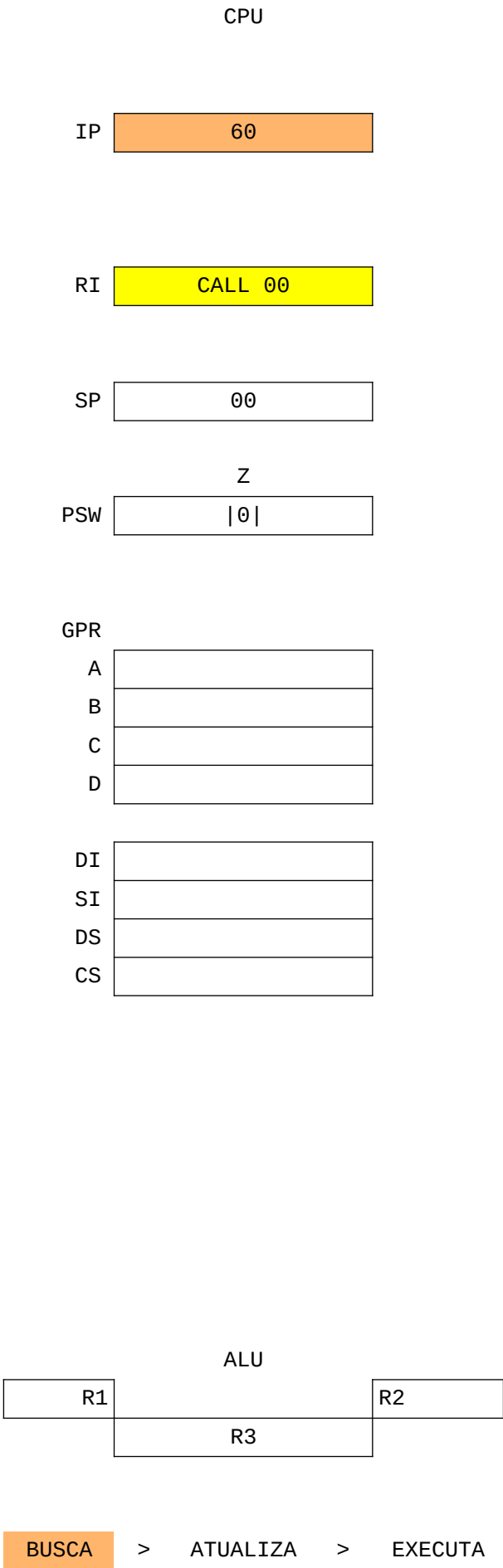


	MP
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	

BE

BD

BC (READ)



MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	00
RI	CALL 00
SP	FF
	Z
PSW	0
GPR	
A	
B	
C	
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

The diagram illustrates the state of a CPU, organized into several sections:

- CPU**: The main container for the state.
- IP (Instruction Pointer)**: A blue box containing the value `01`.
- RI (Register Index)**: A box containing the instruction `MOV A, 00`.
- SP (Stack Pointer)**: A box containing the value `FF`.
- PSW (Program Status Word)**: A box containing the value `| 0 |`, with a `Z` flag indicator above it.
- GPR (General Purpose Register)**: A section containing four empty boxes labeled `A`, `B`, `C`, and `D`.
- DI (Data Index)**, **SI (Source Index)**, **DS (Data Segment)**, and **CS (Code Segment)**: Four empty boxes at the bottom of the diagram.

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU. The output register R3 is shown as a larger rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	01
RI	MOV A, 00
SP	FF
Z	
PSW	0
GPR	
A	00
B	
C	
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) configuration. The ALU is shown above the register R3, receiving inputs from registers R1 and R2. The result of the operation is stored in register R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	01
RI	MOV B, 07
SP	FF
	Z
PSW	0
GPR	
A	00
B	
C	
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	02
RI	MOV B, 07
SP	FF
	Z
PSW	1
GPR	
A	00
B	
C	
D	
DI	
SI	
DS	
CS	

