Simulazione CPU – 1

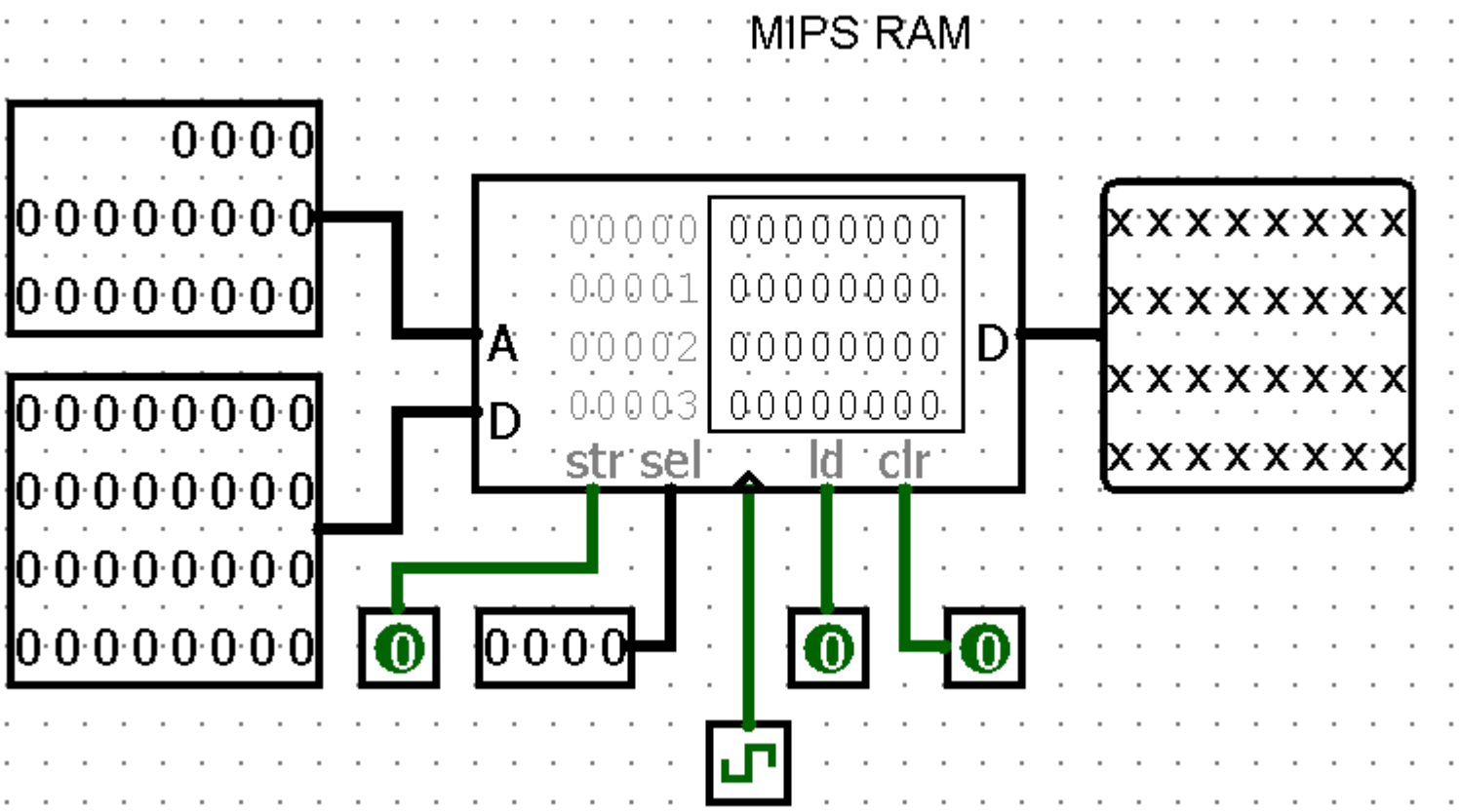
(componenti fondamentali)

