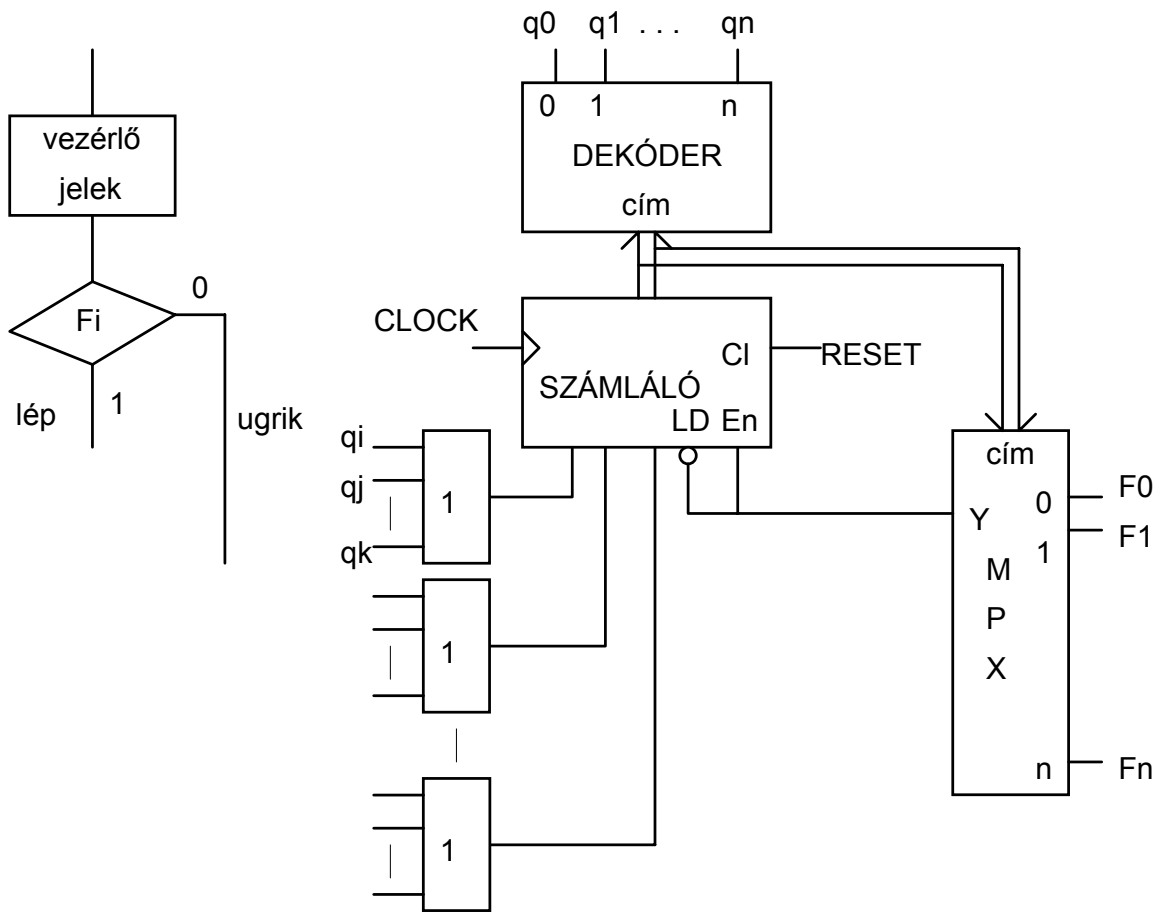
Ma az állapotgráf megadható HW leíró nyelven, és a tervező rendszeren szinte tetszőleges állapotszám esetén megtervezhető és megfelelő ***programozható logikában*** megvalósítható.

