

# Crappy CPU machine code equivalence

Antoine VIALON

6 janvier 2020

## Table des matières

<b>1</b>	<b>Registers</b>	<b>6</b>
<b>2</b>	<b>High level instructions</b>	<b>7</b>
2.1	ABRT	7
2.2	ADD	7
2.3	AND	7
2.4	CALL	7
2.5	CLR	7
2.6	CMP	8
2.7	DISP	8
2.8	HALT	8
2.9	INC	8
2.10	JMP	8
2.11	JMPBIT	8
2.12	JMPEQ	8
2.13	JMPGE	9
2.14	JMPGT	9
2.15	JMPLE	9
2.16	JMPLT	9
2.17	JMPNEQ	9
2.18	JMPPTR	9
2.19	LEDTGL	9
2.20	MOV	10
2.21	NEG	10
2.22	NOP	10
2.23	NOT	10
2.24	OR	10
2.25	RET	10
2.26	SHIFTL	10
2.27	SHIFTR	11

2.28 SLEEP . . . . .	11
2.29 SUB . . . . .	11
2.30 XOR . . . . .	11
<b>3 Instructions (low level) . . . . .</b>	<b>11</b>
3.1 default . . . . .	11
3.1.1 default : 0x00 . . . . .	11
3.2 ADD . . . . .	11
3.2.1 ADD_A_to_A : 0x01 . . . . .	11
3.2.2 ADD_A_to_U0 : 0x02 . . . . .	12
3.2.3 ADD_A_to_U1 : 0x03 . . . . .	12
3.2.4 ADD_A_to_U2 : 0x04 . . . . .	12
3.2.5 ADD_A_to_U3 : 0x05 . . . . .	12
3.2.6 ADD_A_to_mem : 0x06 . . . . .	12
3.2.7 ADD_B_to_A : 0x07 . . . . .	12
3.2.8 ADD_B_to_U0 : 0x08 . . . . .	13
3.2.9 ADD_B_to_U1 : 0x09 . . . . .	13
3.2.10 ADD_B_to_U2 : 0x0a . . . . .	13
3.2.11 ADD_B_to_U3 : 0x0b . . . . .	13
3.2.12 ADD_U0_to_A : 0x0c . . . . .	13
3.2.13 ADD_U0_to_mem : 0x0d . . . . .	13
3.2.14 ADD_U1_to_A : 0x0e . . . . .	14
3.2.15 ADD_U1_to_mem : 0x0f . . . . .	14
3.2.16 ADD_U2_to_A : 0x10 . . . . .	14
3.2.17 ADD_U2_to_mem : 0x11 . . . . .	14
3.2.18 ADD_U3_to_A : 0x12 . . . . .	14
3.2.19 ADD_U3_to_mem : 0x13 . . . . .	15
3.2.20 ADD_const_to_A : 0x14 . . . . .	15
3.2.21 ADD_const_to_U0 : 0x15 . . . . .	15
3.2.22 ADD_const_to_U1 : 0x16 . . . . .	15
3.2.23 ADD_const_to_U2 : 0x17 . . . . .	15
3.2.24 ADD_const_to_U3 : 0x18 . . . . .	16
3.2.25 ADD_const_to_const_in_A : 0x19 . . . . .	16
3.2.26 ADD_const_to_mem : 0x1a . . . . .	16
3.2.27 ADD_mem_to_A : 0x1b . . . . .	16
3.2.28 ADD_mem_to_U0 : 0x1c . . . . .	16
3.2.29 ADD_mem_to_U1 : 0x1d . . . . .	17
3.2.30 ADD_mem_to_U2 : 0x1e . . . . .	17
3.2.31 ADD_mem_to_U3 : 0x1f . . . . .	17
3.2.32 ADD_mem_to_mem : 0x20 . . . . .	17
3.3 AND . . . . .	18
3.3.1 AND_A_B_to_itself : 0x21 . . . . .	18
3.3.2 AND_U0_B_to_itself : 0x22 . . . . .	18
3.3.3 AND_U1_B_to_itself : 0x23 . . . . .	18

3.3.4	AND_U2_B_to_itself : 0x24	18
3.3.5	AND_U3_B_to_itself : 0x25	18
3.4	CALL	18
3.4.1	CALL_addr : 0x26	18
3.5	CMP	19
3.5.1	CMP_A_B : 0x27	19
3.5.2	CMP_A_U0 : 0x28	19
3.5.3	CMP_A_U1 : 0x29	19
3.5.4	CMP_A_U2 : 0x2a	19
3.5.5	CMP_A_U3 : 0x2b	19
3.5.6	CMP_A_const : 0x2c	20
3.5.7	CMP_A_mem : 0x2d	20
3.5.8	CMP_U0_A : 0x2e	20
3.5.9	CMP_U0_const : 0x2f	20
3.5.10	CMP_U0_mem : 0x30	21
3.5.11	CMP_U1_A : 0x31	21
3.5.12	CMP_U1_const : 0x32	21
3.5.13	CMP_U1_mem : 0x33	21
3.5.14	CMP_U2_A : 0x34	22
3.5.15	CMP_U2_const : 0x35	22
3.5.16	CMP_U2_mem : 0x36	22
3.5.17	CMP_U3_A : 0x37	22
3.5.18	CMP_U3_const : 0x38	23
3.5.19	CMP_U3_mem : 0x39	23
3.6	COPY	23
3.6.1	COPY : 0x3a	23
3.6.2	COPY_A_to_A : 0x3b	23
3.6.3	COPY_A_to_B : 0x3c	23
3.6.4	COPY_A_to_U0 : 0x3d	24
3.6.5	COPY_A_to_U1 : 0x3e	24
3.6.6	COPY_A_to_U2 : 0x3f	24
3.6.7	COPY_A_to_U3 : 0x40	24
3.6.8	COPY_A_to_cmp : 0x41	24
3.6.9	COPY_B_to_A : 0x42	24
3.6.10	COPY_B_to_B : 0x43	24
3.6.11	COPY_B_to_U0 : 0x44	25
3.6.12	COPY_B_to_U1 : 0x45	25
3.6.13	COPY_B_to_U2 : 0x46	25
3.6.14	COPY_B_to_U3 : 0x47	25
3.6.15	COPY_B_to_cmp : 0x48	25
3.6.16	COPY_U0_to_A : 0x49	25
3.6.17	COPY_U0_to_B : 0x4a	25
3.6.18	COPY_U0_to_cmp : 0x4b	26
3.6.19	COPY_U1_to_A : 0x4c	26

