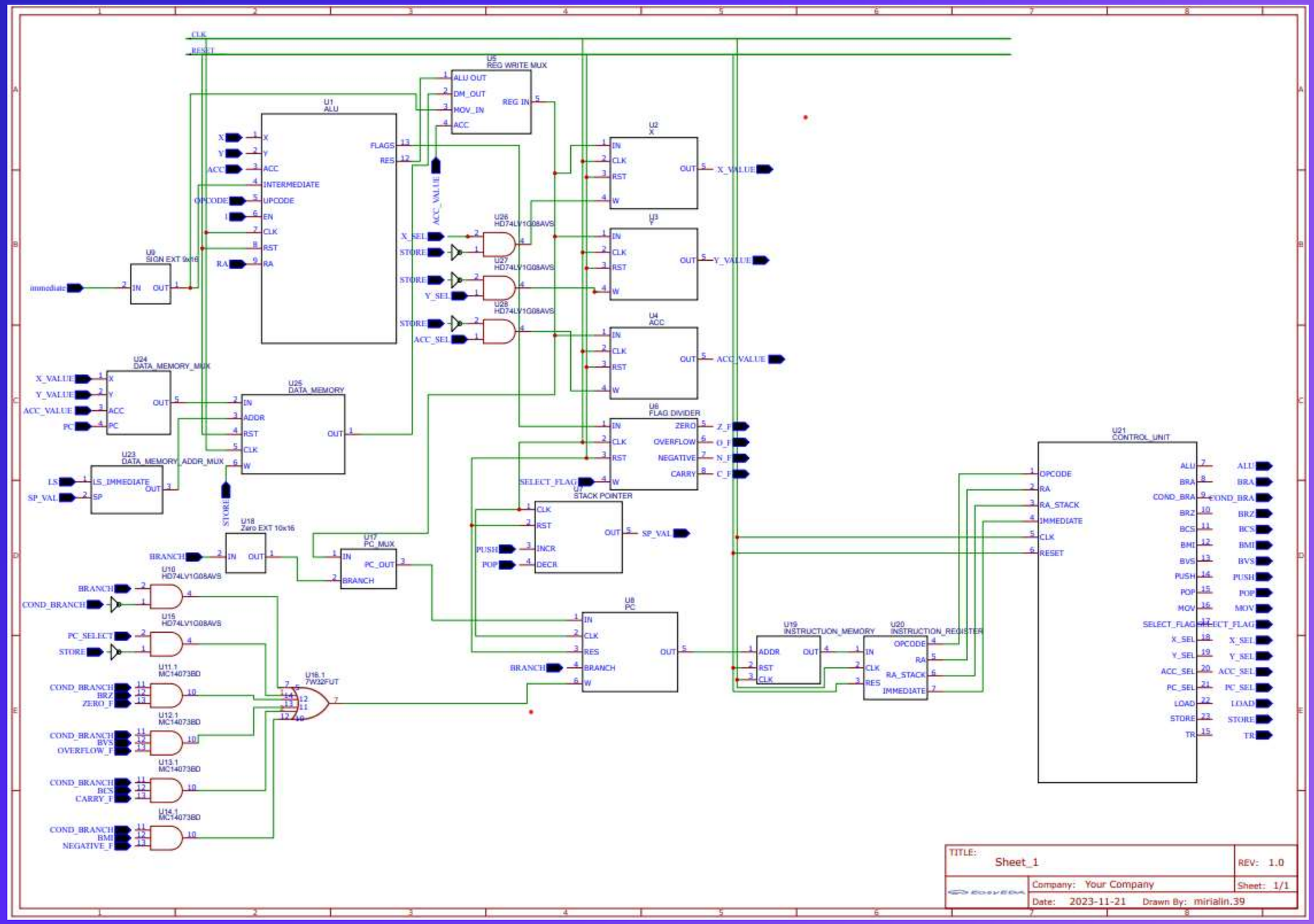
PROCESOR PE 16 BITI

ECHIPA: FIVE GANG

MEMBRII ECHIPEI

- Magdău Antonia-Laura
- Martinovici Iasmina-Patricia
- Maricutu Miriana-Dana
- Miri Marius-Alin
- Marta Ioan-Adrian

ARHIITECTURA



```
module X(
   input [15:0] in,
   input clk,rst,w,
   output reg [15:0] out);
```

- REPREZINTA UNUL DINTRE REGISTRE: X (LA FEL SUNT REALIZATE SI REGISTRUL Y SI ACUMULATORUL).

