**PIC24 Proiect**

**Numarul 26**

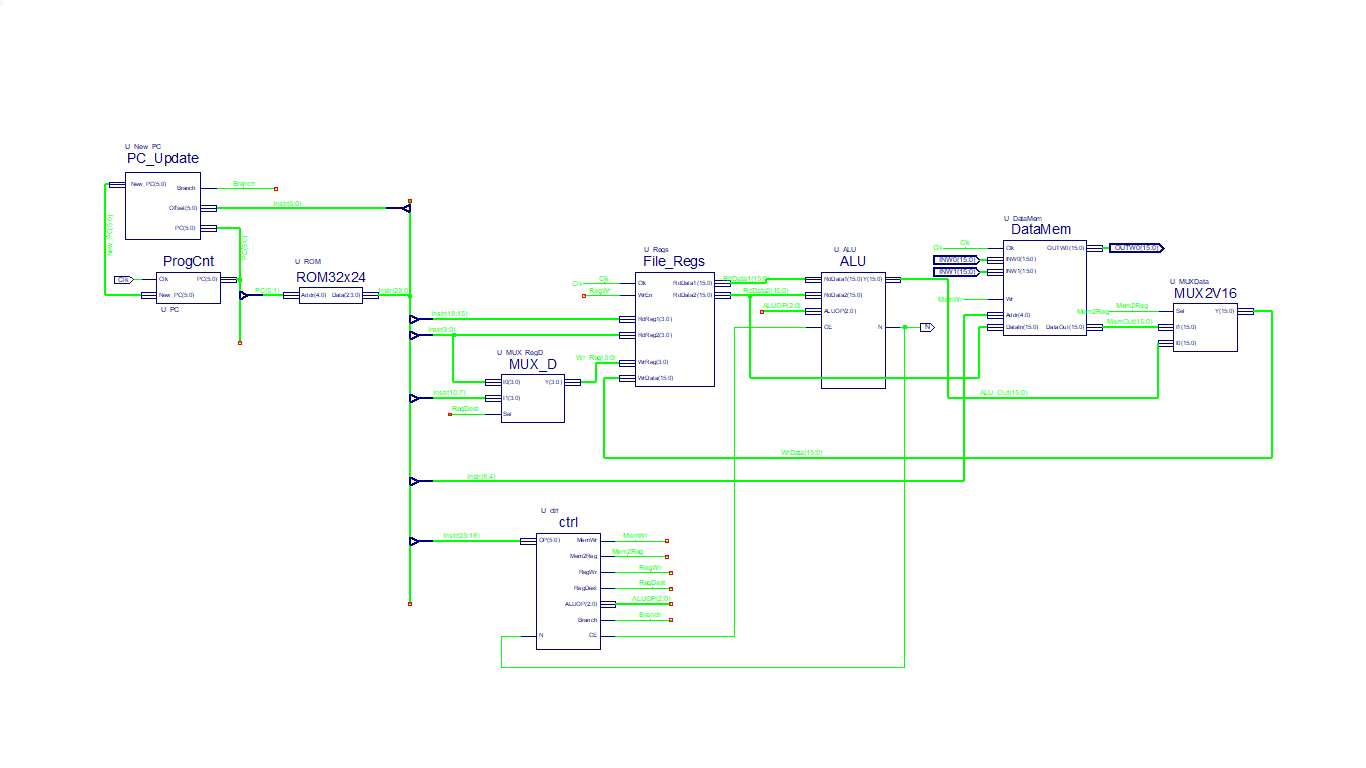
**Gheorge Octavian Petru**

**Voica Catalin Gabriel**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Encoding | 2222  3210 | 1111  9876 | 1111  5432 | 11  1098 | 7654 | 3210 | Flags |
| ADD | 0010 | 0www | wBqq | qddd | dppp | ssss | N,OV,Z,1 |
| SUB | 0101 | 0www | wBqq | qddd | dppp | ssss | N,OV,Z,C |
| AND | 0100 | 0www | wBqq | qddd | qppp | ssss | N ,Z |
| IOR | 0111 | 0www | wBqq | qddd | dppp | ssss | N,Z |
| MOV f,wnd | 1000 | Offff | ffff | ffff | ffff | dddd | none |
| MOV wns,  f | 1000 | 1fff | ffff | ffff | ffff | ssss | none |
| BRA expr | 0011 | 0111 | nnnn | nnnn | nnnn | nnnn | none |
| RLNC | 1101 | 0010 | 0Bqq | qddd | dppp | ssss | N,Z |
| BRA C  expr | 0011 | 0001 | nnnn | nnnn | nnnn | nnnn | none |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | OPCODE | ALUOP | Mem2Reg | MemWr | RegWr | RegDest | BRA | N | CE |
| ADD | 01000 | 000 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| SUB | 01010 | 001 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| AND | 01100 | 010 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| IOR | 01110 | 011 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| MOV f,Wnd | 10000 | 111 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| MOV Wns, f | 10001 | 111 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| BRA expr | 001101 | 111 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| RLNC | 11010 | 100 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| BRA N Expr | 001100 | 111 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Schema Logica:



**PC\_Update**

- gives the new PC value depending on:

* if is not a branch signal the value is PC + 2;
* if there is a branch instruction the offset is converted to integer and multiplied by 2 and after that added to PC + 2;

**ProgCnt**

- stores the New\_PC value when the clock is on rising edge;

**ROM32x24**

**-** in the ROM memory we store the assembly language;

- it has 5 bits for each of 24 instructions that it can store;

**File\_Regs**

- it is composed by 16 registers( W0 – W15);

- in this registers we can read or write data depending on the selected signals;

- there are 2 read registers and 1 write register because there are instructions that can read from 2 registers;

**Ctrl**

**-** this unit decides what is the next instruction for being executed;

- it gives value to each signal;

**ALU**

**-** this unit is responsible with the arithmetic and logic operations;

- the ALUOP signals decides the operation that is made with the two operands(RdData1 and RdData2);

- it computes also the flags like Carry;

**DataMem**

**-** this is the RAM memory block which has in his composition 16 location of 16 bits each one;

- reads the operands from 1020h(INW0 marker) and 1022h(INW1 marker) and writes in 1024h(OUTW0);

**MUX2v16**

**-** is a multiplexer which selects if data comes from the ALU or Memory.

**Signals Role**

Branch – sets ‘1’ the bit if the opcode equals the branch expression opcode else this is ‘0’.

Branch N – it sets the byte to one for branch if his opcode appears else if the opcode is not displayed,then the byte branch it`s zero for other opcodes.

Carry – is the value of the Carry flag generated by ALU while the operation is made.

CE – is the clock enable signal.

MemWr– is used to determine if it is the case to write or not in the memory.

RegWr – is used to determine if it is the case to write or not in the File\_Regs.

Mem2Reg – is used like a selection signal by the MUX2v16 to decide if the data is written in the ALU or Memory block.

RegDest – is used like a selection signal so it determine the position that takes the destination bits.

ALUOP – used for determine the operation type that is computed by ALU.