DOCUMENTAȚIE

MIPS PIPELINE

TEOCAN PATRICIA CLAUDIA

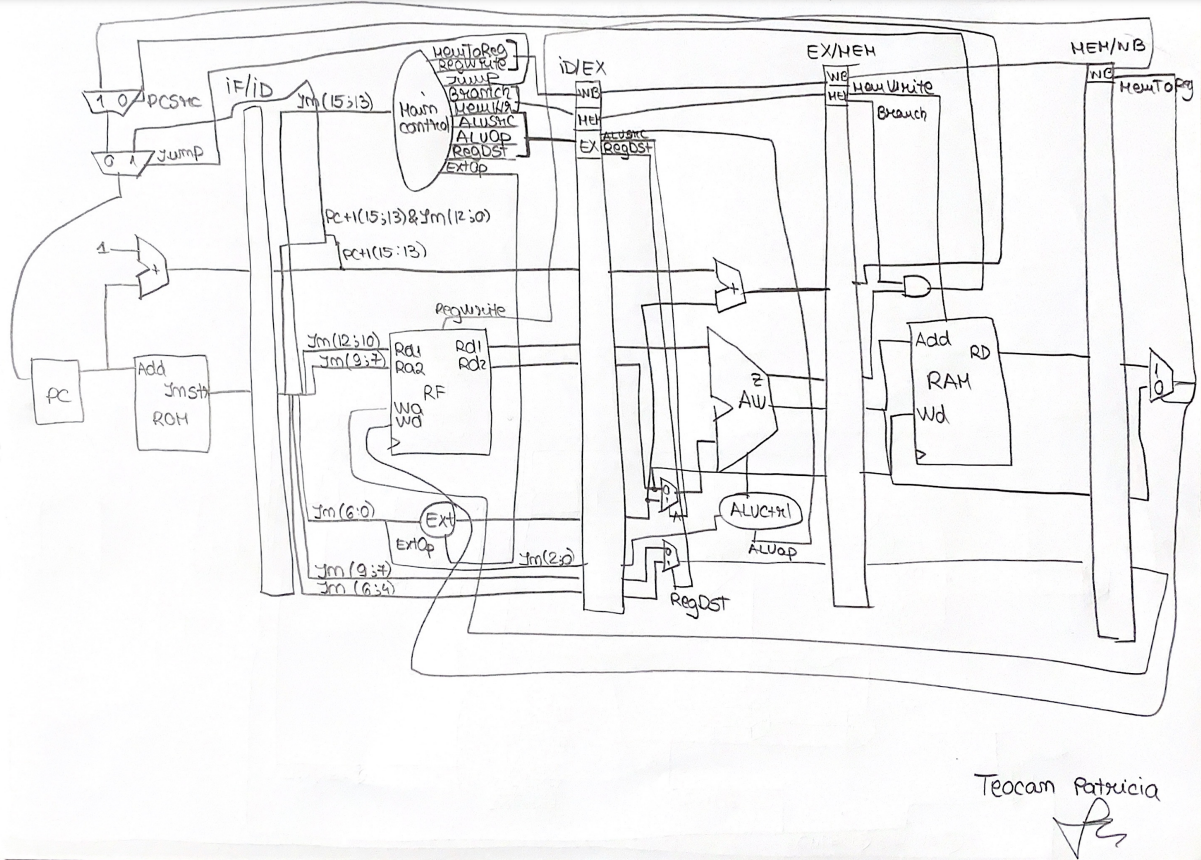
30225

**Descrierea programului scris si executat de procesor**

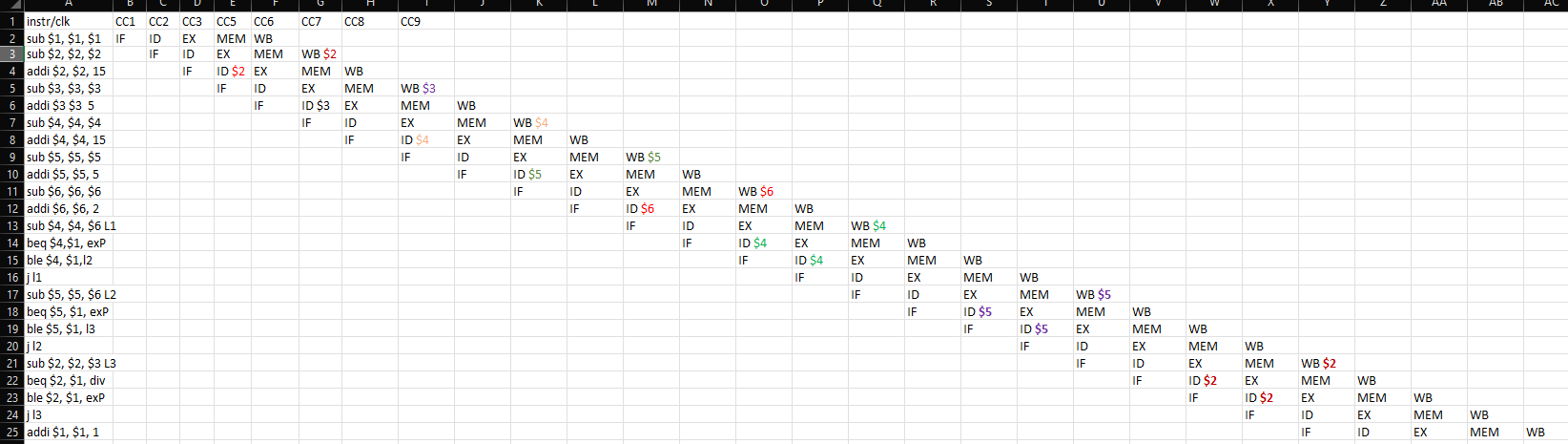
Circuitul Mips pipeline este programat sa determine daca doua numere sunt divizibile, cu conditia ca numerele sa fie impare ( se verifica ). Mai intai se verifica daca primul numar este multiplu de 2, lucru realizat prin scaderi repetate cu 2. Daca rezultatul este 0, se merge la finalul codului pentru ca numarul este par, daca este mai mare decat 0 se mai face o scadere, iar altfel se trece la verificarea celui de-al doilea numar pentru ca primul este impar. La fel se verifica si paritatea celui de-al doilea numar. Pentru a verifica daca cele 2 numere initiale sunt divizibile, se fac de asemenea scaderi repetate. Daca rezultatul este 0 se sare la ultima instructiune a codului, de a pune 1 intr-un registru pentru a semnala ca numerele sunt divizibile. Daca rezultatul este mai mic decat 0 se merge la finalul codului intrucat cele 2 numere nu sunt divizibile. Altfel, se mai face o scadere.

Modificările efectuate asupra mips-ului au constat în mutarea mux-ului ce selecteaza registrul din instrucțiune din zona ID in zona EX și transmiterea tuturor semnalelor de-a lungul fiecărui semnal de ceas.

**Schema procesorului mips pipeline**



**Analiza facută în laboratorul 10**



între intructiunile 2 si 3 este hazard de date dupa reg $2