Register File

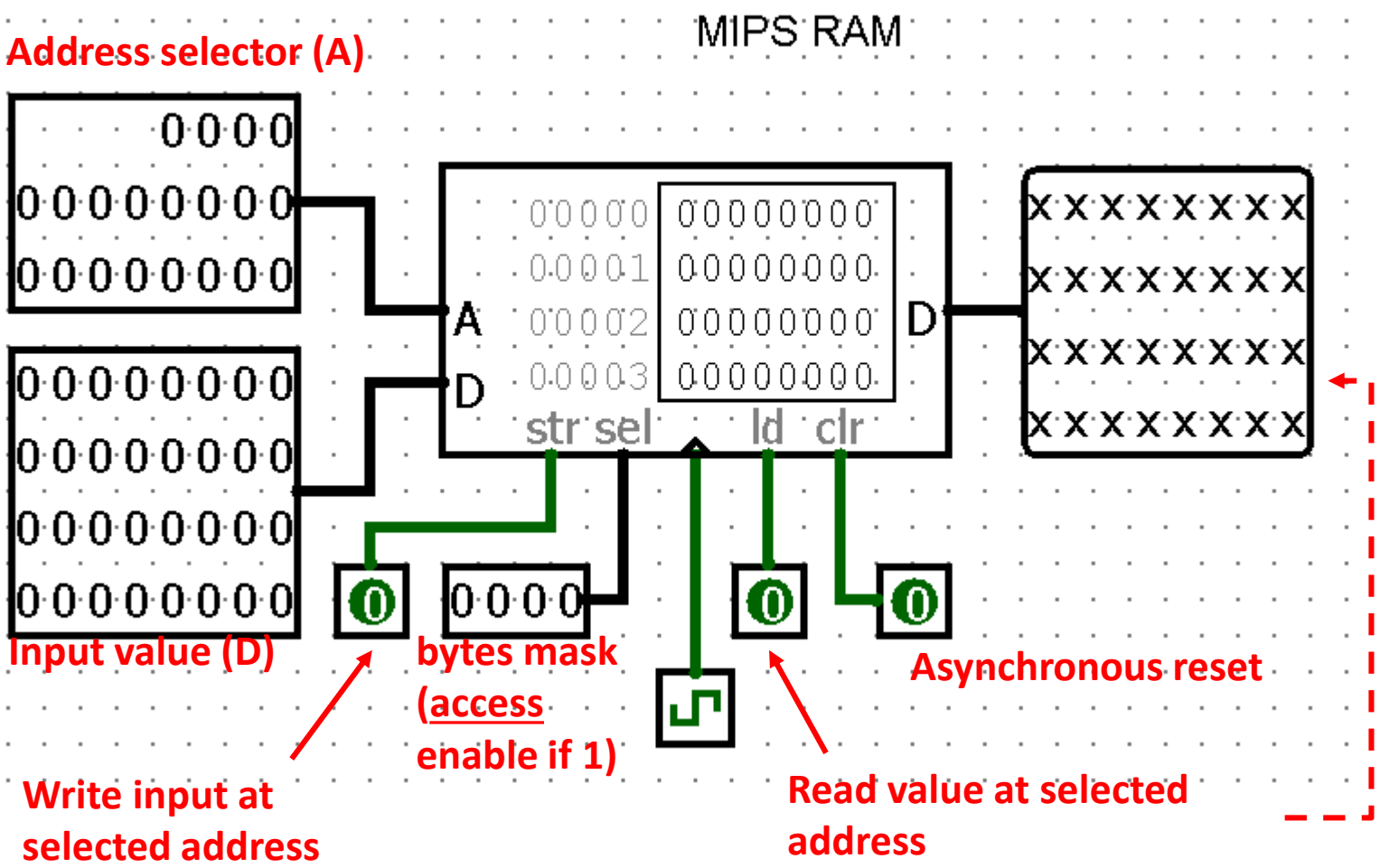
ALU

Memoria

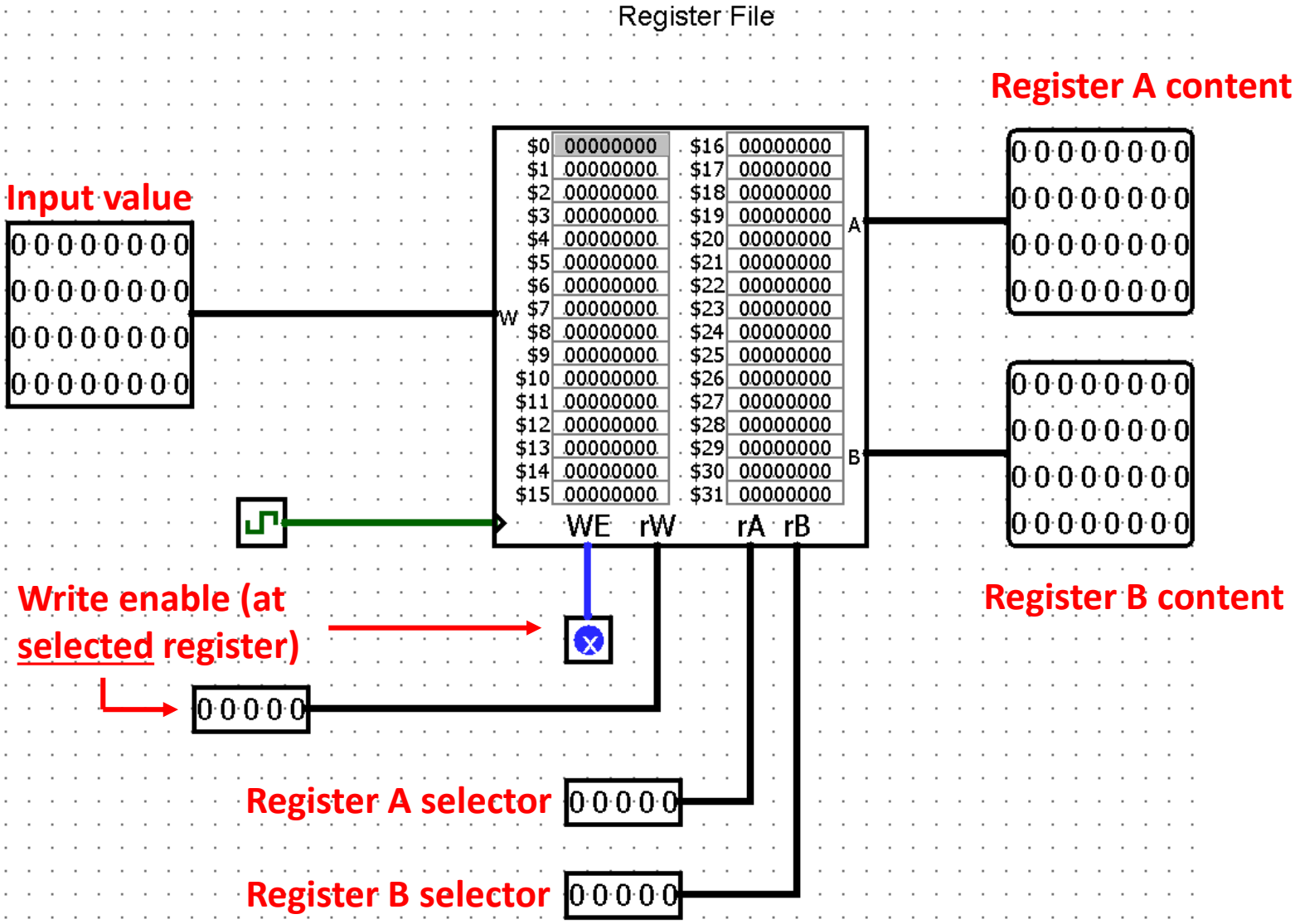
Memoria