Számláló típusú vezérlők

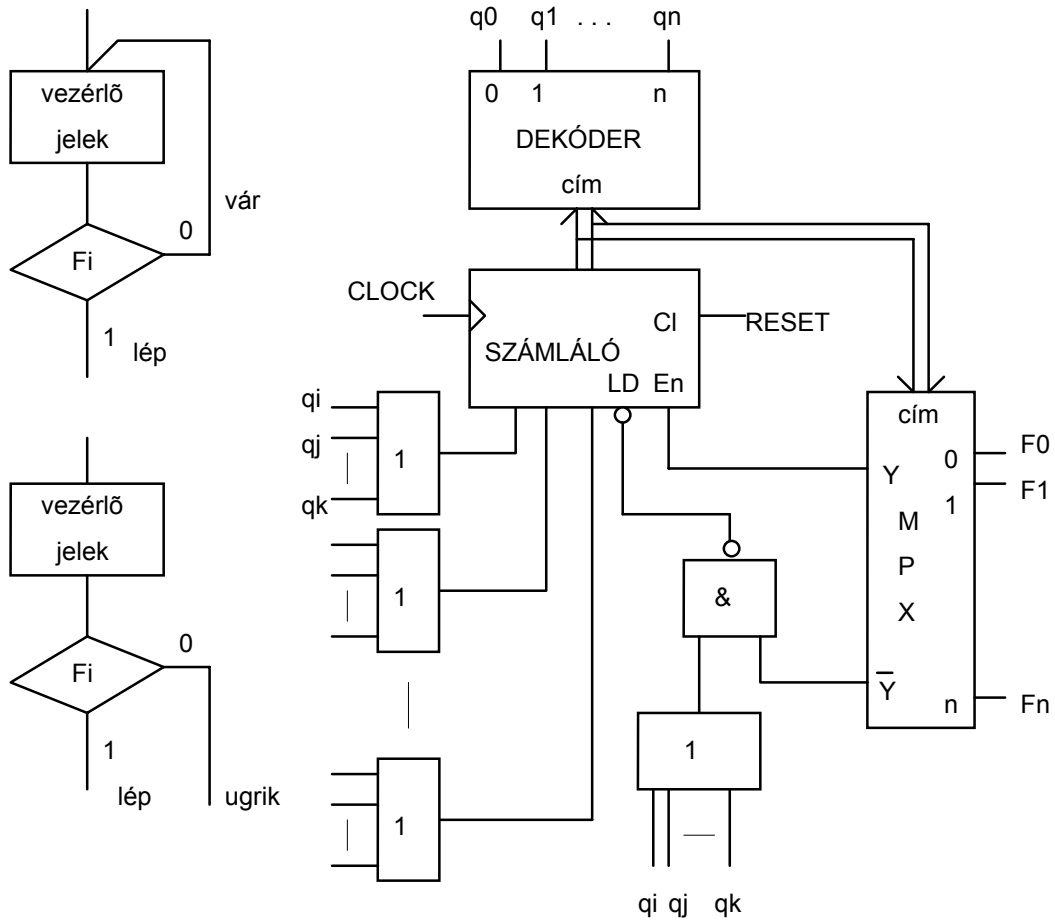
Lép vagy vár típusú vezérlő



Ugrik vagy lép típusú vezérlő



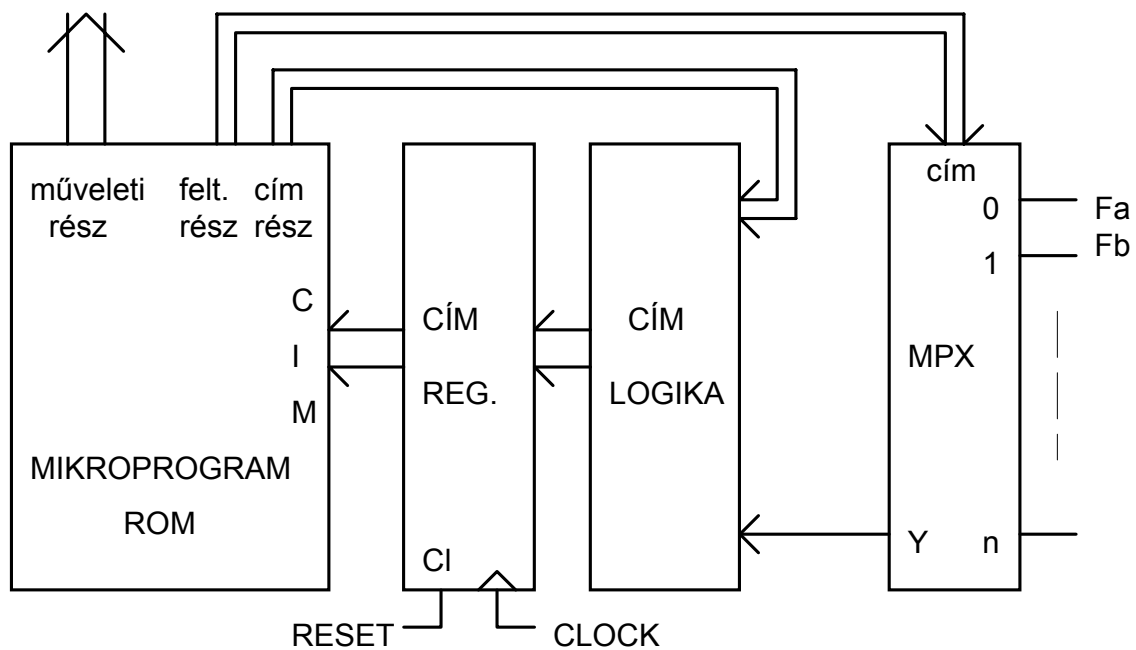
Ugrik vagy lép vagy vár típusú vezérlő



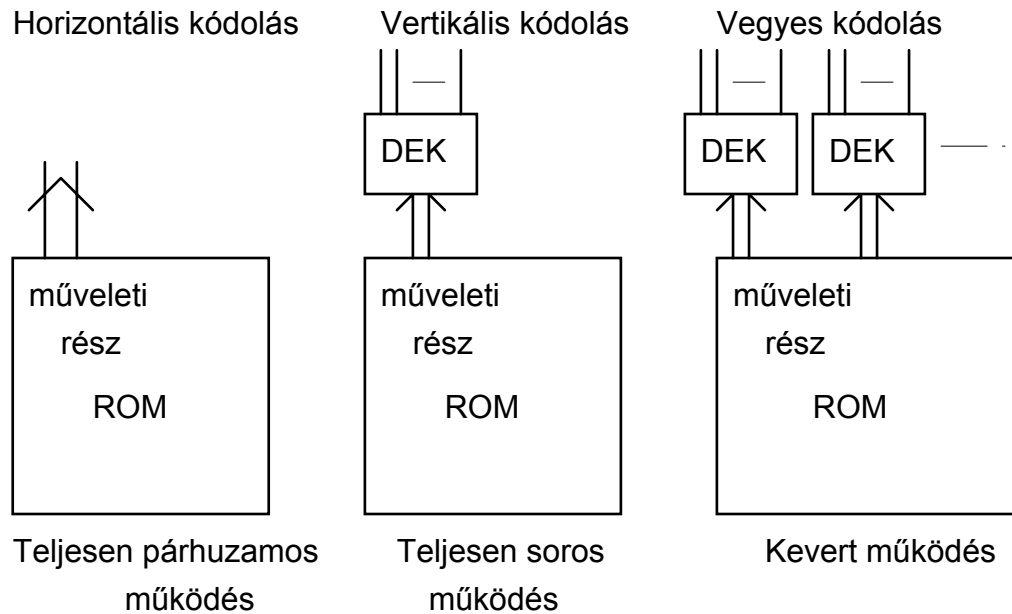
Mikroprogramozott vezérlők általános felépítése