3.6.20	COPY_U1_to_B : 0x4d . . . . .	26
3.6.21	COPY_U1_to_cmp : 0x4e . . . . .	26
3.6.22	COPY_U2_to_A : 0x4f . . . . .	26
3.6.23	COPY_U2_to_B : 0x50 . . . . .	26
3.6.24	COPY_U2_to_cmp : 0x51 . . . . .	26
3.6.25	COPY_U3_to_A : 0x52 . . . . .	27
3.6.26	COPY_U3_to_B : 0x53 . . . . .	27
3.6.27	COPY_U3_to_cmp : 0x54 . . . . .	27
3.6.28	COPY_mem_to_cmp : 0x55 . . . . .	27
3.7	DISPLAY . . . . .	27
3.7.1	DISPLAY_A : 0x56 . . . . .	27
3.7.2	DISPLAY_B : 0x57 . . . . .	27
3.7.3	DISPLAY_U0 : 0x58 . . . . .	27
3.7.4	DISPLAY_U1 : 0x59 . . . . .	28
3.7.5	DISPLAY_U2 : 0x5a . . . . .	28
3.7.6	DISPLAY_U3 : 0x5b . . . . .	28
3.7.7	DISPLAY_mem : 0x5c . . . . .	28
3.8	FAIL . . . . .	28
3.8.1	FAIL : 0x5d . . . . .	28
3.9	HALT . . . . .	28
3.9.1	HALT : 0x5e . . . . .	28
3.10	INC . . . . .	29
3.10.1	INC_A : 0x5f . . . . .	29
3.10.2	INC_B : 0x60 . . . . .	29
3.10.3	INC_U0 : 0x61 . . . . .	29
3.10.4	INC_U1 : 0x62 . . . . .	29
3.10.5	INC_U2 : 0x63 . . . . .	29
3.10.6	INC_U3 : 0x64 . . . . .	29
3.10.7	INC_mem : 0x65 . . . . .	30
3.11	JMP . . . . .	30
3.11.1	JMP_const : 0x66 . . . . .	30
3.11.2	JMP_if_eq : 0x67 . . . . .	30
3.11.3	JMP_if_ge : 0x68 . . . . .	30
3.11.4	JMP_if_gt : 0x69 . . . . .	30
3.11.5	JMP_if_le : 0x6a . . . . .	31
3.11.6	JMP_if_lt : 0x6b . . . . .	31
3.11.7	JMP_if_neq : 0x6c . . . . .	31
3.11.8	JMP_ptr : 0x6d . . . . .	31
3.11.9	JMP_sel_bit_0 : 0x6e . . . . .	31
3.11.10	JMP_sel_bit_1 : 0x6f . . . . .	32
3.11.11	JMP_sel_bit_2 : 0x70 . . . . .	32
3.11.12	JMP_sel_bit_3 : 0x71 . . . . .	32
3.11.13	JMP_sel_bit_4 : 0x72 . . . . .	32
3.11.14	JMP_sel_bit_5 : 0x73 . . . . .	32

3.11.15	JMP_sel_bit_6 : 0x74	33
3.11.16	JMP_sel_bit_7 : 0x75	33
3.12	LED	33
3.12.1	LED_tgl : 0x76	33
3.13	LOAD	33
3.13.1	LOAD_const_to_A : 0x77	33
3.13.2	LOAD_const_to_B : 0x78	33
3.13.3	LOAD_const_to_U0 : 0x79	34
3.13.4	LOAD_const_to_U1 : 0x7a	34
3.13.5	LOAD_const_to_U2 : 0x7b	34
3.13.6	LOAD_const_to_U3 : 0x7c	34
3.13.7	LOAD_ptr_to_A : 0x7d	34
3.13.8	LOAD_ptr_to_B : 0x7e	34
3.13.9	LOAD_ptr_to_U0 : 0x7f	35
3.13.10	LOAD_ptr_to_U1 : 0x80	35
3.13.11	LOAD_ptr_to_U2 : 0x81	35
3.13.12	LOAD_ptr_to_U3 : 0x82	35
3.14	NEG	35
3.14.1	NEG_A : 0x83	35
3.14.2	NEG_B : 0x84	36
3.14.3	NEG_U0 : 0x85	36
3.14.4	NEG_U1 : 0x86	36
3.14.5	NEG_U2 : 0x87	36
3.14.6	NEG_U3 : 0x88	36
3.14.7	NEG_mem : 0x89	36
3.15	NOP	37
3.15.1	NOP : 0x8a	37
3.16	NOT	37
3.16.1	NOT_A : 0x8b	37
3.16.2	NOT_B : 0x8c	37
3.16.3	NOT_U0 : 0x8d	37
3.16.4	NOT_U1 : 0x8e	37
3.16.5	NOT_U2 : 0x8f	37
3.16.6	NOT_U3 : 0x90	38
3.17	OR	38
3.17.1	OR_A_B_to_itself : 0x91	38
3.17.2	OR_U0_B_to_itself : 0x92	38
3.17.3	OR_U1_B_to_itself : 0x93	38
3.17.4	OR_U2_B_to_itself : 0x94	38
3.17.5	OR_U3_B_to_itself : 0x95	38
3.18	RET	39
3.18.1	RET : 0x96	39
3.19	SHIFTL	39
3.19.1	SHIFTL_A : 0x97	39

3.19.2	SHIFTL_B : 0x98	39
3.19.3	SHIFTL_U0 : 0x99	39
3.19.4	SHIFTL_U1 : 0x9a	39
3.19.5	SHIFTL_U2 : 0x9b	39
3.19.6	SHIFTL_U3 : 0x9c	40
3.20	SHIFTR	40
3.20.1	SHIFTR_A : 0x9d	40
3.20.2	SHIFTR_B : 0x9e	40
3.20.3	SHIFTR_U0 : 0x9f	40
3.20.4	SHIFTR_U1 : 0xa0	40
3.20.5	SHIFTR_U2 : 0xa1	40
3.20.6	SHIFTR_U3 : 0xa2	41
3.21	SLEEP	41
3.21.1	SLEEP_A : 0xa3	41
3.21.2	SLEEP_B : 0xa4	41
3.21.3	SLEEP_U0 : 0xa5	41
3.21.4	SLEEP_U1 : 0xa6	41
3.21.5	SLEEP_U2 : 0xa7	41
3.21.6	SLEEP_U3 : 0xa8	41
3.21.7	SLEEP_const : 0xa9	42
3.21.8	SLEEP_mem : 0xaa	42
3.22	STORE	42
3.22.1	STORE_A_to_address : 0xab	42
3.22.2	STORE_B_to_address : 0xac	42

## 1 Registers

*Available registers :*

- **A** : multi-purpose register. Is not overwritten quietly.
- **B** : work register. Used in many operations as a buffer
- **U#** : user registers. Will **never** be overwritten unless *explicitely* mentionned (see the instructions for more detail). There are only 4 of those currently.
- **ret** : not directly accessible. Used with **CALL** and **RET**
- **cmp** : not directly accessible. Used with **CMP** and **JMPxxx**
- **disp** : used to display something in decimal. Write-only.
- **C** : single bit register

## 2 High level instructions

### 2.1 ABRT

*Set error bit and halt*

— ABRT (size : 1, duration : 3)

### 2.2 ADD

*Add a value from a register/memory address/const to a register or a memory address and save it in register A*

- ADD A, U# (size : 1, duration : 6)
- ADD R, B (size : 1, duration : 5)
- ADD R, A (size : 1, duration : 6)
- ADD R, @0xHH (size : 2, duration : 9)
- ADD R, #0xHH (size : 2, duration : 7)
- ADD @0xHH, @0xHH (size : 3, duration : 13)
- ADD @0xHH, R (size : 2, duration : 9)
- ADD #0xHH, #0xHH (size : 3, duration : 9)
- ADD @0xHH, #0xHH (size : 3, duration : 10)

### 2.3 AND

*AND two registers, save the result to the first operand*

- AND R, B (size : 1, duration : 5)

### 2.4 CALL

*Jump to specified address and save current PC in Ret register. Useful for subroutines.*

- CALL #0xHH (size : 2, duration : 7)

### 2.5 CLR

*Clear a register/mem address. Clears B if parameter is an address*

- CLR A (size : 1, duration : 6)
- CLR U# (size : 1, duration : 6)
- CLR @0xHH (size : 3, duration : 11)

