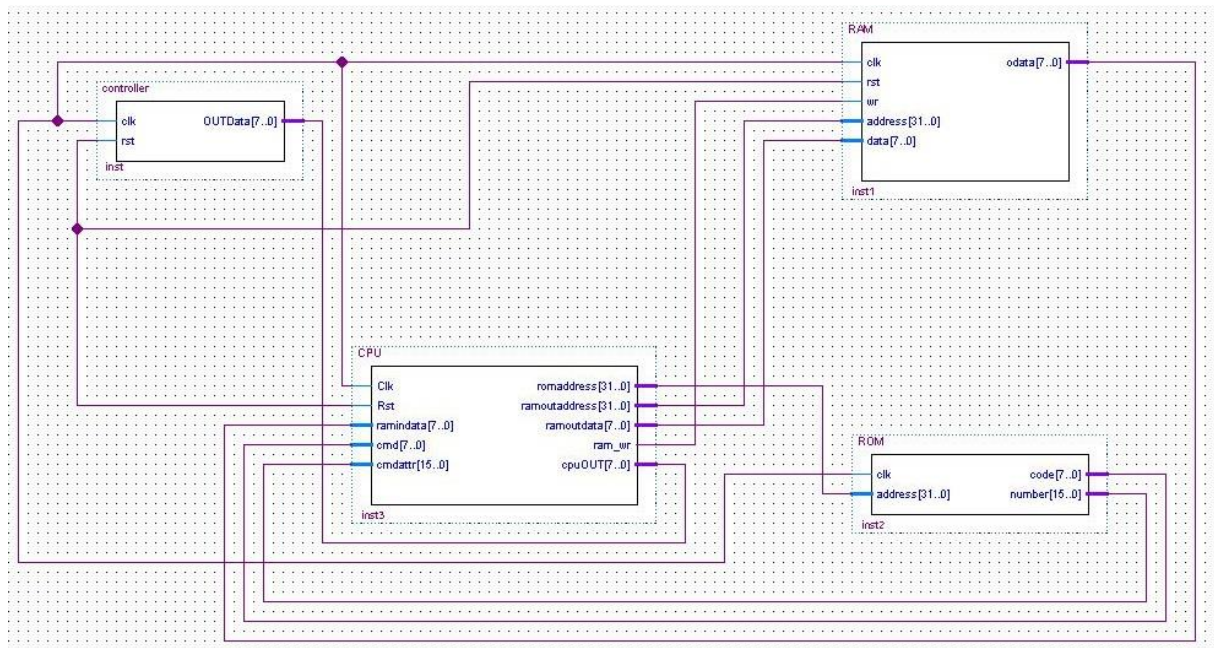


Készítse el egy mikrogép architektúráját. A mikrogép 8 bites adatszélességgel rendelkezik, a programmemóriája 1K, illetve az adatmemória mérete 512 Byte.



Controller: Vezérli a mikrogépet. Tartja a kapcsolatot a külvilággal ahonnan a „clk” bemenetére várja az óra jelet az „rst” bemenetére érkező jellel újra indítható mikrogép. A 8 bites „outdata” kimeneti porton a mikrogép programja által a kimenetre küldött bitek jelennek meg.

CPU: a mikrogép parancs feldolgozója.

ROM: tartalmazza a mikrogép programmemóriáját. /Egy program sor 3 byte-ból áll, az egy byte-os utasítás kódból és a két byte-os attribútumból./

RAM: a mikrogép adatmemóriája.

A mikrogép utasítás készlete:

- **ADD** Az utasítás attribútumában megadott RAM címen lévő érték hozzáadása az akkumulátor regiszterhez és az eredmény függvényében beállításra kerül a negatív ill. 0 jelzők.
- **SUB** Kivonja az utasítás attribútumában megadott RAM címen lévő értéket az akkumulátor regiszter aktuális értékéből és az eredmény függvényében beállításra kerül a negatív ill. 0 jelzők.
- **STA** Az akkumulátor regiszter aktuális értékét eltárolja az utasítás attribútumában megadott RAM címen.
- **LOA** Betölti az utasítás attribútumában megadott RAM címen lévő értéket az akkumulátor regiszterbe.
- **DAT** Beállítja az akkumulátor regiszter értékét az utasítás attribútumában megadott értékre. Mivel a mikrogép 8 bites akkumulátor regiszterrel rendelkezik ezért csak 256-nál kisebb érték lesz figyelembe vagyis az attribútum 2. byte-a.
- **BRA** Beállítja a program mutatót az az utasítás attribútumában megadott értékre és onnan folytatódik a program futása.

- **BRZ** Ha a 0 jelző értéke 1 akkor átállítja a program mutatót az az utasítás attribútumában megadott értékre és onnan folytatódik a program futása.
- **BRP** Ha a negatív jelző 1 akkor átállítja a program mutatót az az utasítás attribútumában megadott értékre és onnan folytatódik a program futása.
- **OUT** Kiírja az akkumulátor aktuális értékét a kimenetre.
- **BRK** Program vége.

A mikrogép programozásához készült egy egyszerű fordító így lehetőség van szöveg fileban megírt a programot lefordítani. A szöveg file egy-egy sorába adjuk meg a fenti utasítások valamelyikét majd utána az attribútumot. Számok esetén 10 számrendszer szerint adjuk meg az értékeket. Használhatunk címkéket is melyek az ugró utasítások használatát nagyban megkönnyítik. A címkéket '#' vezessük be.

```
pl.:    #FOR    LOA 1
        BRP #FOR
```

Az OUT és a BRK utasításoknak elvileg nincs szükségük attribútumra, de mivel a fordító szintaxis elemzője elvárja, hogy minden utasításnak kell lennie attribútumának ezért egy számot ezen utasítások attribútumában is meg kell adni. pl.: OUT 0

A fordító által előállított TestCode.hex file tartalmát kell a Helper.vhd `constant PROGRAM:ROMType:=` sorában az = jel után bemásolni.

A fordító ezen a címen érhető <https://github.com/koletom/VHDL/tree/master/microgep/Compiler> el. A TestCode.asm file-ba kell elhelyezni a mikrogép programjának olvasható kódját majd LMCCompiler.exe lefuttatásával elkészül a TestCode.hex. Ha szintaktikai v. más hiba van a kódban azt az a program standard kimenetre írja ki. /A fordító forráskódja (VS2017 solution) itt <https://github.com/koletom/LMCCompiler> található./