Mikroutasítás felépítése:

<műveleti mező><feltétel mező><cím mező>



A műveleti rész kódolása

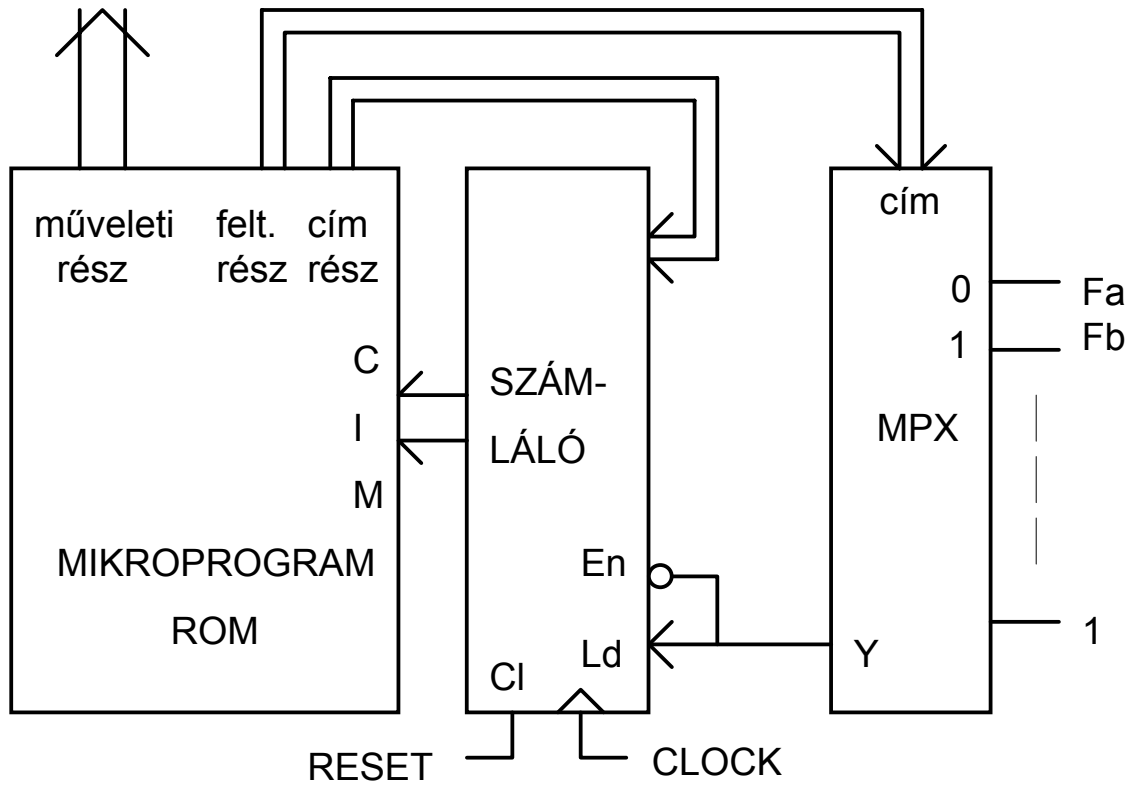


Legpazarlóbban bánik a ROM szószélességével a horizontális kódolás, viszont az összes jel egyszerre aktív lehet. (Maximálisan párhuzamos működést tesz lehetővé)