## 2.6 CMP

*Compare two values (subtracts them) and store the result in CMP register. Overwrites B.*

- CMP R, @0xHH (size : 2, duration : 12)
- CMP A, B (size : 1, duration : 8)
- CMP A, U# (size : 1, duration : 8)
- CMP R, #0xHH (size : 2, duration : 10)
- CMP U#, A (size : 1, duration : 8)
- CMP R (size : 1, duration : 4)
- CMP @0xHH (size : 2, duration : 7)

## 2.7 DISP

*Display a value contained in specified register/memory address as an unsigned integer.*

- DISP R (size : 1, duration : 4)
- DISP @0xHH (size : 1, duration : 7)

## 2.8 HALT

*Halt the CPU*

- HALT (size : 1, duration : 3)

## 2.9 INC

*Increment register or value at memory address*

- INC R (size : 1, duration : 5)
- INC @0xHH (size : 2, duration : 8)

## 2.10 JMP

*Go to specified address*

- JMP @0xHH (size : 2, duration : 7)
- JMP #0xHH (size : 2, duration : 5)

## 2.11 JMPBIT

*Go to specified address if selected bit of comparison register is 1.*

- JMPBIT %b, #0xHH (size : 2, duration : 6)

## 2.12 JMPEQ

*Go to specified address if comparison register is zero.*

- JMPEQ #0xHH (size : 2, duration : 6)



### 2.13 JMPGE

*Go to specified address if comparison register is positive (or zero).*

— JMPGE #0xHH (size : 2, duration : 6)

### 2.14 JMPGT

*Go to specified address if comparison register is strictly positive.*

— JMPGT #0xHH (size : 2, duration : 6)

### 2.15 JMPLE

*Go to specified address if comparison register is negative or zero.*

— JMPLE #0xHH (size : 2, duration : 6)

### 2.16 JMPLT

*Go to specified address if comparison register is strictly negative.*

— JMPLT #0xHH (size : 2, duration : 6)

### 2.17 JMPNEQ

*Go to specified address if comparison register is NOT zero.*

— JMPNEQ #0xHH (size : 2, duration : 6)

### 2.18 JMPPTR

*Go to address value at memory address.*

— JMPPTR @0xHH (size : 2, duration : 7)

### 2.19 LEDTGL

*Toggle led. Useful for debugging.*

— LEDTGL (size : 1, duration : 4)

## 2.20 MOV

*Move a value from a register/memory address/const to a register or a memory address*

- MOV A, R (size : 1, duration : 4)
- MOV B, R (size : 1, duration : 4)
- MOV U#, A (size : 1, duration : 4)
- MOV U#, B (size : 1, duration : 4)
- MOV R, @0xHH (size : 2, duration : 7)
- MOV R, #0xHH (size : 2, duration : 5)
- MOV @0xHH, R (size : 2, duration : 6)
- MOV @0xHH, @0xHH (size : 3, duration : 10)
- MOV @0xHH, #0xHH (size : 3, duration : 7)

## 2.21 NEG

*Compute two's complement of register/memory (useful for substractions), and store result in itself.*

- NEG R (size : 1, duration : 6)
- NEG @0xHH (size : 2, duration : 9)

## 2.22 NOP

*Go to next address*

- NOP (size : 1, duration : 3)

## 2.23 NOT

*Invert bit by bit register, and store result in itself*

- NOT R (size : 1, duration : 5)

## 2.24 OR

*OR two registers, save the result to the first operand*

- OR A, B (size : 1, duration : 5)
- OR U#, B (size : 1, duration : 5)

## 2.25 RET

*Revert PC to value saved in Ret register. Use with CALL.*

- RET (size : 1, duration : 4)

## 2.26 SHIFTL

*Shift register to the left*

- SHIFTL R (size : 1, duration : 5)

## 2.27 SHIFTR

*Shift register to the right*

— SHIFTR R (size : 1, duration : 5)

## 2.28 SLEEP

*Pause clock for specified amount of ticks*

— SLEEP R (size : 1, duration : 4)

— SLEEP #0xHH (size : 1, duration : 5)

— SLEEP @0xHH (size : 1, duration : 7)

## 2.29 SUB

*Sub a value from a memory/register to register A and save it in register*

*A. Overwrites B.*

— SUB R, @0xHH (size : 2, duration : 12)

— SUB A, R (size : 1, duration : 9)

## 2.30 XOR

*XOR two registers, save the result to the first operand*

— XOR A, B (size : 1, duration : 5)

— XOR A, A (size : 1, duration : 6)

— XOR U#, U# (size : 1, duration : 6)

# 3 Instructions (low level)

## 3.1 default

### 3.1.1 default : 0x00

*Micro-instructions :*

- |                   |                                      |
|-------------------|--------------------------------------|
| 1. outPC, loadRAM | 2. outRAM, loadInstruction,<br>incPC |
|-------------------|--------------------------------------|

## 3.2 ADD

### 3.2.1 ADD\_A\_to\_A : 0x01

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outA, loadB              | 3. outALU, loadA |
| 2. enableAdd, loadALU, outA | 4. clearMIconter |

### 3.2.2 ADD\_\_A\_\_to\_\_U0 : 0x02

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outA, loadB               | 3. outALU, loadU0 |
| 2. enableAdd, loadALU, outU0 | 4. clearMIconter  |

### 3.2.3 ADD\_\_A\_\_to\_\_U1 : 0x03

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outA, loadB               | 3. outALU, loadU1 |
| 2. enableAdd, loadALU, outU1 | 4. clearMIconter  |

### 3.2.4 ADD\_\_A\_\_to\_\_U2 : 0x04

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outA, loadB               | 3. outALU, loadU2 |
| 2. enableAdd, loadALU, outU2 | 4. clearMIconter  |

### 3.2.5 ADD\_\_A\_\_to\_\_U3 : 0x05

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outA, loadB               | 3. outALU, loadU3 |
| 2. enableAdd, loadALU, outU3 | 4. clearMIconter  |

### 3.2.6 ADD\_\_A\_\_to\_\_mem : 0x06

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outA |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, storeRAM         |
| 3. loadRAM, outMemAddr        |                             |
| 4. outRAM, loadB              | 7. clearMIconter            |

### 3.2.7 ADD\_\_B\_\_to\_\_A : 0x07

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. enableAdd, loadALU, outA | 3. clearMIconter |
| 2. outALU, loadA            |                  |

### 3.2.8 ADD\_B\_to\_U0 : 0x08

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableAdd, loadALU, outU0 | 3. clearMIconter |
| 2. outALU, loadU0            |                  |

### 3.2.9 ADD\_B\_to\_U1 : 0x09

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableAdd, loadALU, outU1 | 3. clearMIconter |
| 2. outALU, loadU1            |                  |

### 3.2.10 ADD\_B\_to\_U2 : 0x0a

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableAdd, loadALU, outU2 | 3. clearMIconter |
| 2. outALU, loadU2            |                  |

### 3.2.11 ADD\_B\_to\_U3 : 0x0b

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableAdd, loadALU, outU3 | 3. clearMIconter |
| 2. outALU, loadU3            |                  |

### 3.2.12 ADD\_U0\_to\_A : 0x0c

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outU0, loadB             | 3. outALU, loadA |
| 2. enableAdd, loadALU, outA | 4. clearMIconter |

### 3.2.13 ADD\_U0\_to\_mem : 0x0d

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU0 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, storeRAM          |
| 3. loadRAM, outMemAddr        | 7. clearMIconter             |
| 4. outRAM, loadB              |                              |