- REPREZINTA UN MODUL PT A EXTINDE:

ADRESA DE BRANCH

```
module ZE10x16(
   input [9:0] in,
   output reg [15:0] out);
```

VALOAREA IMEDIATA

```
module SE9x16(
   input [8:0] in,
   output reg [15:0] out);
```

- REPREZINTĂ MODULUL CARE DECIDE CE ADRESA SE PUNE IN PC

```
module PCInputDecider(
    input [15:0] POP_input,
    input [15:0] BRA_input,
    input STACK_POP, BRA,
    output reg [15:0] PC_in);
```

- REPREZINTĂ MODULUL PT IMPLEMENTAREA PROGRAM COUNTER-ULUI IN CADRUL PROCESORULUI.

```
module PC(
   input [15:0] in,
   input clk,rst,w,BRA,STACK_POP,
   output reg [15:0] out
);
```

- REPREZINTĂ MODULUL PT SETAREA FLAG-URILOR

```
module FLAGS(
  input [3:0] in,
  input clk,rst,w,
  output reg ZERO,NEGATIVE,CARRY,OVERFLOW
);
```

•REPREZINTA UN MODUL CARE ÎNDEPLINEȘTE ROLUL UNEI MEMORII DE DATE UTILIZATĂ PENTRU STOCAREA ȘI ACCESUL LA DATE

```
module DM(
  input signed [15:0] in,
  input [8:0] addr,
  input clk,rst,w,
  output reg signed [15:0] out);
```

```
module REG_DM_IN_MUX(
  input [15:0] in_X,
  input [15:0] in_Y,
  input [15:0] in_ACC,
  input [15:0] in_PC,
  input X_select,Y_select,ACC_select,PC_select,STORE,
  output reg [15:0] in);
```

- REPREZINTĂ UN MULTIPLEXOR PT A DETERMINA DIN CE REGISTRU VOM SALVA DATELE

```
module REG_WR_MUX(
  input [15:0] in_ALU,
  input [15:0] in_MOV,
  input [15:0] in_DM,
  input [15:0] in_ACC_TRANSFER,
  input ALU,MOV,LOAD,TR,
  output reg [15:0] in);
```

- REPREZINTĂ UN MULTIPLEXOR PT A DETERMINA DE UNDE SE SALVEAZA DATELE IN REGISTRII

```
module REG_DM_ADDRESS_MUX(
  input [8:0] in_LS_immediate,
  input [8:0] in_SP_val,
  input LOAD,STORE,STACK_PSH,STACK_POP,
  output reg [8:0] in);
```

- REPREZINTĂ UN MULTIPLEXOR PT A DETERMINA ADRESA LA CARE SA SCRIE SAU SA SE CITEASCA DIN MEMORIE

- REPREZINTĂ MEMORIA UNDE SUNT SALVATE INSTRUCTIUNIILE

```
module IM(
  input [9:0] addr,
  input rst,
  output reg [15:0] out);
```

```
module IR(
  input [15:0] in,
  input clk,rst,w,
  output reg [15:0] out,
  output reg [5:0] opcode,
  output reg RA,
  output reg [9:0] BA,
  output reg [8:0] IMM,
  output reg [1:0] RA_stack);
```

- REPREZINTĂ MODULUL CARE IMPARTE INSTRUCTIUNEA IN PARTI (OPCODE,Register Adress,Branch Adress,VALOAREA IMEDIATA,Register Adress Stiva)

- REPREZINTĂ MODULUL CARE CONTORIZEAZA ADRESA CURENTA IN STIVA

```
module SP(
  input clk,rst,inc,dec,
  output reg [15:0] out);
```

- REPREZINTĂ MODULUL CARE DECIDE ,IN FUNCTIE DE OPCODE, CE SEMNALE SA ACTIVEZE

```
module CU(
  input [5:0] opcode,
  input RA, clk, rst,
  input [1:0] RA_stack,
  input [8:0] Immediate,
  output reg ALU, BRA, COND_BRA, COND_BRA_Z,
  output COND_BRA_N, COND_BRA_C, COND_BRA_OF, LOAD,
  output STORE, TR, STACK_PSH,STACK_POP, MOV, flag_select,
  output ACC_select, X_select, Y_select,PC_select);
```

```
module ALU(
  input signed [15:0] ACC,X,Y,Immediate,
  input [5:0] opcode,
  input en,clk,rst,RA,
  output reg signed [15:0] res,
  output reg [3:0] flags);
```

- ARITHMETHIC LOGICAL UNIT, CARE EXECUTA OPERATIILE MATEMATICE SI LOGICE(ADUNARE, SCADERE, SHIFTARI ETC)

- REPREZINTĂ UN MODUL IN CARE SE VA REALIZA LEGATURA DINTRE TOATE CELELALTE MODULE

```
module top(
  input clk, rst);
```