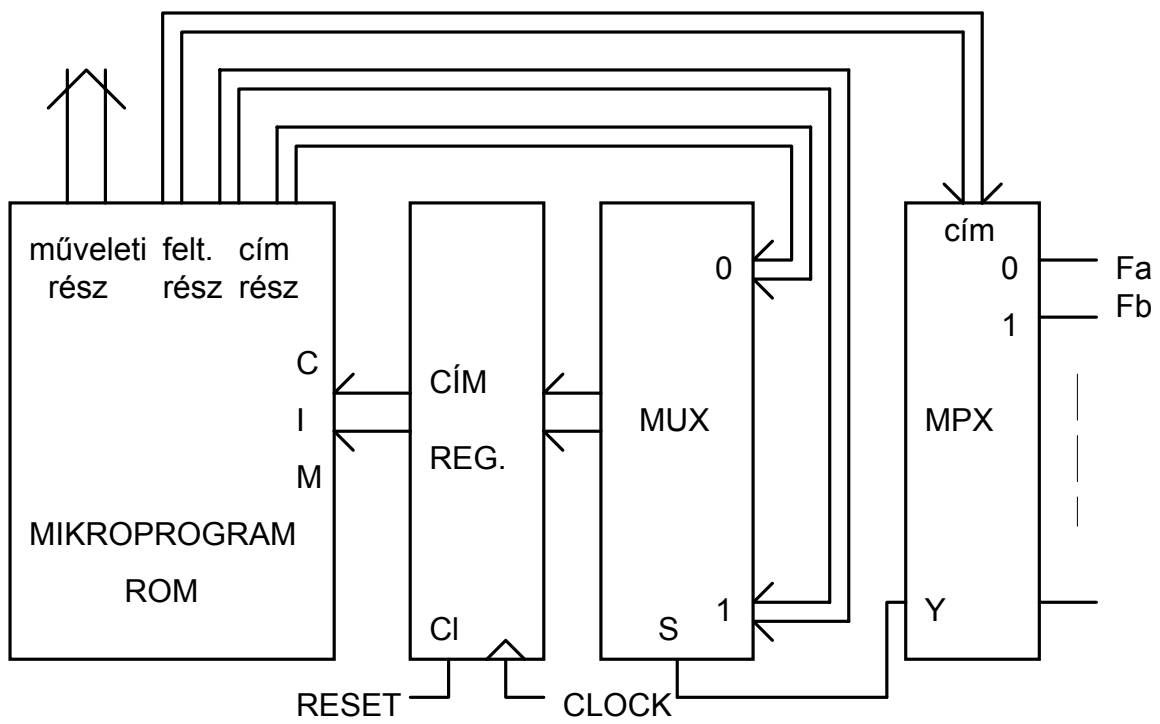
A vertikális kódolás esetén egyszerre csak egy kimeneti jel lehet aktív, így ezzel csak egymás után lehet kiadni az egyes vezérlő jeleket (maximálisan soros működés).

A vegyes kódolással lehet legoptimálisabban kihasználni a ROM szószélességét, a párhuzamos működés lehetősége mellett (egyszerre több jel is kiadható).

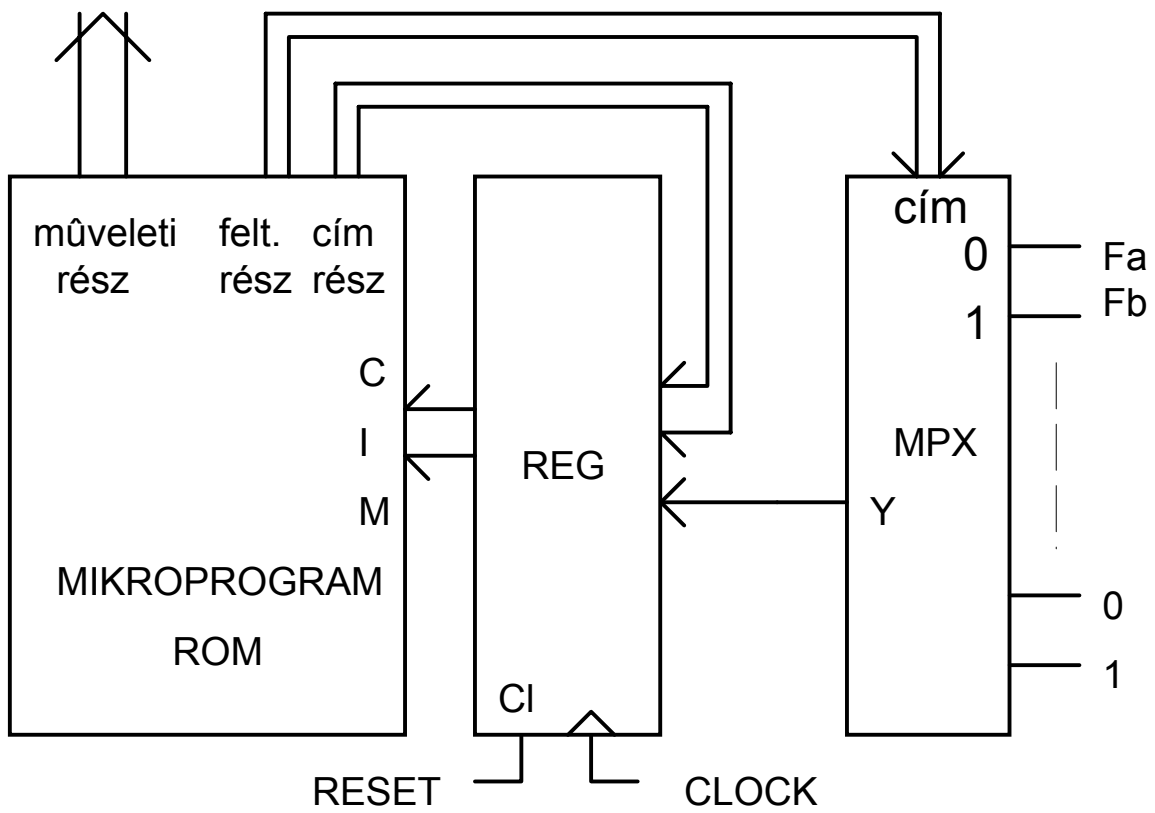
Számlálós címképzésű mikroprogramozott vezérlő