### 3.2.14 ADD\_U1\_to\_A : 0x0e

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outU1, loadB             | 3. outALU, loadA |
| 2. enableAdd, loadALU, outA | 4. clearMCounter |

### 3.2.15 ADD\_U1\_to\_mem : 0x0f

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU1 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, storeRAM          |
| 3. loadRAM, outMemAddr        |                              |
| 4. outRAM, loadB              | 7. clearMCounter             |

### 3.2.16 ADD\_U2\_to\_A : 0x10

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outU2, loadB             | 3. outALU, loadA |
| 2. enableAdd, loadALU, outA | 4. clearMCounter |

### 3.2.17 ADD\_U2\_to\_mem : 0x11

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU2 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, storeRAM          |
| 3. loadRAM, outMemAddr        |                              |
| 4. outRAM, loadB              | 7. clearMCounter             |

### 3.2.18 ADD\_U3\_to\_A : 0x12

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outU3, loadB             | 3. outALU, loadA |
| 2. enableAdd, loadALU, outA | 4. clearMCounter |

### 3.2.19 ADD\_U3\_to\_mem : 0x13

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU3 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, storeRAM          |
| 3. loadRAM, outMemAddr        | 7. clearMIconter             |
| 4. outRAM, loadB              |                              |

### 3.2.20 ADD\_const\_to\_A : 0x14

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outPC, loadRAM           | 4. outALU, loadA |
| 2. outRAM, loadB, incPC     |                  |
| 3. enableAdd, loadALU, outA | 5. clearMIconter |

### 3.2.21 ADD\_const\_to\_U0 : 0x15

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outPC, loadRAM            | 4. outALU, loadU0 |
| 2. outRAM, loadB, incPC      |                   |
| 3. enableAdd, loadALU, outU0 | 5. clearMIconter  |

### 3.2.22 ADD\_const\_to\_U1 : 0x16

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outPC, loadRAM            | 4. outALU, loadU1 |
| 2. outRAM, loadB, incPC      |                   |
| 3. enableAdd, loadALU, outU1 | 5. clearMIconter  |

### 3.2.23 ADD\_const\_to\_U2 : 0x17

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outPC, loadRAM            | 4. outALU, loadU2 |
| 2. outRAM, loadB, incPC      |                   |
| 3. enableAdd, loadALU, outU2 | 5. clearMIconter  |

### 3.2.24 ADD\_const\_to\_U3 : 0x18

*Micro-instructions :*

- |                              |                   |
|------------------------------|-------------------|
| 1. outPC, loadRAM            | 4. outALU, loadU3 |
| 2. outRAM, loadB, incPC      |                   |
| 3. enableAdd, loadALU, outU3 | 5. clearMIconter  |

### 3.2.25 ADD\_const\_to\_const\_in\_A : 0x19

*Micro-instructions :*

- |                         |                             |
|-------------------------|-----------------------------|
| 1. outPC, loadRAM       | 5. enableAdd, loadALU, outA |
| 2. outRAM, loadA, incPC | 6. outALU, loadA            |
| 3. outPC, loadRAM       |                             |
| 4. outRAM, loadB, incPC | 7. clearMIconter            |

### 3.2.26 ADD\_const\_to\_mem : 0x1a

*Micro-instructions :*

- |                               |                                      |
|-------------------------------|--------------------------------------|
| 1. outPC, loadRAM             | 6. enableAdd, loadALU, outRAM, incPC |
| 2. outRAM, loadMemAddr, incPC |                                      |
| 3. loadRAM, outMemAddr        | 7. outALU, storeRAM                  |
| 4. outRAM, loadB              |                                      |
| 5. outPC, loadRAM             | 8. clearMIconter                     |

### 3.2.27 ADD\_mem\_to\_A : 0x1b

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outA |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, loadA            |
| 3. loadRAM, outMemAddr        |                             |
| 4. outRAM, loadB              | 7. clearMIconter            |

### 3.2.28 ADD\_mem\_to\_U0 : 0x1c

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU0 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, loadU0            |
| 3. loadRAM, outMemAddr        |                              |
| 4. outRAM, loadB              | 7. clearMIconter             |



### 3.2.29 ADD\_\_mem\_\_to\_\_U1 : 0x1d

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU1 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, loadU1            |
| 3. loadRAM, outMemAddr        | 7. clearMIconter             |
| 4. outRAM, loadB              |                              |

### 3.2.30 ADD\_\_mem\_\_to\_\_U2 : 0x1e

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU2 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, loadU2            |
| 3. loadRAM, outMemAddr        | 7. clearMIconter             |
| 4. outRAM, loadB              |                              |

### 3.2.31 ADD\_\_mem\_\_to\_\_U3 : 0x1f

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. enableAdd, loadALU, outU3 |
| 2. outRAM, loadMemAddr, incPC | 6. outALU, loadU3            |
| 3. loadRAM, outMemAddr        | 7. clearMIconter             |
| 4. outRAM, loadB              |                              |

### 3.2.32 ADD\_\_mem\_\_to\_\_mem : 0x20

*Micro-instructions :*

- |                         |                             |
|-------------------------|-----------------------------|
| 1. outPC, loadRAM       | 7. outMemAddr, loadRAM      |
| 2. outRAM, loadMemAddr  | 8. outRAM, loadB, incPC     |
| 3. outMemAddr, loadRAM  | 9. loadALU, enableAdd, outA |
| 4. outRAM, loadA, incPC | 10. outALU, storeRAM        |
| 5. outPC, loadRAM       | 11. clearMIconter           |
| 6. outRAM, loadMemAddr  |                             |

### 3.3 AND

#### 3.3.1 AND\_\_A\_\_B\_\_to\_\_itself : 0x21

*Micro-instructions :*

1. enableAND, loadALU, outA
2. outALU, loadA
3. clearMICounter

#### 3.3.2 AND\_\_U0\_\_B\_\_to\_\_itself : 0x22

*Micro-instructions :*

1. enableAND, loadALU, outU0
2. outALU, loadU0
3. clearMICounter

#### 3.3.3 AND\_\_U1\_\_B\_\_to\_\_itself : 0x23

*Micro-instructions :*

1. enableAND, loadALU, outU1
2. outALU, loadU1
3. clearMICounter

#### 3.3.4 AND\_\_U2\_\_B\_\_to\_\_itself : 0x24

*Micro-instructions :*

1. enableAND, loadALU, outU2
2. outALU, loadU2
3. clearMICounter

#### 3.3.5 AND\_\_U3\_\_B\_\_to\_\_itself : 0x25

*Micro-instructions :*

1. enableAND, loadALU, outU3
2. outALU, loadU3
3. clearMICounter

### 3.4 CALL

#### 3.4.1 CALL\_\_addr : 0x26

*Micro-instructions :*

1. outPC, loadRAM
2. outRAM, loadMemAddr, incPC
3. outPC, loadRet
4. loadPC, cond\_always, outMemAddr
5. clearMICounter

## 3.5 CMP

### 3.5.1 CMP\_\_A\_\_B : 0x27

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outB, enableNOT, loadALU   | 4. enableAdd, loadALU, outA |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp          |
| 3. outALU, loadB              | 6. clearMIconter            |

### 3.5.2 CMP\_\_A\_\_U0 : 0x28

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outU0, enableNOT, loadALU  | 4. enableAdd, loadALU, outA |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp          |
| 3. outALU, loadB              | 6. clearMIconter            |

