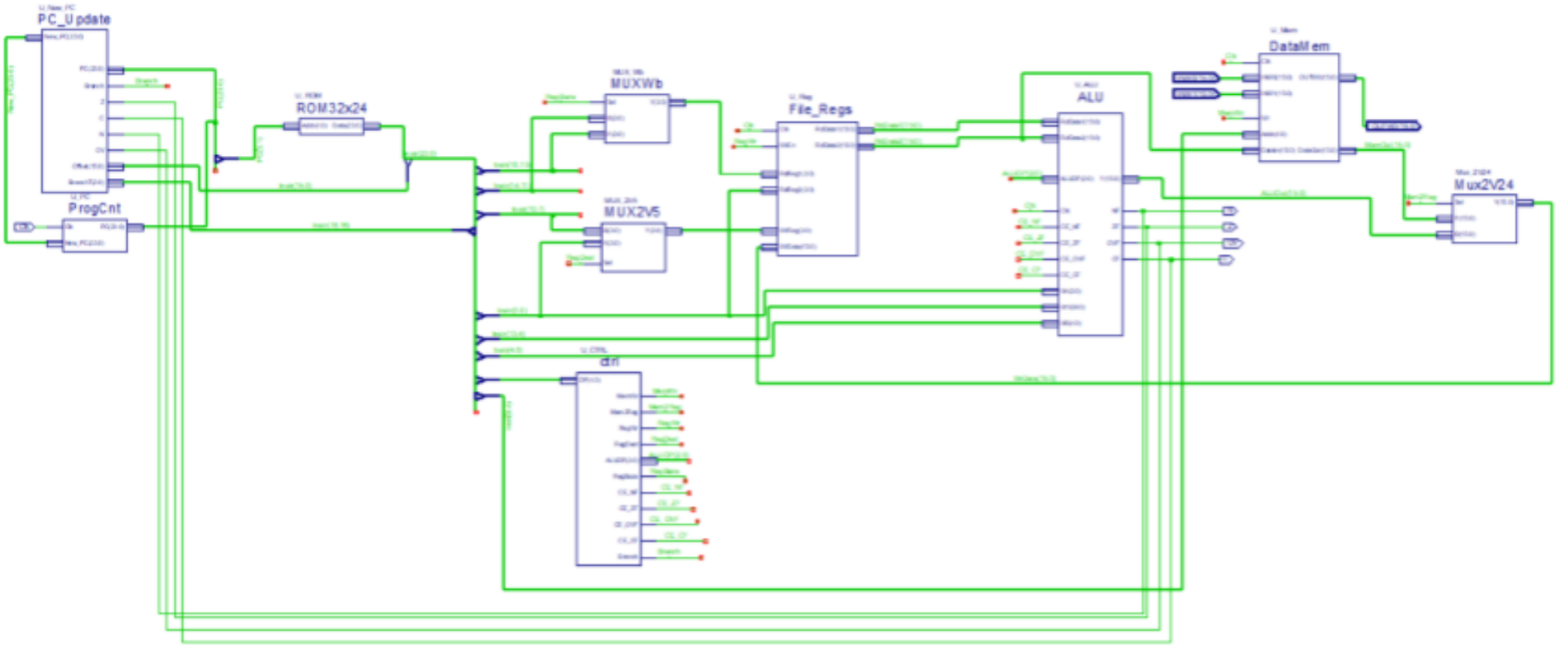
Filip Alina-Andreea
CR 3.2A

Proiectul 38 miniPIC24

- Schema procesorului pentru toate instructiunile:

Filip Alina-Andreea
CR 3.2A



1. Blocul ProgCnt are rolul de a schimba valoarea lui PC atunci cand apare un front crescator al semnalului CLK.
2. Blocul PC_Update incrementeaza valoarea anterioara a lui PC cu 2 pentru a trece la urmatoarea instructiune din memorie sau cu valoarea offset-ului atunci cand apare o instructiune branch. Blocul primeste tipul de branch al instructiunii dupa care verifica daca semnalul de branch si conditia (flag-urile sunt active pt instructiunea respectiva) de branch sunt indeplinite.
3. In blocul ROM se stocheaza programul pe care il executa procesorul. In functie de valoarea adresei citite din PC se extrage instructiunea de la acea adresa.
4. Blocul MUXWb extrage portiunea de instructiune in care se afla codul pentru valoarea registrului de baza ce trebuie citit. Folosim un mux deoarece unele instructiuni au codul pe o anumita portiune si altele pe o portiune diferita (in functie de valoarea semnalului RegBaza).