Két címből választó mp. vezérlő



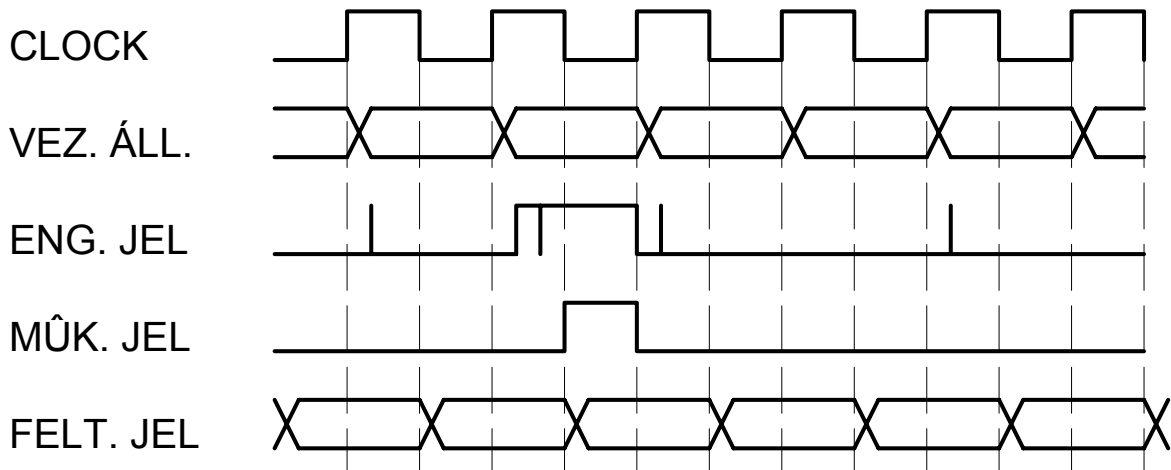
Feltételt címbe másoló mp. vezérlő



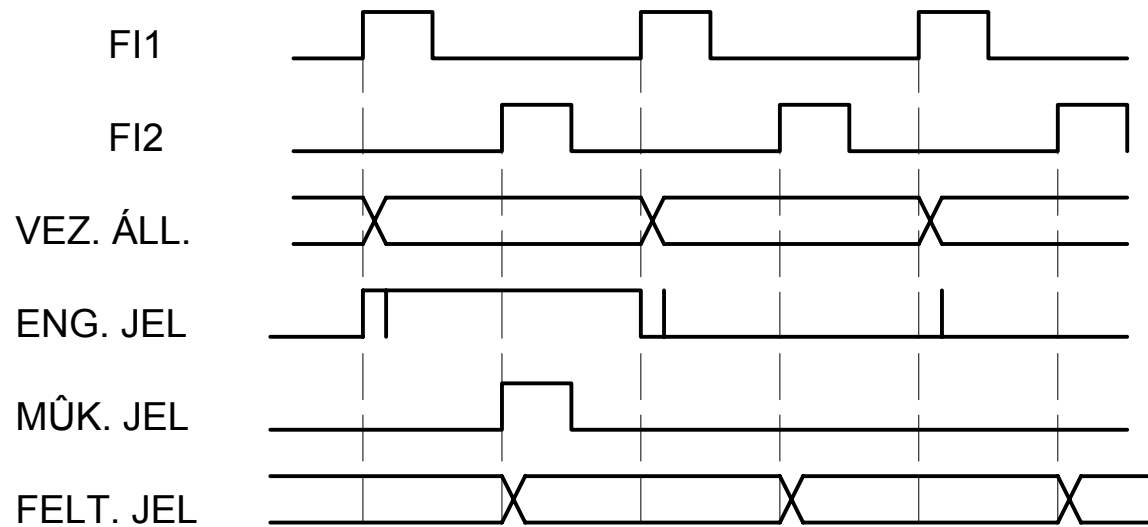
Órajelezési technikák

Hazárdmentes jelek kialakítása, az órajrel kapuzásával.