### 3.5.3 CMP\_\_A\_\_U1 : 0x29

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outU1, enableNOT, loadALU  | 4. enableAdd, loadALU, outA |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp          |
| 3. outALU, loadB              | 6. clearMIconter            |

### 3.5.4 CMP\_\_A\_\_U2 : 0x2a

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outU2, enableNOT, loadALU  | 4. enableAdd, loadALU, outA |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp          |
| 3. outALU, loadB              | 6. clearMIconter            |

### 3.5.5 CMP\_\_A\_\_U3 : 0x2b

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outU3, enableNOT, loadALU  | 4. enableAdd, loadALU, outA |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp          |
| 3. outALU, loadB              | 6. clearMIconter            |

### 3.5.6 CMP\_\_A\_\_const : 0x2c

*Micro-instructions :*

- |                               |                             |
|-------------------------------|-----------------------------|
| 1. outPC, loadRAM             | 5. outALU, loadB            |
| 2. outRAM, loadB, incPC       | 6. enableAdd, loadALU, outA |
| 3. outB, enableNOT, loadALU   | 7. outALU, loadCmp          |
| 4. outALU, enableInc, loadALU | 8. clearMIconter            |

### 3.5.7 CMP\_\_A\_\_mem : 0x2d

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 6. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 7. outALU, loadB              |
| 3. outMemAddr, loadRAM        | 8. enableAdd, loadALU, outA   |
| 4. outRAM, loadB              | 9. outALU, loadCmp            |
| 5. outB, enableNOT, loadALU   | 10. clearMIconter             |

### 3.5.8 CMP\_\_U0\_\_A : 0x2e

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outA, enableNOT, loadALU   | 4. enableAdd, loadALU, outU0 |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp           |
| 3. outALU, loadB              | 6. clearMIconter             |

### 3.5.9 CMP\_\_U0\_\_const : 0x2f

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. outALU, loadB             |
| 2. outRAM, loadB, incPC       | 6. enableAdd, loadALU, outU0 |
| 3. outB, enableNOT, loadALU   | 7. outALU, loadCmp           |
| 4. outALU, enableInc, loadALU | 8. clearMIconter             |

### 3.5.10 CMP\_\_U0\_\_mem : 0x30

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 6. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 7. outALU, loadB              |
| 3. outMemAddr, loadRAM        | 8. enableAdd, loadALU, outU0  |
| 4. outRAM, loadB              | 9. outALU, loadCmp            |
| 5. outB, enableNOT, loadALU   | 10. clearMIconter             |

### 3.5.11 CMP\_\_U1\_\_A : 0x31

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outA, enableNOT, loadALU   | 4. enableAdd, loadALU, outU1 |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp           |
| 3. outALU, loadB              | 6. clearMIconter             |

### 3.5.12 CMP\_\_U1\_\_const : 0x32

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. outALU, loadB             |
| 2. outRAM, loadB, incPC       | 6. enableAdd, loadALU, outU1 |
| 3. outB, enableNOT, loadALU   | 7. outALU, loadCmp           |
| 4. outALU, enableInc, loadALU | 8. clearMIconter             |

### 3.5.13 CMP\_\_U1\_\_mem : 0x33

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 6. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 7. outALU, loadB              |
| 3. outMemAddr, loadRAM        | 8. enableAdd, loadALU, outU1  |
| 4. outRAM, loadB              | 9. outALU, loadCmp            |
| 5. outB, enableNOT, loadALU   | 10. clearMIconter             |

#### 3.5.14 CMP\_\_U2\_\_A : 0x34

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outA, enableNOT, loadALU   | 4. enableAdd, loadALU, outU2 |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp           |
| 3. outALU, loadB              | 6. clearMIconter             |

#### 3.5.15 CMP\_\_U2\_\_const : 0x35

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. outALU, loadB             |
| 2. outRAM, loadB, incPC       | 6. enableAdd, loadALU, outU2 |
| 3. outB, enableNOT, loadALU   | 7. outALU, loadCmp           |
| 4. outALU, enableInc, loadALU | 8. clearMIconter             |

#### 3.5.16 CMP\_\_U2\_\_mem : 0x36

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 6. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 7. outALU, loadB              |
| 3. outMemAddr, loadRAM        | 8. enableAdd, loadALU, outU2  |
| 4. outRAM, loadB              | 9. outALU, loadCmp            |
| 5. outB, enableNOT, loadALU   | 10. clearMIconter             |

#### 3.5.17 CMP\_\_U3\_\_A : 0x37

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outA, enableNOT, loadALU   | 4. enableAdd, loadALU, outU3 |
| 2. outALU, enableInc, loadALU | 5. outALU, loadCmp           |
| 3. outALU, loadB              | 6. clearMIconter             |

### 3.5.18 CMP\_U3\_const : 0x38

*Micro-instructions :*

- |                               |                              |
|-------------------------------|------------------------------|
| 1. outPC, loadRAM             | 5. outALU, loadB             |
| 2. outRAM, loadB, incPC       | 6. enableAdd, loadALU, outU3 |
| 3. outB, enableNOT, loadALU   | 7. outALU, loadCmp           |
| 4. outALU, enableInc, loadALU | 8. clearMIconter             |

### 3.5.19 CMP\_U3\_mem : 0x39

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 6. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 7. outALU, loadB              |
| 3. outMemAddr, loadRAM        | 8. enableAdd, loadALU, outU3  |
| 4. outRAM, loadB              | 9. outALU, loadCmp            |
| 5. outB, enableNOT, loadALU   | 10. clearMIconter             |

## 3.6 COPY

### 3.6.1 COPY : 0x3a

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 5. outPC, loadRAM             |
| 2. outRAM, loadMemAddr, incPC | 6. outRAM, loadMemAddr, incPC |
| 3. outMemAddr, loadRAM        | 7. storeRAM, outB             |
| 4. outRAM, loadB              | 8. clearMIconter              |

### 3.6.2 COPY\_A\_to\_A : 0x3b

*Micro-instructions :*

- |                |                  |
|----------------|------------------|
| 1. outA, loadA | 2. clearMIconter |
|----------------|------------------|

### 3.6.3 COPY\_A\_to\_B : 0x3c

*Micro-instructions :*

- |                |                  |
|----------------|------------------|
| 1. outA, loadB | 2. clearMIconter |
|----------------|------------------|

#### **3.6.4 COPY\_A\_to\_U0 : 0x3d**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outA, loadU0 | 2. clearMIconter |
|-----------------|------------------|

#### **3.6.5 COPY\_A\_to\_U1 : 0x3e**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outA, loadU1 | 2. clearMIconter |
|-----------------|------------------|

#### **3.6.6 COPY\_A\_to\_U2 : 0x3f**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outA, loadU2 | 2. clearMIconter |
|-----------------|------------------|

#### **3.6.7 COPY\_A\_to\_U3 : 0x40**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outA, loadU3 | 2. clearMIconter |
|-----------------|------------------|

#### **3.6.8 COPY\_A\_to\_cmp : 0x41**

*Micro-instructions :*

- |                  |                  |
|------------------|------------------|
| 1. outA, loadCmp | 2. clearMIconter |
|------------------|------------------|

#### **3.6.9 COPY\_B\_to\_A : 0x42**

*Micro-instructions :*

- |                |                  |
|----------------|------------------|
| 1. outB, loadA | 2. clearMIconter |
|----------------|------------------|

#### **3.6.10 COPY\_B\_to\_B : 0x43**

*Micro-instructions :*

- |                |                  |
|----------------|------------------|
| 1. outB, loadB | 2. clearMIconter |
|----------------|------------------|