5. MUX2V5 face acelasi lucru ca MUXWb dar de aceasta data se extrage instructiunea pentru registrul destinatie in care se strie rezultatul instructiunii (in functie de valoarea semnalului RegDest).
6. Codul pentru registrul sursa se afla mereu printre ultimi 4 biti ai instructiunii iar el nu are nevoie de un mux.
7. Blocul File_Regs extrage din registri procesorului valorile de la adresele din instructiune si scrie atunci cand semnalul WrEn este activ valoarea rezultat la adresa de destinatie.
8. Blocul ALU efectueaza operatiile instructiunilor, calculeaza valorile flag-urilor si scoate la iesire rezultatul aferent. Blocul primeste semnalele de enable ale flag-urilor care ii spun ca instructiunea in lucru poate afecta valoarea flag-ului pt care enable este activ, de asemenea primeste ALUOP care ii spune ce operatie se executa pentru instructiunea data, un semnal Clk pentru flag-uri si lit4, lit5 si lit10 folosite pentru instructiunile specifice SL, ADDC si SUBB.
9. Blocul DataMem primeste valorile introduse INW0 si INW1, un semnal de write care ii spune daca se face o scriere in memorie, o adresa de scriere in memorie pe care o extragem din instructiune si o valoare ce trebuie scrisa in memorie (valoarea din registrul cerut). El fiseaza valoarea scrisa in memorie si trimite la iesire valoarea citita din memorie.
10. Blocul MUX2V24 selectreaza in functie de valoarea lui Mem2Reg una dintre intrarile sale oentru a determina daca se scrie din memorie in registru sau se scrie valoarea calculata de ALU.
11. Blocul ctrl stabileste toate semnalele din procesor care le comunica celorlalte blocuri ce trebuie sa faca in functie de opcode-ul instructiunii curente.

- Tabela de adevar a semnalelor generate de blocul de control:

	Opcode	CE_NF	CE_ZF	CE_OVF	CE_CF	ALUOP	MemWr	Mem2Reg	RegWr	RegDest	RegBaza	Branch
ADD Wb, Ws, Wd	01000	1	1	1	1	000	0	0	1	0	1	0
SUB Wb, Ws, Wd	01010	1	1	1	1	001	0	0	1	0	1	0
AND Wb, Ws, Wd	01100	1	1	0	0	010	0	0	1	0	1	0
IOR Wb, Ws, Wd	01110	1	1	0	0	011	0	0	1	0	1	0
MOV f, wnd	10000	0	0	0	0	000	0	1	1	1	1	0
MOV wns, f	10001	0	0	0	0	000	1	0	0	1	1	0
BRA Expr	00110	0	0	0	0	000	0	0	0	1	1	1
BRA OV, Expr	00110	0	0	0	0	000	0	0	0	1	1	1
BRA C, Expr	00110	0	0	0	0	000	0	0	0	1	1	1
BRA N, Expr	00110	0	0	0	0	000	0	0	0	1	1	1
BRA Z, Expr	00110	0	0	0	0	000	0	0	0	1	1	1
SL Wb,#lit4,Wnd	11011	1	1	0	0	100	0	0	1	0	0	0
ADDC #lit10,Wn	10110	1	1	1	1	101	0	0	1	1	1	0

Filip Alina-Andreea
CR 3.2A

SUBB Wb,#lit5,Wd	01011	1	1	1	1	110	0	0	1	0	1	0
SETM Wd	11101	0	0	0	0	000	0	0	1	0	1	0

1. Opcode-ul este extras din inctructiune de la 23 la 19, in functie de el se stabilesc celelalte semnale.
2. CE_NF, CE_ZF, CE_OVF si CE_CF sunt 1 cand instructiunea poate afecta valoarea flag-ului N, Z, OV, respectiv C.
3. ALUOP este folosit pentru a ii spune lui ALU ce operatie trebuie facuta.
4. MemWr este 1 daca are loc o scriere in memorie.
5. Mem2Reg este 1 daca are loc o scriere din memorie in registru
6. RegWr este 1 daca are loc o scriere in registri
7. RegDest este 1 cand adresa de destinatie se gaseste pe pozitia 3:0 a instructiunii si 0 daca se gaseste pe pozitia 10:7.
8. Reg Baza este 1 daca adresa registrului de baza se gaseste pe pozitia 18:15 in instructiune si 0 daca se gaseste pe portiunea 14:11.
9. Semnalul Branch este activ atunci cand avem o instructiune de branch si ii spune numaratorului de program ca s-ar putea executa un salt.

Pozitia pe care se afla valoarea adresei din ROM s-a identificat facand adunari succesive si observand pe ce pozitii apar valorile 0, 1, 2, etc, iar pozitia adresei din memorie a fost gasita identificand valorile adresei pentr registri speciali si identificarea acestor valori in instructiune. Restul valorilor extrase din instructiune cum ar fi lit4, lit 5 si lit10 au fost identificate folosind codificarea din manual a instructiunilor.