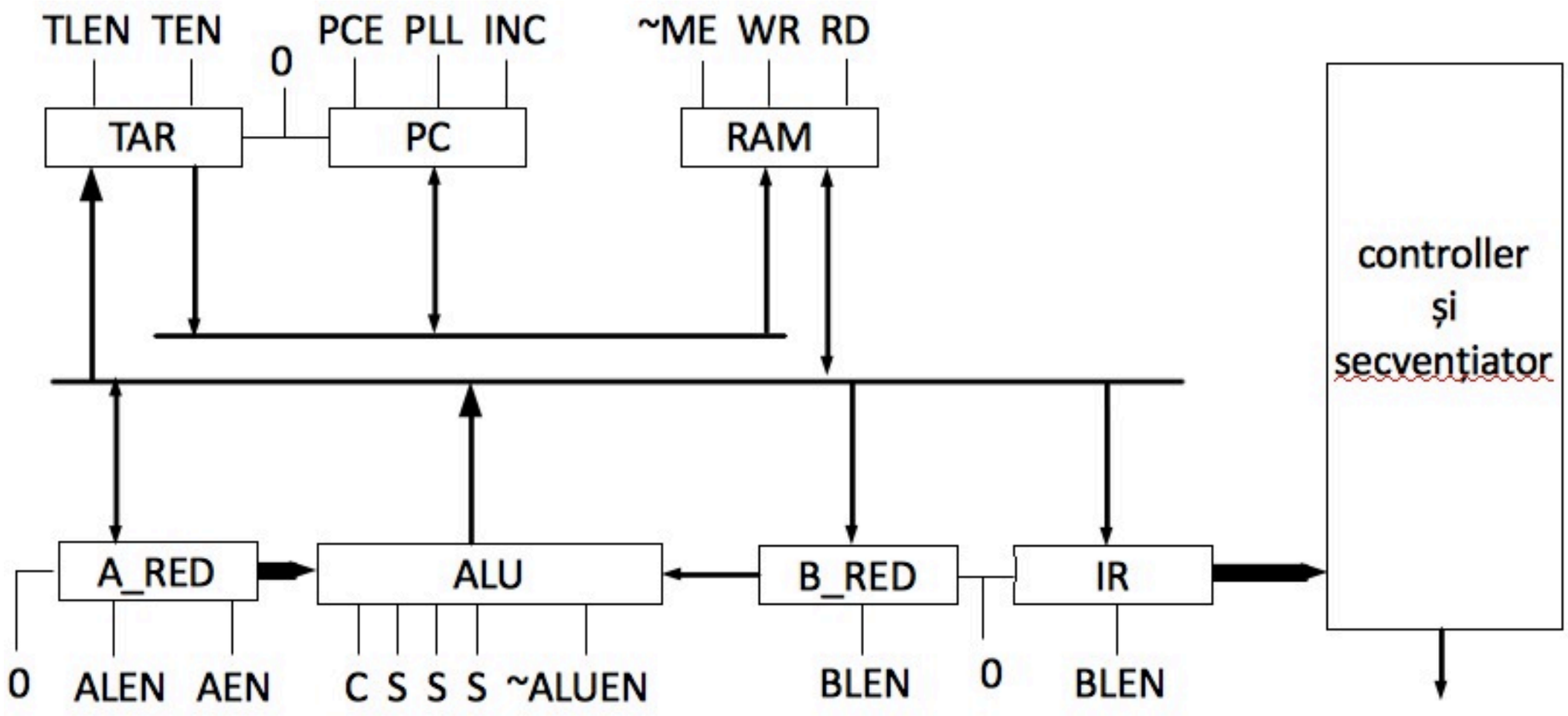
DESCRIEREA UNUI CALCULATOR SIMPLU ÎN VHDL

OBIECTIVE

- Descrierea procesorului
- Declararea pachetului folosit în descrierea VHDL a procesorului
- Descrierea comportamentală în VHDL
- Descrierea fluxului de date

