

Tema 1 PSSC

Conceptul de *pipeline* este folosit în designul procesoarelor și al altor dispozitive electronice pentru a crește numărul de instrucțiuni ce pot fi executate într-o unitate de timp. În tehnica *pipeline* se execută simultan (dar decalat) mai multe instrucțiuni, acest lucru realizându-se fără componente hardware suplimentare.

Pipeline este o tehnică de creștere a vitezei de execuție totale a procesoarelor, fără a ridica tactul. Ea constă în subdivizarea fiecărei instrucțiuni într-un număr de etape sau segmente, fiecare etapă fiind executată de câte o unitate funcțională separată a procesorului (segment *pipeline*). Segmentele *pipeline* sunt conectate între ele într-un mod analog asamblării unei conducte din segmente de țevă. Segmentele tipice de execuție ale unei instrucțiuni mașină pe un procesor sunt:

- FI - aducere instrucțiune (fetch instruction). Se referă la un ciclu special în care procesorul citește din **memorie** instrucțiunea următoare ce trebuie executată.
- DI - decodare instrucțiune (decode instruction). În această etapă instrucțiunea este recunoscută și procesorul colectează toate informațiile necesare pentru executarea ei.
- CO - calculare adresă operand (calculate operand address). Se calculează adresa de memorie de unde se aduce primul operand.
- FO - aducere operand (fetch operand). Citește operandul aflat în memorie la adresa calculată în pasul anterior.
- Dacă e nevoie, cele 2 etape anterioare se repetă și pentru al doilea operand etc.
- EI - execută instrucțiune (execute instruction). Execuția propriu-zisă a instrucțiunii inclusiv calculul rezultatului ei.
- WO - scriere operand (write operand). Scrierea rezultatului înapoi în memorie.

Avantaje:

Rata de execuție a instrucțiunilor crește dacă se folosește o arhitectură de tip pipeline. Unele circuite combinatoriale cum ar fi sumatoarele sau multiplicatoarele pot fi făcute să funcționeze mai rapid prin mărirea complexității circuitelor electronice respective, dar dacă se folosește pipelining se poate obține același spor de performanță fără a crește complexitatea.