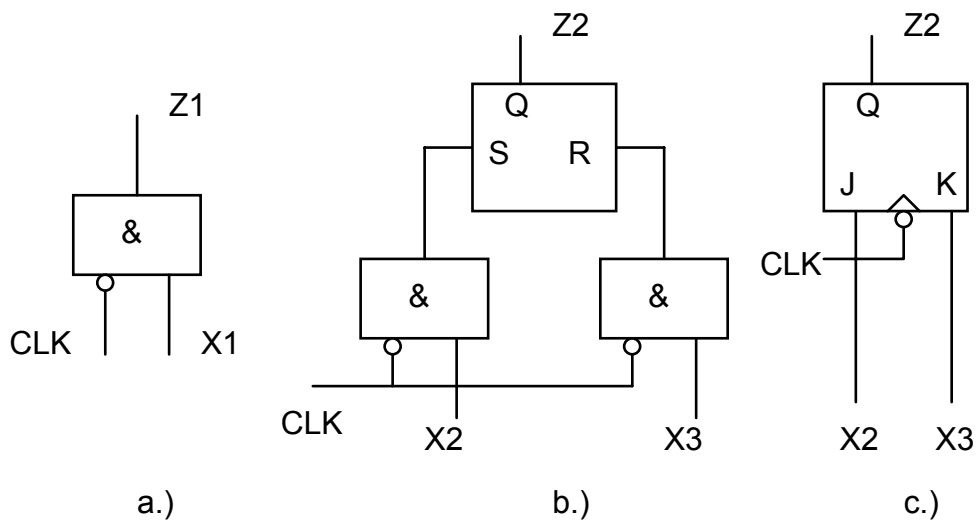
Egyfázisú órajelezés



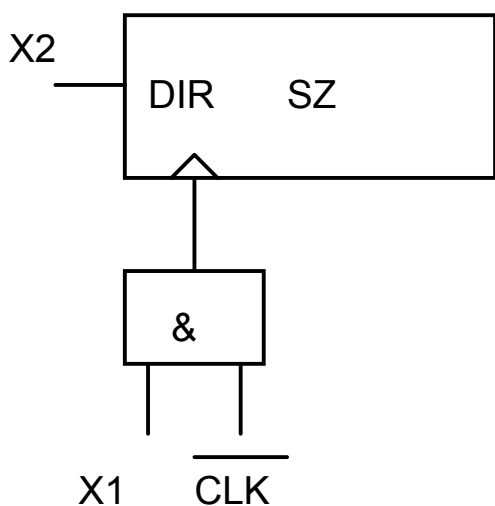
Kétfázisú órajelezés



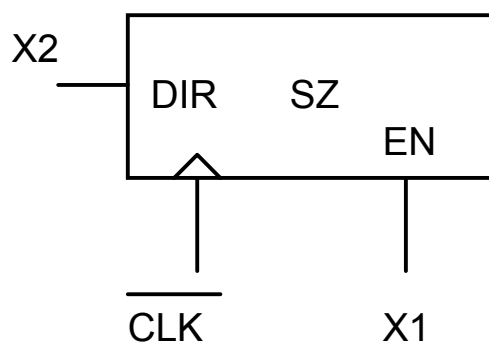
A modern tervezés során kerüljük az órajel kapuzást, mert az így előállított jelek órajelhez képesti fázisa a különböző késleltetések miatt eltérő, ami a további felhasználásuknál problémák forrása lehet.



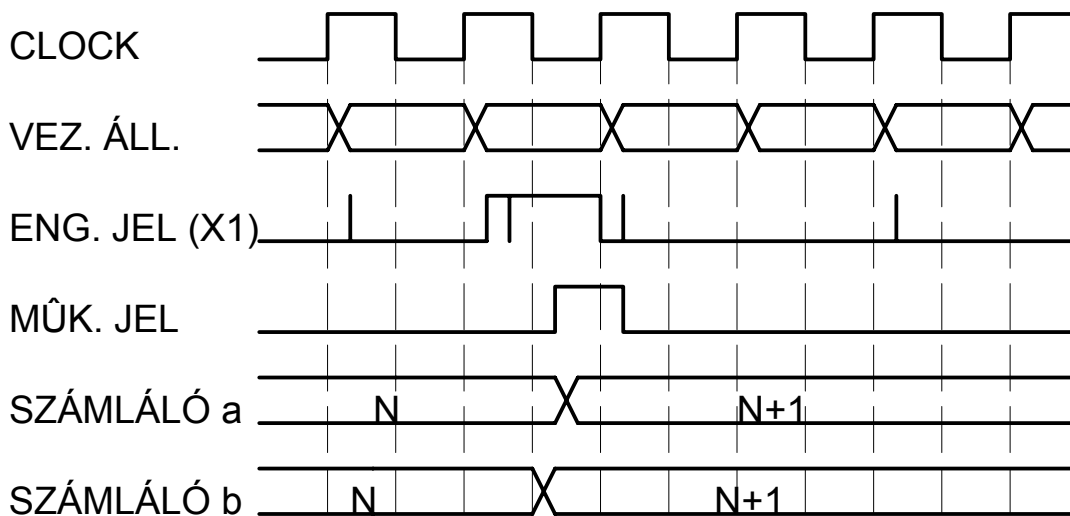
Ezért tiszta szinkron tervezést célszerű alkalmazni, vagyis minden áramkör órajele ugyanaz és a működtetéseket engedélyező jelekkel oldjuk meg (pl. fenti c ábra).



a.)