### **3.6.11 COPY\_B\_to\_U0 : 0x44**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outB, loadU0 | 2. clearMIconter |
|-----------------|------------------|

### **3.6.12 COPY\_B\_to\_U1 : 0x45**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outB, loadU1 | 2. clearMIconter |
|-----------------|------------------|

### **3.6.13 COPY\_B\_to\_U2 : 0x46**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outB, loadU2 | 2. clearMIconter |
|-----------------|------------------|

### **3.6.14 COPY\_B\_to\_U3 : 0x47**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outB, loadU3 | 2. clearMIconter |
|-----------------|------------------|

### **3.6.15 COPY\_B\_to\_cmp : 0x48**

*Micro-instructions :*

- |                  |                  |
|------------------|------------------|
| 1. outB, loadCmp | 2. clearMIconter |
|------------------|------------------|

### **3.6.16 COPY\_U0\_to\_A : 0x49**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outU0, loadA | 2. clearMIconter |
|-----------------|------------------|

### **3.6.17 COPY\_U0\_to\_B : 0x4a**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outU0, loadB | 2. clearMIconter |
|-----------------|------------------|

### **3.6.18 COPY\_U0\_to\_cmp : 0x4b**

*Micro-instructions :*

1. outU0, loadCmp
2. clearMIconter

### **3.6.19 COPY\_U1\_to\_A : 0x4c**

*Micro-instructions :*

1. outU1, loadA
2. clearMIconter

### **3.6.20 COPY\_U1\_to\_B : 0x4d**

*Micro-instructions :*

1. outU1, loadB
2. clearMIconter

### **3.6.21 COPY\_U1\_to\_cmp : 0x4e**

*Micro-instructions :*

1. outU1, loadCmp
2. clearMIconter

### **3.6.22 COPY\_U2\_to\_A : 0x4f**

*Micro-instructions :*

1. outU2, loadA
2. clearMIconter

### **3.6.23 COPY\_U2\_to\_B : 0x50**

*Micro-instructions :*

1. outU2, loadB
2. clearMIconter

### **3.6.24 COPY\_U2\_to\_cmp : 0x51**

*Micro-instructions :*

1. outU2, loadCmp
2. clearMIconter

### **3.6.25 COPY\_U3\_to\_A : 0x52**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outU3, loadA | 2. clearMIconter |
|-----------------|------------------|

### **3.6.26 COPY\_U3\_to\_B : 0x53**

*Micro-instructions :*

- |                 |                  |
|-----------------|------------------|
| 1. outU3, loadB | 2. clearMIconter |
|-----------------|------------------|

### **3.6.27 COPY\_U3\_to\_cmp : 0x54**

*Micro-instructions :*

- |                   |                  |
|-------------------|------------------|
| 1. outU3, loadCmp | 2. clearMIconter |
|-------------------|------------------|

### **3.6.28 COPY\_mem\_to\_cmp : 0x55**

*Micro-instructions :*

- |                               |                    |
|-------------------------------|--------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadCmp |
| 2. outRAM, loadMemAddr, incPC |                    |
| 3. outMemAddr, loadRAM        | 5. clearMIconter   |

## **3.7 DISPLAY**

### **3.7.1 DISPLAY\_A : 0x56**

*Micro-instructions :*

- |                      |                  |
|----------------------|------------------|
| 1. outA, loadDisplay | 2. clearMIconter |
|----------------------|------------------|

### **3.7.2 DISPLAY\_B : 0x57**

*Micro-instructions :*

- |                      |                  |
|----------------------|------------------|
| 1. outB, loadDisplay | 2. clearMIconter |
|----------------------|------------------|

### **3.7.3 DISPLAY\_U0 : 0x58**

*Micro-instructions :*

- |                       |                  |
|-----------------------|------------------|
| 1. outU0, loadDisplay | 2. clearMIconter |
|-----------------------|------------------|

### **3.7.4 DISPLAY\_\_U1 : 0x59**

*Micro-instructions :*

- |                       |                  |
|-----------------------|------------------|
| 1. outU1, loadDisplay | 2. clearMIconter |
|-----------------------|------------------|

### **3.7.5 DISPLAY\_\_U2 : 0x5a**

*Micro-instructions :*

- |                       |                  |
|-----------------------|------------------|
| 1. outU2, loadDisplay | 2. clearMIconter |
|-----------------------|------------------|

### **3.7.6 DISPLAY\_\_U3 : 0x5b**

*Micro-instructions :*

- |                       |                  |
|-----------------------|------------------|
| 1. outU3, loadDisplay | 2. clearMIconter |
|-----------------------|------------------|

### **3.7.7 DISPLAY\_\_mem : 0x5c**

*Micro-instructions :*

- |                               |                        |
|-------------------------------|------------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadDisplay |
| 2. outRAM, loadMemAddr, incPC |                        |
| 3. loadRAM, outMemAddr        | 5. clearMIconter       |

## **3.8 FAIL**

### **3.8.1 FAIL : 0x5d**

*Micro-instructions :*

1. error, halt

## **3.9 HALT**

### **3.9.1 HALT : 0x5e**

*Micro-instructions :*

1. halt

### 3.10 INC

#### 3.10.1 INC\_\_A : 0x5f

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outA, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadA            |                  |

#### 3.10.2 INC\_\_B : 0x60

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outB, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadB            |                  |

#### 3.10.3 INC\_\_U0 : 0x61

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outU0, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadU0            |                  |

#### 3.10.4 INC\_\_U1 : 0x62

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outU1, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadU1            |                  |

#### 3.10.5 INC\_\_U2 : 0x63

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outU2, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadU2            |                  |

#### 3.10.6 INC\_\_U3 : 0x64

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outU3, enableInc, loadALU | 3. clearMIconter |
| 2. outALU, loadU3            |                  |

### 3.10.7 INC\_\_mem : 0x65

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 4. outRAM, enableInc, loadALU |
| 2. outRAM, loadMemAddr, incPC | 5. outALU, storeRAM           |
| 3. outMemAddr, loadRAM        | 6. clearMIconter              |

## 3.11 JMP

### 3.11.1 JMP\_\_const : 0x66

*Micro-instructions :*

- |                                |                  |
|--------------------------------|------------------|
| 1. outPC, loadRAM              | 3. clearMIconter |
| 2. outRAM, loadPC, cond_always |                  |

### 3.11.2 JMP\_\_if\_\_eq : 0x67

*Micro-instructions :*

- |                   |                              |
|-------------------|------------------------------|
| 1. outPC, loadRAM | 3. outRAM, loadPC, cond_null |
| 2. incPC          | 4. clearMIconter             |

### 3.11.3 JMP\_\_if\_\_ge : 0x68

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outPC, loadRAM            | cond_null        |
| 2. incPC                     |                  |
| 3. outRAM, loadPC, cond_pos, | 4. clearMIconter |

### 3.11.4 JMP\_\_if\_\_gt : 0x69

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outPC, loadRAM            | cond_not_null    |
| 2. incPC                     |                  |
| 3. outRAM, loadPC, cond_pos, | 4. clearMIconter |

### 3.11.5 JMP\_if\_le : 0x6a

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outPC, loadRAM            | cond_null        |
| 2. incPC                     |                  |
| 3. outRAM, loadPC, cond_neg, | 4. clearMIconter |