Diagram of a 3-operand ALU. It shows three registers: R1, R2, and R3. R1 and R2 are at the top, and R3 is below them. An ALU symbol is positioned above R1 and R2, with arrows pointing from R1 and R2 to the ALU. An arrow points from R3 to the ALU. The ALU output is connected to R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	02
RI	MOV B, 07
SP	FF
Z	
PSW	0
GPR	
A	00
B	07
C	
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	02
RI	MOV C, 03
SP	FF
PSW	Z 0
GPR	
A	00
B	07
C	
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The output register R3 is shown as a larger rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

	MP
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

BE

BD

BC

	CPU
IP	03
RI	MOV C, 03
SP	FF
	Z
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

	ALU	
R1		R2
	R3	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	03
RI	CALL 05
SP	FF
Z	
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with 'ALU' written above it. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The result register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	
FF	61

[illegible][illegible][illegible]

CPU	
IP	04
RI	CALL 05
SP	FF
Z	
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

	MP
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU

IP	04
RI	CALL 05
SP	FE
PSW	Z 0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central unit, with registers R1 and R2 providing inputs to it. The result of the ALU operation is stored in register R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	05
RI	CALL 05
SP	FE
	Z
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central unit with three registers (R1, R2, and R3) connected to it. R1 and R2 are positioned above the ALU, and R3 is positioned below it. The ALU is labeled "ALU" above it. Arrows indicate data flow from R1 and R2 into the ALU, and an output from R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	05
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
PSW	Z 0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU unit. It has two 16-bit registers, R1 and R2, which serve as inputs to the ALU. The ALU output is connected to a 32-bit register R3. The ALU is labeled "ALU" and has a "32-bit" label next to it. The registers R1, R2, and R3 are labeled "16-bit" and "32-bit" respectively.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	00
B	07
C	03
D	
DI	
SI	
DS	
CS	

ALU		
R1 < A	ADD	R2 < B
00	R3	07
07		

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	03
D	
DI	
SI	
DS	
CS	

ALU		
R1 < A	ADD	R2 < B
00	R3 > A	07
	07	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	07
B	07
C	03
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU structure. The ALU is positioned above the registers R1 and R2, which are connected to it. The result of the ALU operation is stored in register R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	07
B	07
C	03
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central unit with three registers (R1, R2, and R3) connected to it. R1 and R2 are positioned above the ALU, and R3 is positioned below it. The ALU is labeled "ALU" above it. Arrows indicate data flow from R1 and R2 into the ALU, and an output from R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	03
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
03	R3 > C	FF
02		

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	02
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
03	R3 > C	FF
	02	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	JNZ 05
SP	FE
PSW	Z 0
GPR	
A	07
B	07
C	02
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The output register R3 is shown as a rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	08
RI	JNZ 05
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	02
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central unit, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
PSW	Z 0
GPR	
A	07
B	07
C	02
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) configuration. The ALU is shown above the register R3, receiving inputs from registers R1 and R2. The ALU performs an operation on the values in R1 and R2, and the result is stored in R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	02
D	
DI	
SI	
DS	
CS	

ALU		
R1 < A	ADD	R2 < B
07	R3	07

14

BUSCA > ATUALIZA > EXECUTA

ALU		
R1 < A	ADD	R2 < B
07	R3 > A	07
	14	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	14
B	07
C	02
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	14
B	07
C	02
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU. The output register R3 is shown as a larger rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	14
B	07
C	02
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
02	R3 > C	FF
01		

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
Z	
PSW	0
GPR	
A	07
B	07
C	01
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
02	R3 > C	FF
	01	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	JNZ 05
SP	FE
Z	
PSW	0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

	MP
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	08
RI	JNZ 05
SP	FE
PSW	Z 0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is positioned above the register R3, receiving inputs from registers R1 and R2. The ALU performs operations on the data from R1 and R2, and the result is stored in R3.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	05
RI	JNZ 05
SP	FE
Z	
PSW	0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit registers, R1 and R2, which serve as inputs. The output is a 32-bit register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
PSW	Z 0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

ALU		
R1 < A	ADD	R2 < B
14	R3	07
21		

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible]