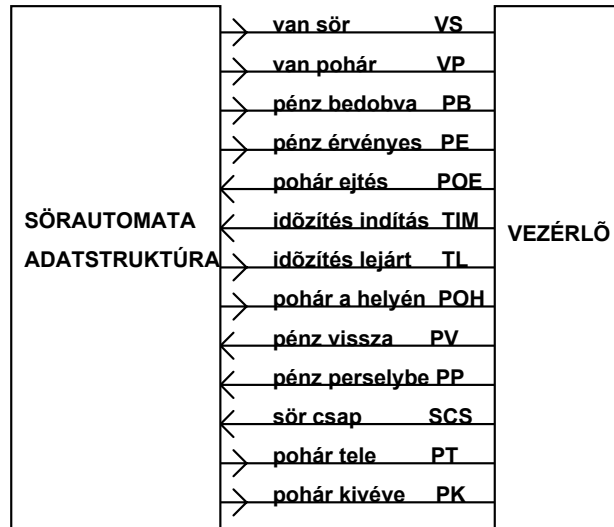


b.)

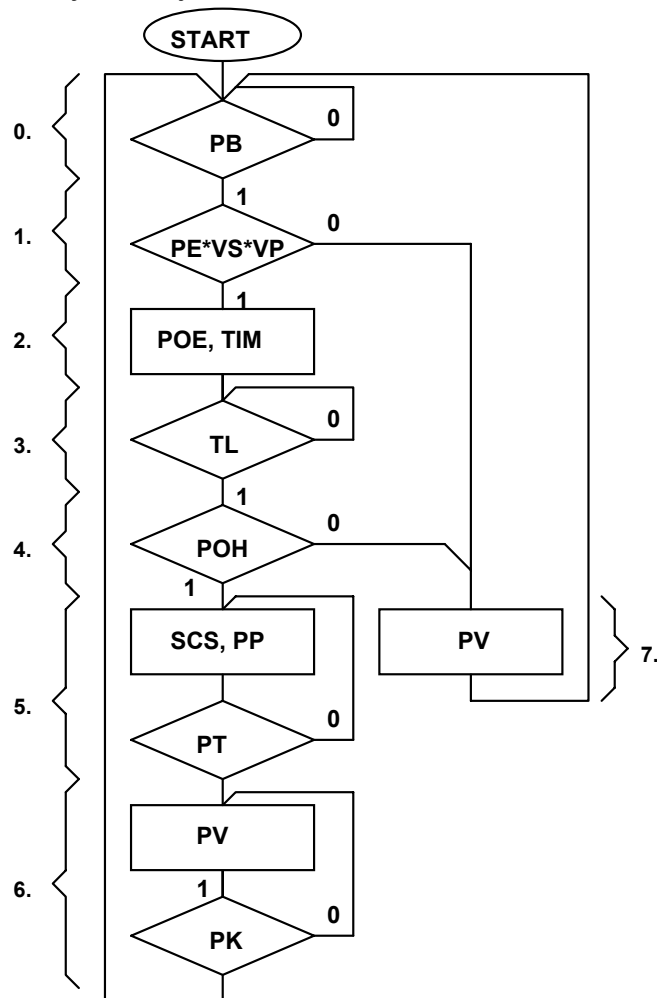


Az idődiagram mindkét esetben hasonló, de a “b” esetben a számláló hamarabb lép az órajelhez képest.