### 3.11.6 JMP\_if\_lt : 0x6b

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. outPC, loadRAM            | cond_not_null    |
| 2. incPC                     |                  |
| 3. outRAM, loadPC, cond_neg, | 4. clearMIconter |

### 3.11.7 JMP\_if\_neq : 0x6c

*Micro-instructions :*

- |                               |                  |
|-------------------------------|------------------|
| 1. outPC, loadRAM             | invert_cond      |
| 2. incPC                      |                  |
| 3. outRAM, loadPC, cond_null, | 4. clearMIconter |

### 3.11.8 JMP\_ptr : 0x6d

*Micro-instructions :*

- |                        |                                |
|------------------------|--------------------------------|
| 1. outPC, loadRAM      | 4. outRAM, loadPC, cond_always |
| 2. outRAM, loadMemAddr |                                |
| 3. outMemAddr, loadRAM | 5. clearMIconter               |

### 3.11.9 JMP\_sel\_bit\_0 : 0x6e

*Micro-instructions :*

- |                   |                                      |
|-------------------|--------------------------------------|
| 1. outPC, loadRAM | 3. outRAM, loadPC, cond_selected_bit |
| 2. incPC          | 4. clearMIconter                     |

#### 3.11.10 JMP\_sel\_bit\_1 : 0x6f

*Micro-instructions :*

1. outPC, loadRAM selector0
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

#### 3.11.11 JMP\_sel\_bit\_2 : 0x70

*Micro-instructions :*

1. outPC, loadRAM selector1
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

#### 3.11.12 JMP\_sel\_bit\_3 : 0x71

*Micro-instructions :*

1. outPC, loadRAM selector0, selector1
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

#### 3.11.13 JMP\_sel\_bit\_4 : 0x72

*Micro-instructions :*

1. outPC, loadRAM selector2
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

#### 3.11.14 JMP\_sel\_bit\_5 : 0x73

*Micro-instructions :*

1. outPC, loadRAM selector0, selector2
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter



### 3.11.15 JMP\_sel\_bit\_6 : 0x74

*Micro-instructions :*

1. outPC, loadRAM selector1, selector2
2. incPC
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

### 3.11.16 JMP\_sel\_bit\_7 : 0x75

*Micro-instructions :*

1. outPC, loadRAM selector0, selector1,
2. incPC selector2
3. outRAM, loadPC, cond\_selected\_bit, 4. clearMIconter

## 3.12 LED

### 3.12.1 LED\_tgl : 0x76

*Micro-instructions :*

1. flipLed
2. clearMIconter

## 3.13 LOAD

### 3.13.1 LOAD\_const\_to\_A : 0x77

*Micro-instructions :*

1. outPC, loadRAM
2. outRAM, loadA, incPC
3. clearMIconter

### 3.13.2 LOAD\_const\_to\_B : 0x78

*Micro-instructions :*

1. outPC, loadRAM
2. outRAM, loadB, incPC
3. clearMIconter

### 3.13.3 LOAD\_\_const\_\_to\_\_U0 : 0x79

*Micro-instructions :*

- |                          |                  |
|--------------------------|------------------|
| 1. outPC, loadRAM        | 3. clearMIconter |
| 2. outRAM, loadU0, incPC |                  |

### 3.13.4 LOAD\_\_const\_\_to\_\_U1 : 0x7a

*Micro-instructions :*

- |                          |                  |
|--------------------------|------------------|
| 1. outPC, loadRAM        | 3. clearMIconter |
| 2. outRAM, loadU1, incPC |                  |

### 3.13.5 LOAD\_\_const\_\_to\_\_U2 : 0x7b

*Micro-instructions :*

- |                          |                  |
|--------------------------|------------------|
| 1. outPC, loadRAM        | 3. clearMIconter |
| 2. outRAM, loadU2, incPC |                  |

### 3.13.6 LOAD\_\_const\_\_to\_\_U3 : 0x7c

*Micro-instructions :*

- |                          |                  |
|--------------------------|------------------|
| 1. outPC, loadRAM        | 3. clearMIconter |
| 2. outRAM, loadU3, incPC |                  |

### 3.13.7 LOAD\_\_ptr\_\_to\_\_A : 0x7d

*Micro-instructions :*

- |                               |                  |
|-------------------------------|------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadA |
| 2. outRAM, loadMemAddr, incPC |                  |
| 3. outMemAddr, loadRAM        | 5. clearMIconter |

### 3.13.8 LOAD\_\_ptr\_\_to\_\_B : 0x7e

*Micro-instructions :*

- |                               |                  |
|-------------------------------|------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadB |
| 2. outRAM, loadMemAddr, incPC |                  |
| 3. outMemAddr, loadRAM        | 5. clearMIconter |

### 3.13.9 LOAD\_ptr\_to\_U0 : 0x7f

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadU0 |
| 2. outRAM, loadMemAddr, incPC |                   |
| 3. outMemAddr, loadRAM        | 5. clearMIconter  |

### 3.13.10 LOAD\_ptr\_to\_U1 : 0x80

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadU1 |
| 2. outRAM, loadMemAddr, incPC |                   |
| 3. outMemAddr, loadRAM        | 5. clearMIconter  |

### 3.13.11 LOAD\_ptr\_to\_U2 : 0x81

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadU2 |
| 2. outRAM, loadMemAddr, incPC |                   |
| 3. outMemAddr, loadRAM        | 5. clearMIconter  |

### 3.13.12 LOAD\_ptr\_to\_U3 : 0x82

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadU3 |
| 2. outRAM, loadMemAddr, incPC |                   |
| 3. outMemAddr, loadRAM        | 5. clearMIconter  |

## 3.14 NEG

### 3.14.1 NEG\_A : 0x83

*Micro-instructions :*

- |                               |                  |
|-------------------------------|------------------|
| 1. outA, enableNOT, loadALU   | 3. outALU, loadA |
| 2. outALU, enableInc, loadALU | 4. clearMIconter |

### 3.14.2 NEG\_B : 0x84

*Micro-instructions :*

- |                               |                  |
|-------------------------------|------------------|
| 1. outB, enableNOT, loadALU   | 3. outALU, loadB |
| 2. outALU, enableInc, loadALU | 4. clearMIconter |

### 3.14.3 NEG\_U0 : 0x85

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outU0, enableNOT, loadALU  | 3. outALU, loadU0 |
| 2. outALU, enableInc, loadALU | 4. clearMIconter  |

### 3.14.4 NEG\_U1 : 0x86

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outU1, enableNOT, loadALU  | 3. outALU, loadU1 |
| 2. outALU, enableInc, loadALU | 4. clearMIconter  |

### 3.14.5 NEG\_U2 : 0x87

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outU2, enableNOT, loadALU  | 3. outALU, loadU2 |
| 2. outALU, enableInc, loadALU | 4. clearMIconter  |

### 3.14.6 NEG\_U3 : 0x88

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outU3, enableNOT, loadALU  | 3. outALU, loadU3 |
| 2. outALU, enableInc, loadALU | 4. clearMIconter  |

### 3.14.7 NEG\_mem : 0x89

*Micro-instructions :*

- |                               |                               |
|-------------------------------|-------------------------------|
| 1. outPC, loadRAM             | 5. outALU, enableInc, loadALU |
| 2. outRAM, loadMemAddr        | 6. outALU, storeRAM           |
| 3. outMemAddr, loadRAM, incPC | 7. clearMIconter              |
| 4. outRAM, enableNOT, loadALU |                               |