între intructiunile 4 si 5 este hazard de date dupa reg $3

între intructiunile 6 si 7 este hazard de date dupa reg $4

între intructiunile 8 si 9 este hazard de date dupa reg $5

între intructiunile 10 si 11 este hazard de date dupa reg $6

între intructiunile 12 si 13 este hazard de date dupa reg $4

între intructiunile 12 si 14 este hazard de date dupa reg $4

între intructiunile 13 si 14 este hazard de control

între intructiunile 15 si 16 este hazard de control

între intructiunile 16 si 17 este hazard de date dupa reg $5

între intructiunile 16 si 18 este hazard de date dupa reg $5

între intructiunile 17 si 18 este hazard de control

între intructiunile 19 si 20 este hazard de control

între intructiunile 20 si 21 este hazard de date dupa reg $2

între intructiunile 20 si 22 este hazard de date dupa reg $2

între intructiunile 21 si 22 este hazard de control

între intructiunile 23 si 24 este hazard de control

**Codul după inserare de noop**

O imagine care conține masă

Descriere generată automat

O imagine care conține masă

Descriere generată automat

Întregul program a fost descris în VHDL și nu există erori de compilare.

O imagine care conține text, electronice

Descriere generată automat

Codul a fost testat pe placă și este funcțional parțial. Se executa instrucțiunile de inițializare, se execută prima buclă, însa există probleme la operatiile de tip Branch. Totodată, pentru a executa de mai multe ori bucla, am folosit jump-uri care sunt funționale.