

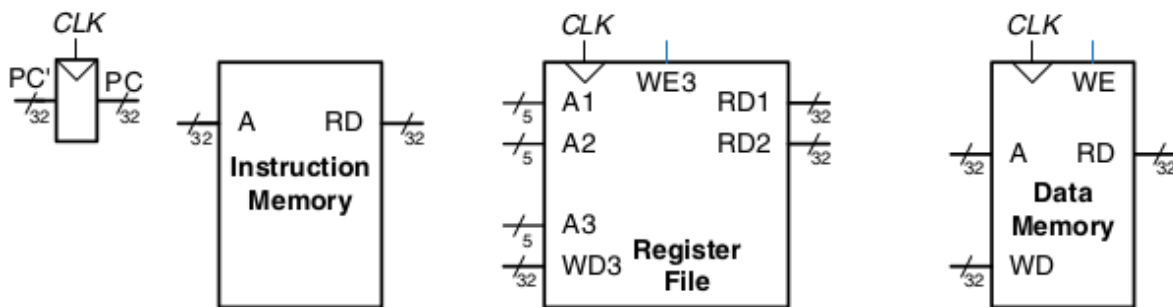
Implementarea traseului de date a procesorului MIPS de un ciclu:

O arhitectură a calculatorului este definită de setul de instrucțiuni și de *starea sa arhitecturală*. Starea arhitecturală a procesorului MIPS constă din contorul de programe și cele 32 de registre. Orice microarhitectură MIPS trebuie să conțină toată această stare. Pe baza stării arhitecturale actuale, procesorul execută o instrucțiune specială cu un anumit set de date pentru a produce o stare arhitecturală nouă. Unele microarhitecturi conțin o stare nonarhitecturală suplimentară pentru a simplifica logica sau a îmbunătăți performanța;

Pentru a menține microarhitecturile să fie ușor de înțeles, vom realiza doar un subset al setului de instrucțiuni MIPS. Vor fi realizate doar următoarele instrucțiuni:

- Instrucțiuni aritmetice / logice de tip R: add, sub, și, sau, slt
- Instrucțiuni cu acces la memorie: lw, sw
- Ramificări: beq

Elementele de bază care determină starea curentă a execuției în microarhitectura MIPS:



Contorul de programe (PC) este un registru obișnuit de 32 de biți. Ieșirea, PC, indică instrucțiunea curentă. Intrarea lui, PC', indică adresa următoarei instrucțiuni.

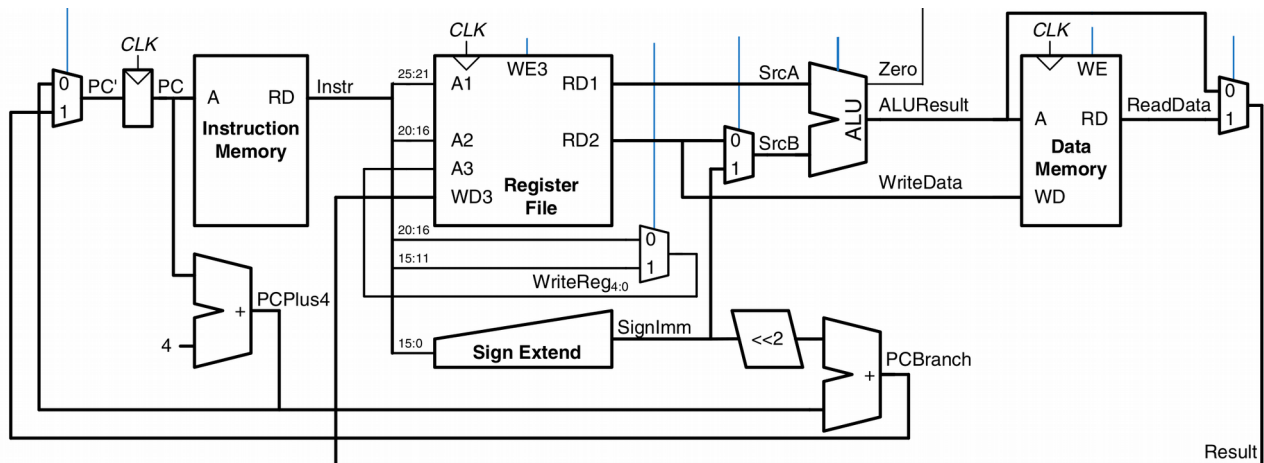
Memoria de instrucțiuni are un singur port de citire. Are o intrare de adresă de 32 de biți, A, și se citește datele de 32 biți (instrucțiuni) de la acea adresă spre ieșirea de date, RD.

Setul de regiștri de 32 de biți câte 32 de cuvinte are două porturi de citire și un port de scriere. Porturile de citire primesc combinații de adrese de 5 biți, A1 și A2, fiecare specificând unul dintre cele $2^5 = 32$ registre ca operatori sursă. Ei citesc valorile regiștrilor de 32 biți spre ieșirile de date citite RD1 și, respectiv, RD2. Portul de scriere are o intrare de adresă pe 5 biți, A3; o intrare de date de scriere pe 32 de biți, WD; o intrare de permisul de scriere, WE3; și un semnal de ceas. Dacă funcția de scriere este în 1 logic, setul de regiștri scrie datele în registrul specificat pe frontul pozitiv a semnalului de clock.

Memoria de date are un singur port de citire/scriere. Dacă scrierea se permite, semnalul WE este în starea de 1 logic, datele WD se scrie în adresa A pe front pozitiv semnalului de clock. Dacă WE este în stare de 0 logic, datele se citesc de la adresa A pe magistrala RD.

Memoria de instrucțiuni, setul de regiștri și memoria de date realizează citirea în mod combinațional. Cu alte cuvinte, dacă adresa se modifică, noile date apar la RD după o anumită întârziere de propagare; semnalul de clock nu este implicat. Însa datele se scrie doar pe frontul pozitiv a semnalului de clock. În acest caz, starea sistemului se schimbă numai la anumit front de clock. Adrese, date și procesul de scriere trebuie să fie preconfigurate înainte de frontul pozitiv de clock și trebuie să rămână stabile un anumit timp de reținere după frontul clockului.

Procesorul poate fi privit ca o mașină gigantică de stări finite sau ca o colecție de mașini de stări finite mai mici care interacționează între ele.



Schema generală a traseului de date a microprocesorului MIPS

Mersul lucrării:

1. În această lucrare trebuie să fie implementate și testate modulele următoare:

1. **Instruction** memory (pe baza la **RAM**)
2. **Register file** (pe baza la **RAM**)
3. **ALU**
4. **Data** memory (pe baza la **RAM**)
5. **Multiplexor**
6. **Modul de extensie** cu pastrare semnelui
7. Modul de **siftare** la stânga cu 2 poziții
8. Modulul MIPSdataPath trebuie să fie **testat**
9. Modulul MIPSdataPath trebuie să fie **sentitezat**

ALU suportă următoarele coduri operațiilor:

| Codul | Funcția |
|-------|---------------|
| 010 | add |
| 110 | sub |
| 000 | and |
| 001 | or |
| 111 | set less than |

Exemplu de realizare a **modulului de extensie**:

```

module signext (a, y);
    input [15:0] a;
    input [15:0] y;
    assign y = {{16{a[15]}}, a};
endmodule
    
```