CPU	
IP	06
RI	ADD A, B
SP	FE
Z	
PSW	0
GPR	
A	21
B	07
C	01
D	
DI	
SI	
DS	
CS	

ALU		
R1 < A	ADD	R2 < B
14	R3 > A	07
	21	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	06
RI	DEC C
SP	FE
PSW	Z 0
GPR	
A	21
B	07
C	01
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU, respectively. The output register R3 is shown as a larger rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
Z	
PSW	0
GPR	
A	14
B	07
C	01
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
01	R3 > C	FF
00		

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	DEC C
SP	FE
PSW	Z 1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

ALU		
R1 < C	DEC	R2
01	R3 > C	FF
	00	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	07
RI	JNZ 05
SP	FE
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU. The output register R3 is shown as a rectangle at the bottom of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	08
RI	RET
SP	FE
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU. The output register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	09
RI	RET
SP	FE
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The registers R1 and R2 are positioned on the left and right sides of the ALU, respectively. The output register R3 is positioned below the ALU. The ALU is connected to the registers R1 and R2, and the output register R3.

BUSCA > ATUALIZA > EXECUTA

	MP
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	04
RI	RET
SP	FE
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 32-bit ALU (Arithmetic Logic Unit) structure. The ALU is represented by a large rectangle. Two 16-bit registers, R1 and R2, are shown as smaller rectangles on the left and right sides of the ALU. A 32-bit output register, R3, is shown below the ALU, connected to its bottom edge.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	04
RI	RET
SP	FF
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	04
RI	RET
SP	FF
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

The diagram shows a 32-bit ALU. It has two 16-bit input registers, R1 and R2, and a 32-bit output register R3. The ALU is represented by a large rectangle with the label 'ALU' at the top. The input registers R1 and R2 are shown as smaller rectangles on the left and right sides of the ALU. The output register R3 is shown as a larger rectangle at the bottom, spanning the width of the ALU.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	05
RI	RET
SP	FF
Z	
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

The diagram illustrates the state of various CPU registers. The registers are organized into several groups:

- IP (Instruction Pointer):** A single register containing the value 61.
- RI (Register Index):** A register containing the value RET, highlighted in red.
- SP (Stack Pointer):** A register containing the value 00, highlighted in red.
- PSW (Program Status Word):** A register containing the value | 1 |.
- Z (Zero Flag):** A single flag.
- GPR (General Purpose Registers):** A group of four registers labeled A, B, C, and D, each containing a two-digit hexadecimal value:
 - A: 21
 - B: 07
 - C: 00
 - D: (empty)
- DI (Data Index):** A register (empty).
- SI (Source Index):** A register (empty).
- DS (Data Segment):** A register (empty).
- CS (Code Segment):** A register (empty).

Diagram illustrating a 3-operand ALU (Arithmetic Logic Unit) structure. The ALU is shown as a central block, with registers R1 and R2 connected to its inputs, and register R3 connected to its output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	61
RI	MOV [64],A
SP	00
	Z
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU structure. The ALU is positioned above the registers R1 and R2, which are connected to the ALU. Register R3 is shown below R1 and R2, connected to the ALU output.

BUSCA > ATUALIZA > EXECUTA

MP	
00	MOV A, 00
01	MOV B, 07
02	MOV C, 03
03	CALL 05
04	RET
05	ADD A, B
06	DEC C
07	JNZ 05
08	RET
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
5B	
5C	
5D	
5E	
5F	
60	CALL 00
61	MOV [64], A
62	
63	
64	
STACK	
FB	
FC	
FD	
FE	04
FF	61

[illegible][illegible][illegible]

CPU	
IP	62
RI	MOV [64],A
SP	00
	Z
PSW	1
GPR	
A	21
B	07
C	00
D	
DI	
SI	
DS	
CS	

Diagram illustrating a 3-operand ALU structure. The ALU is positioned above the registers. The registers are labeled R1, R2, and R3. The ALU is connected to the outputs of R1 and R2, and its result is stored in R3.

BUSCA > ATUALIZA > EXECUTA