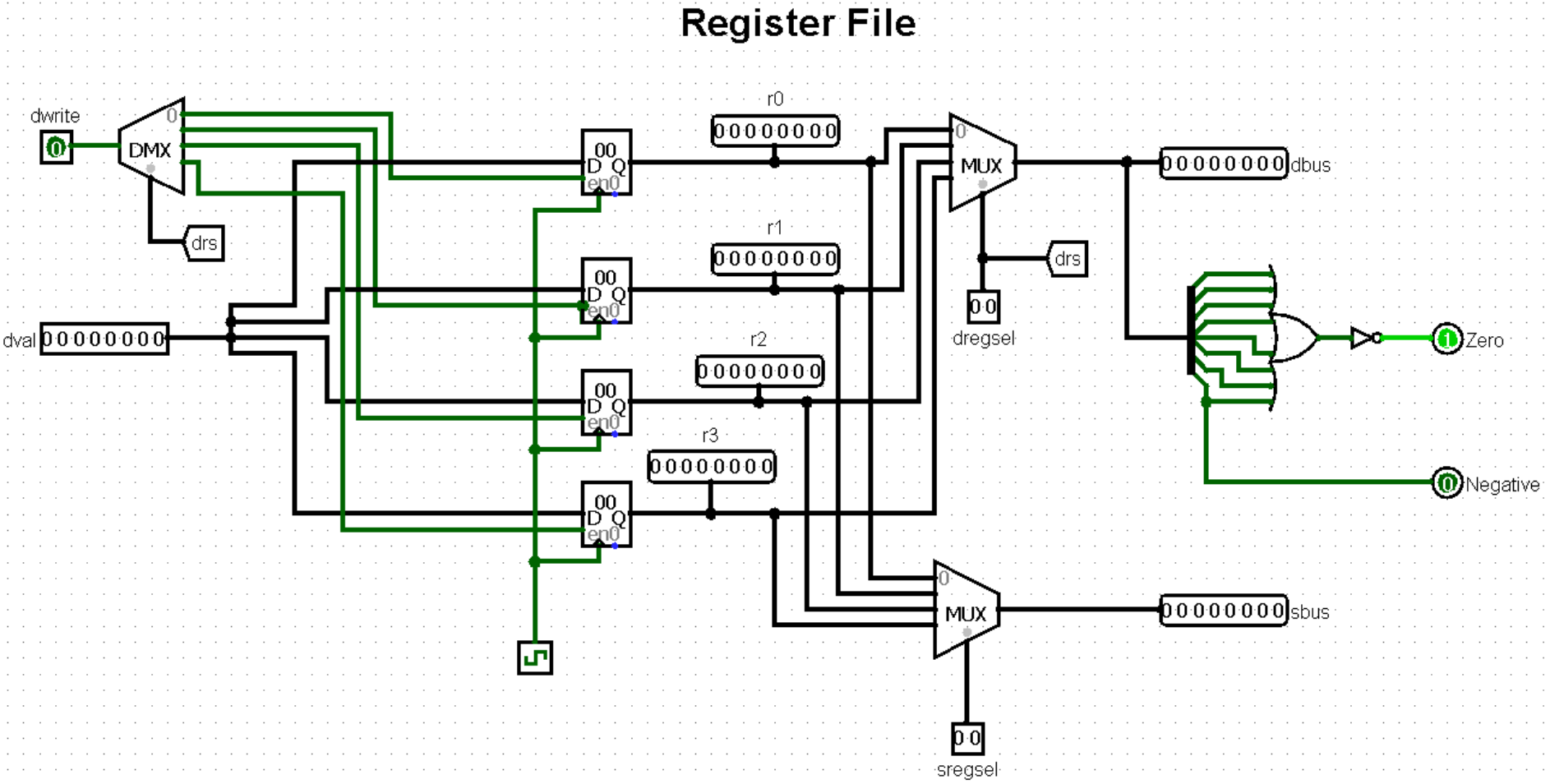

MIPS RAM
0000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
A
D
00000 00000000
00001 00000000
00002 00000000
00003 00000000
D
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
str sel ld clr
0
0000
0
0
5