Calea de date



Scopul - evitarea greșelilor

Definirea tipurilor de date folosite

```
1   -- Package File Template
2   --
3   -- Purpose: This package defines supplemental types, subtypes,
4   --      constants, and functions
5
6
7   library IEEE;
8   use IEEE.STD_LOGIC_1164.all;
9
10  package pack3 is
11
12
13    type bit3 is ('0','1','Z');
14    type bit_array is array (integer range <>) of bit3;
15    function resolved_fun (bitin:bit_array) return bit3;
16    subtype r_bit is resolved_fun bit3;
17    type tsv is array (integer range <>) of r_bit;
18    subtype r_word is tsv (3 downto 0);
19    type r_type is array (integer range 0 to 15) of r_word;
20
21  -- Declare constants
```

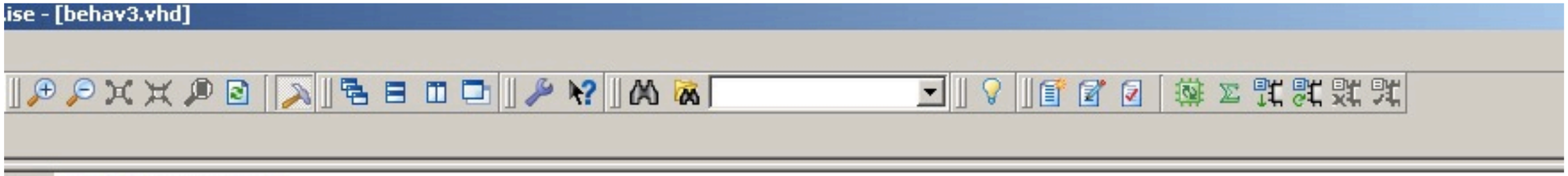
Scopul - evitarea greșelilor

Setul de instrucțiuni

Mnemonica	Cod Mașină	Operația	Observație
LDA aaaa	0001 aaaa	ACC <= Raaaa	încarcă acumulator
ADD aaaa	0010 aaaa	ACC <= ACC + Raaaa	C0=0; S1=S0=1; S2=0
SUB aaaa	0011 aaaa	ACC <= ACC - Raaaa	C0=1; S1=1; S0=S2=0
STA aaaa	0100 aaaa	Raaaa <= ACC	Memorează ACC în Raaaa
AND aaaa	0101 aaaa	ACC <= ACC & Raaaa	S2=S1=1; S0=C0=0;
OR aaaa	0110 aaaa	ACC <= ACC | Raaaa	S2=S0=1; S1=C0=0;
XOR aaaa	0111 aaaa	ACC <= ACC XOR Raaaa	S2=1; S0=S1=C0=0;
LDI dddd	1000 aaaa	ACC <= dddd	Încarcă date imediate
JUMP aaaa	1001 aaaa	PC <= aaaa	salt la instruct aaaa
JAN aaaa	1010 aaaa	If [ACC]<0, PC<= aaaa else next instr	jump la aaaa daca [ACC] negativ
INCA	1011	ACC <= ACC + 1	incrementare acc
SHAL	1100	ACC <= [A2A1A0]0	shift [ACC] left
CLA	1101	ACC <= 0000	șterge acc
CMA	1110	ACC <= [ACC]’	complem față de 1 al lui [ACC]
HALT	1111	stop execuție	

Calea de date -VHDL

ise - [behav3.vhd]



```
1   entity behav3 is
2   end behav3;
3
4   use work.pack3.all;
5
6   architecture Behavioral of behav3 is
7      signal P_C : r_word := "0000";
8      signal A_CC, IN_STR : r_word;
9      signal RAM:r_type := ("0001","1110","0100","1111","1111","1111","0110","1110","0111","1110","1110",
0                          "1100", "1101", "1111","1100","0001");
1   begin
2
3     process(P_C)
4        variable ir, b_reg, tar : r_word;
5        variable int_pc, int_tar, int_acc : integer;
```

Controlul-VHDL