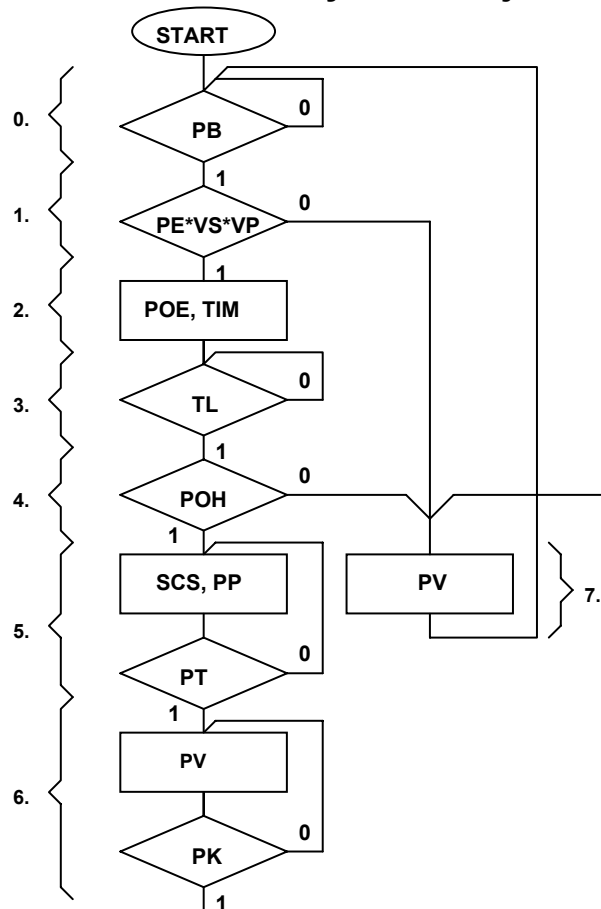
Sör automata



Vezérlő folyamatábrája

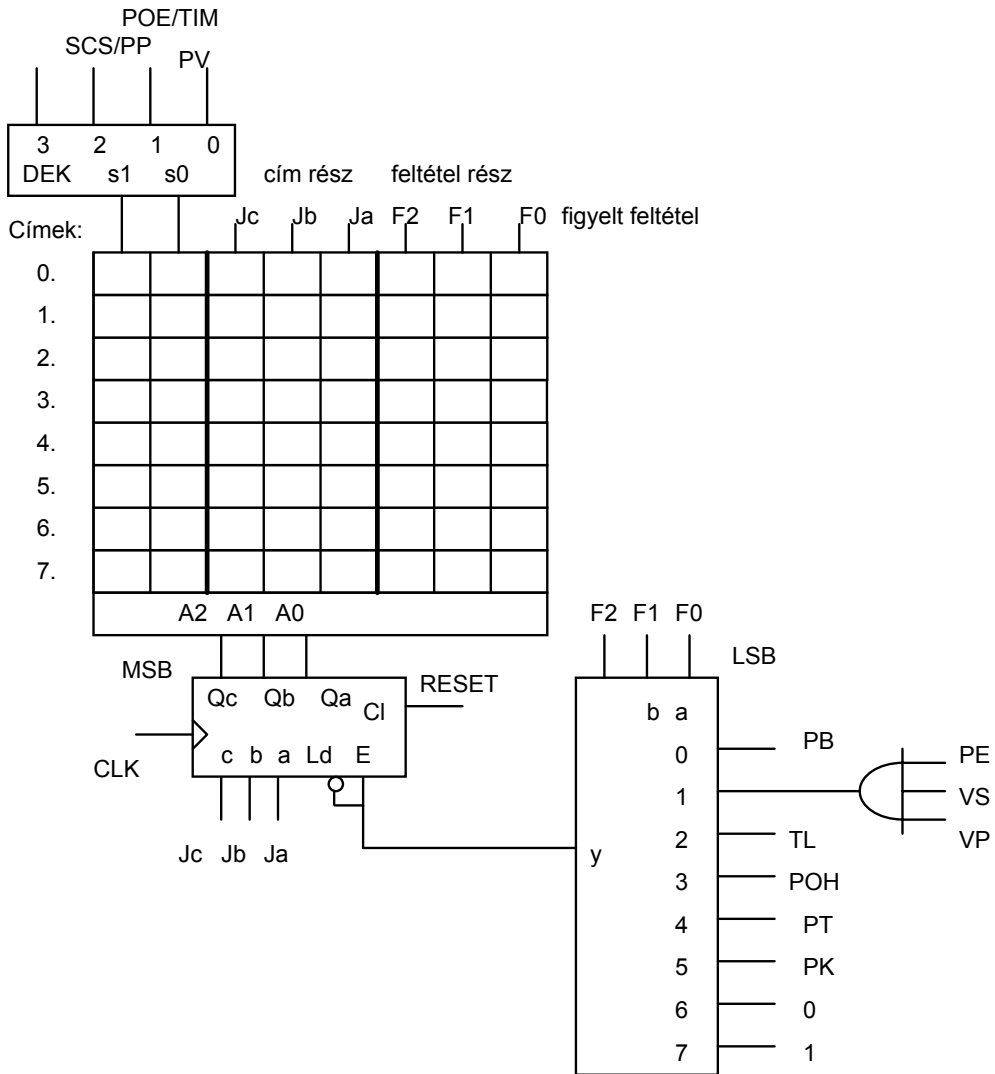


Számlálós vezérlő folyamatábrája



A számlálós vezérlőnél, ha feltételes elágazás van, az n -edik állapotra következő állapotok egyikének $n+1$ -ediknek kell lenni. Néha ennek teljesüléséhez üres állapot beiktatása szükséges. A példában a 6. állapot után a 6. vagy a 0. következett volna. Ezért a 6. után be kellene iktatni egy 7-es kódú üres állapotot. Ezzel 9 állapotunk lenne (0.-8.). Azonban, ha az eredeti 8. állapotot 7.-nek nevezzük át, ebbe léphetünk a 6.-ból. Itt ugyan PV-t vezéreljük, de ez nem okoz problémát.

Megvalósítás számlálós címképzésű vezérlővel, vertikális kódolással



Megvalósítás két címből választó vezérlővel, horizontális kódolással