Memoria

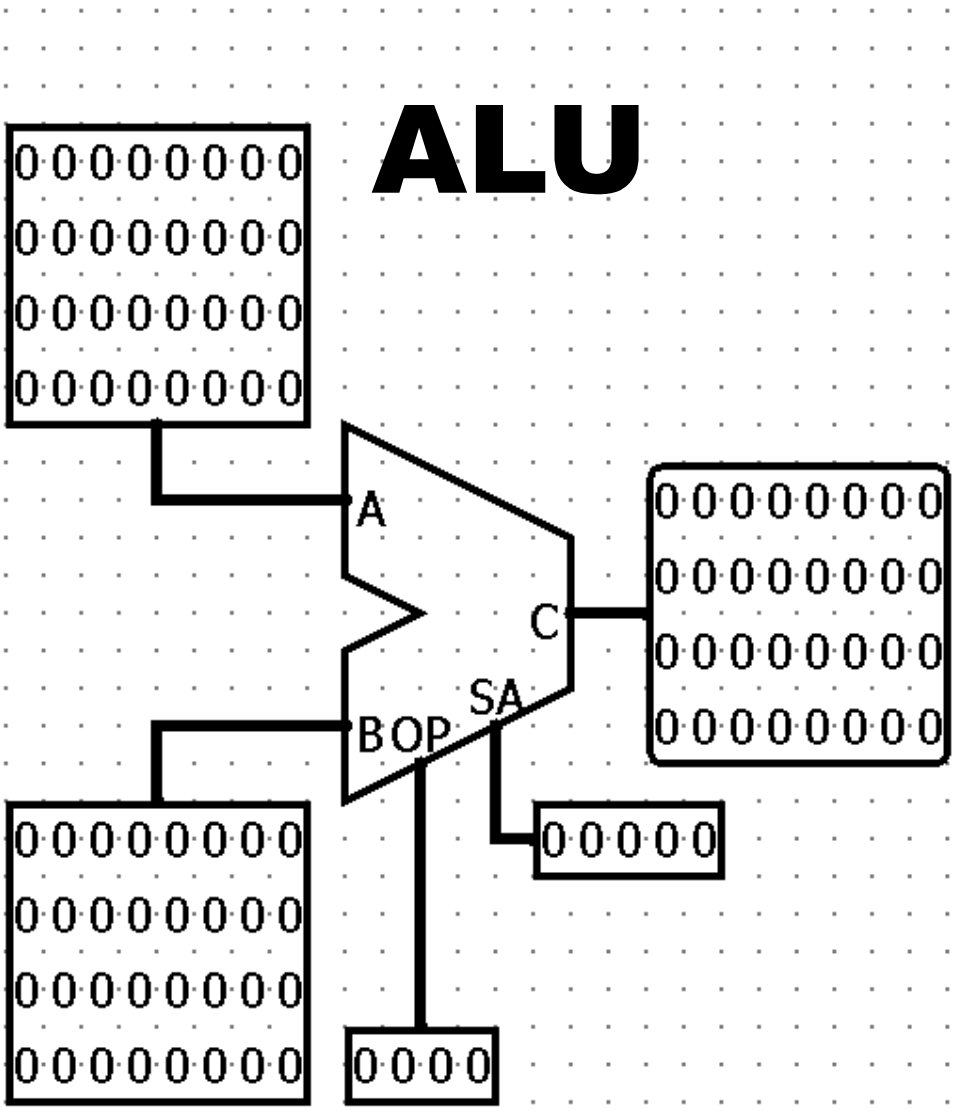

MIPS RAM
Address selector (A)
0000
00000000
00000000
00000000
00000000
00000000
00000000
A
D
00000 00000000
00001 00000000
00002 00000000
00003 00000000
D
str sel ld clr
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
0
0000
0
0
Input value (D)
bytes mask (access enable if 1)
Asynchronous reset
Write input at selected address
Read value at selected address