### 3.15 NOP

#### 3.15.1 NOP : 0x8a

*Micro-instructions :*

1. clearMIconter

### 3.16 NOT

#### 3.16.1 NOT\_\_A : 0x8b

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. enableNOT, loadALU, outA | 3. clearMIconter |
| 2. outALU, loadA            |                  |

#### 3.16.2 NOT\_\_B : 0x8c

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. enableNOT, loadALU, outB | 3. clearMIconter |
| 2. outALU, loadB            |                  |

#### 3.16.3 NOT\_\_U0 : 0x8d

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableNOT, loadALU, outU0 | 3. clearMIconter |
| 2. outALU, loadU0            |                  |

#### 3.16.4 NOT\_\_U1 : 0x8e

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableNOT, loadALU, outU1 | 3. clearMIconter |
| 2. outALU, loadU1            |                  |

#### 3.16.5 NOT\_\_U2 : 0x8f

*Micro-instructions :*

- |                              |                  |
|------------------------------|------------------|
| 1. enableNOT, loadALU, outU2 | 3. clearMIconter |
| 2. outALU, loadU2            |                  |

### 3.16.6 NOT\_U3 : 0x90

*Micro-instructions :*

1. enableNOT, loadALU, outU3
2. outALU, loadU3
3. clearMICounter

## 3.17 OR

### 3.17.1 OR\_A\_B\_to\_itself : 0x91

*Micro-instructions :*

1. enableOR, loadALU, outA
2. outALU, loadA
3. clearMICounter

### 3.17.2 OR\_U0\_B\_to\_itself : 0x92

*Micro-instructions :*

1. enableOR, loadALU, outU0
2. outALU, loadU0
3. clearMICounter

### 3.17.3 OR\_U1\_B\_to\_itself : 0x93

*Micro-instructions :*

1. enableOR, loadALU, outU1
2. outALU, loadU1
3. clearMICounter

### 3.17.4 OR\_U2\_B\_to\_itself : 0x94

*Micro-instructions :*

1. enableOR, loadALU, outU2
2. outALU, loadU2
3. clearMICounter

### 3.17.5 OR\_U3\_B\_to\_itself : 0x95

*Micro-instructions :*

1. enableOR, loadALU, outU3
2. outALU, loadU3
3. clearMICounter

### 3.18 RET

#### 3.18.1 RET : 0x96

*Micro-instructions :*

1. outRet, loadPC, cond\_always
2. clearMIconter

### 3.19 SHIFTL

#### 3.19.1 SHIFTL\_\_A : 0x97

*Micro-instructions :*

1. outA, enableSHIFTL, loadALU
2. outALU, loadA
3. clearMIconter

#### 3.19.2 SHIFTL\_\_B : 0x98

*Micro-instructions :*

1. outB, enableSHIFTL, loadALU
2. outALU, loadB
3. clearMIconter

#### 3.19.3 SHIFTL\_\_U0 : 0x99

*Micro-instructions :*

1. outU0, enableSHIFTL, loadALU
2. outALU, loadU0
3. clearMIconter

#### 3.19.4 SHIFTL\_\_U1 : 0x9a

*Micro-instructions :*

1. outU1, enableSHIFTL, loadALU
2. outALU, loadU1
3. clearMIconter

#### 3.19.5 SHIFTL\_\_U2 : 0x9b

*Micro-instructions :*

1. outU2, enableSHIFTL, loadALU
2. outALU, loadU2
3. clearMIconter

### 3.19.6 SHIFTL\_\_U3 : 0x9c

*Micro-instructions :*

1. outU3, enableSHIFTL, loadALU
2. outALU, loadU3
3. clearMIconter

## 3.20 SHIFTR

### 3.20.1 SHIFTR\_\_A : 0x9d

*Micro-instructions :*

1. outA, enableSHIFTR, loadALU
2. outALU, loadA
3. clearMIconter

### 3.20.2 SHIFTR\_\_B : 0x9e

*Micro-instructions :*

1. outB, enableSHIFTR, loadALU
2. outALU, loadB
3. clearMIconter

### 3.20.3 SHIFTR\_\_U0 : 0x9f

*Micro-instructions :*

1. outU0, enableSHIFTR, loadALU
2. outALU, loadU0
3. clearMIconter

### 3.20.4 SHIFTR\_\_U1 : 0xa0

*Micro-instructions :*

1. outU1, enableSHIFTR, loadALU
2. outALU, loadU1
3. clearMIconter

### 3.20.5 SHIFTR\_\_U2 : 0xa1

*Micro-instructions :*

1. outU2, enableSHIFTR, loadALU
2. outALU, loadU2
3. clearMIconter



### 3.20.6 SHIFTR\_\_U3 : 0xa2

*Micro-instructions :*

- |                                 |                  |
|---------------------------------|------------------|
| 1. outU3, enableSHIFTR, loadALU | 3. clearMIconter |
| 2. outALU, loadU3               |                  |

## 3.21 SLEEP

### 3.21.1 SLEEP\_\_A : 0xa3

*Micro-instructions :*

- |                           |                  |
|---------------------------|------------------|
| 1. outA, loadSleep, incPC | 2. clearMIconter |
|---------------------------|------------------|

### 3.21.2 SLEEP\_\_B : 0xa4

*Micro-instructions :*

- |                           |                  |
|---------------------------|------------------|
| 1. outB, loadSleep, incPC | 2. clearMIconter |
|---------------------------|------------------|

### 3.21.3 SLEEP\_\_U0 : 0xa5

*Micro-instructions :*

- |                            |                  |
|----------------------------|------------------|
| 1. outU0, loadSleep, incPC | 2. clearMIconter |
|----------------------------|------------------|

### 3.21.4 SLEEP\_\_U1 : 0xa6

*Micro-instructions :*

- |                            |                  |
|----------------------------|------------------|
| 1. outU1, loadSleep, incPC | 2. clearMIconter |
|----------------------------|------------------|

### 3.21.5 SLEEP\_\_U2 : 0xa7

*Micro-instructions :*

- |                            |                  |
|----------------------------|------------------|
| 1. outU2, loadSleep, incPC | 2. clearMIconter |
|----------------------------|------------------|

### 3.21.6 SLEEP\_\_U3 : 0xa8

*Micro-instructions :*

- |                            |                  |
|----------------------------|------------------|
| 1. outU3, loadSleep, incPC | 2. clearMIconter |
|----------------------------|------------------|

### 3.21.7 SLEEP\_const : 0xa9

*Micro-instructions :*

- |                             |                  |
|-----------------------------|------------------|
| 1. outPC, loadRAM           | 3. clearMIconter |
| 2. outRAM, loadSleep, incPC |                  |

### 3.21.8 SLEEP\_mem : 0xaa

*Micro-instructions :*

- |                               |                      |
|-------------------------------|----------------------|
| 1. outPC, loadRAM             | 4. outRAM, loadSleep |
| 2. outRAM, loadMemAddr, incPC |                      |
| 3. loadRAM, outMemAddr        | 5. clearMIconter     |

## 3.22 STORE

### 3.22.1 STORE\_A\_to\_address : 0xab

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 3. storeRAM, outA |
| 2. outRAM, loadMemAddr, incPC | 4. clearMIconter  |

### 3.22.2 STORE\_B\_to\_address : 0xac

*Micro-instructions :*

- |                               |                   |
|-------------------------------|-------------------|
| 1. outPC, loadRAM             | 3. storeRAM, outB |
| 2. outRAM, loadMemAddr, incPC | 4. clearMIconter  |
-