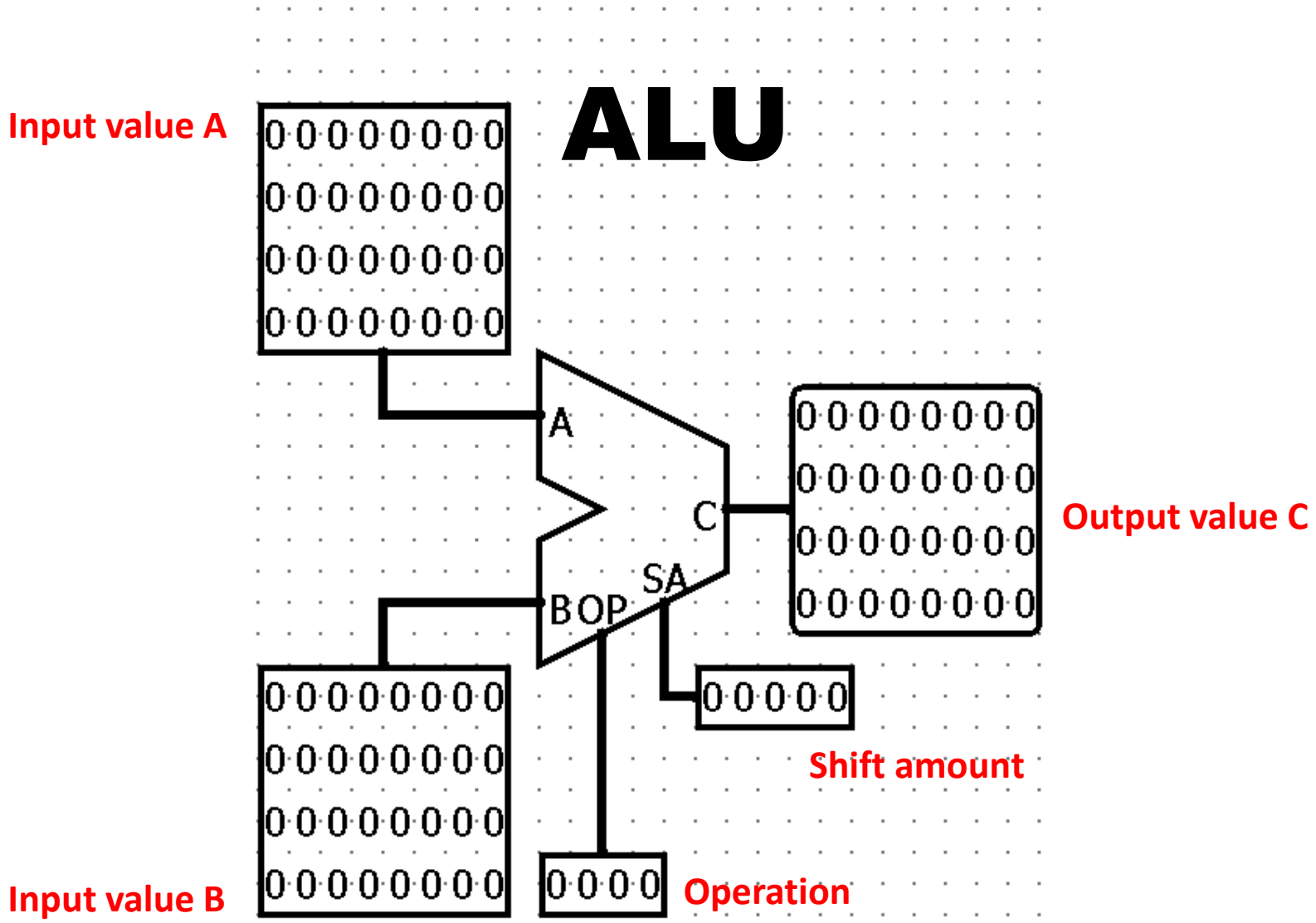
Register File


Register File
Register A content
Input value
$0 00000000
$1 00000000
$2 00000000
$3 00000000
$4 00000000
$5 00000000
$6 00000000
$7 00000000
$8 00000000
$9 00000000
$10 00000000
$11 00000000
$12 00000000
$13 00000000
$14 00000000
$15 00000000
$16 00000000
$17 00000000
$18 00000000
$19 00000000
$20 00000000
$21 00000000
$22 00000000
$23 00000000
$24 00000000
$25 00000000
$26 00000000
$27 00000000
$28 00000000
$29 00000000
$30 00000000
$31 00000000
W
A
B
WE
rW
rA
rB
Register B content
Write enable (at selected register)
00000
Register A selector
00000
Register B selector
00000



Implementazione mediante componenti libreria standard Logisim


ALU
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
A
B
OP
C
SA
0 0 0 0 0
0 0 0 0


ALU
Input value A
00000000
00000000
00000000
00000000
A
B
OP
C
SA
00000000
00000000
00000000
00000000
00000
0000
Output value C
Shift amount
Operation
Input value B

MIPS ALU. Computes a result as follows:

Op	name	C
000x	shift left	C = B << Sa;
001x	add	C = A + B
0100	shift right logical	C = B >>> Sa
0101	shift right arithmetic	C = B >> Sa
011x	subtract	C = A - B
1000	and	C = A & B
1010	or	C = A | B
1100	xor	C = A ^ B
1110	nor	C = ~(A | B)
1001	eq	C = (A == B) ? 1 : 0
1011	ne	C = (A != B) ? 1 : 0
1101	gtz	C = (A > 0) ? 1 : 0
1111	lez	C = (A ≤ 0) ? 1 : 0

Esercizio teoria:

Si traduca il seguente frammento di codice assembly MIPS in linguaggio macchina in formato esadecimale calcolando prima i valori esadecimali Loc1 e Loc2 che permettono di saltare esattamente all'indirizzo indicato nel commento corrispondente.

```
0xA00:      bne  $10,  $11,  Loc1  #salta a: 0x19B8
            j    Loc2              #salta a: 0x01234560
```