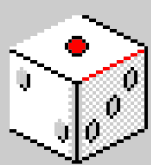


Intel 4004 Calculator



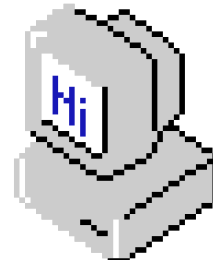
Used the same emulator
in the assignment

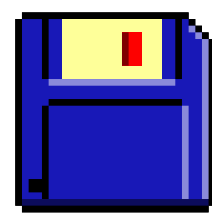


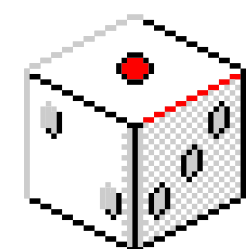
11:11PM

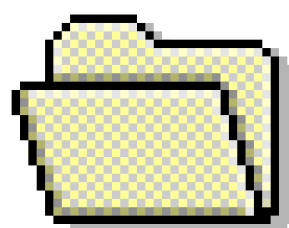
Agenda

Topics Covered


Intel 4004


hardware constraints


Algorithm


Future Plans

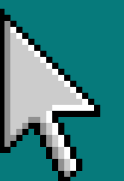
Start



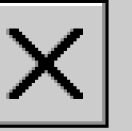
Intel 4004



- The Intel 4004 is a 4-bit central processing unit (CPU) released by Intel Corporation in 1971.
- It was the first commercially produced microprocessor.
- Designed primarily for use in small business systems like calculators, automated teller machines and cash machines
- Clock speed = 740 kHz
- 92,600 instructions per second
- 12-bit addresses and a 4-bit address bus
- separate memory for both data and program



Hardware constraints



It is a 4-bit processor, which means it is primarily intended for use with 4-bit binary values. Because the chip only has 16 pins, there isn't much room to begin working with.

The CPU is relatively primitive, with just a few ALU operations in its instruction set(45).

- operation on 4-bit operands only
- complement
- rotate left or right

Program can not be long because program counter is 12-bit wide.

Jump conditions - both conditional or indirect jumps are short and can address only the 8-bit range

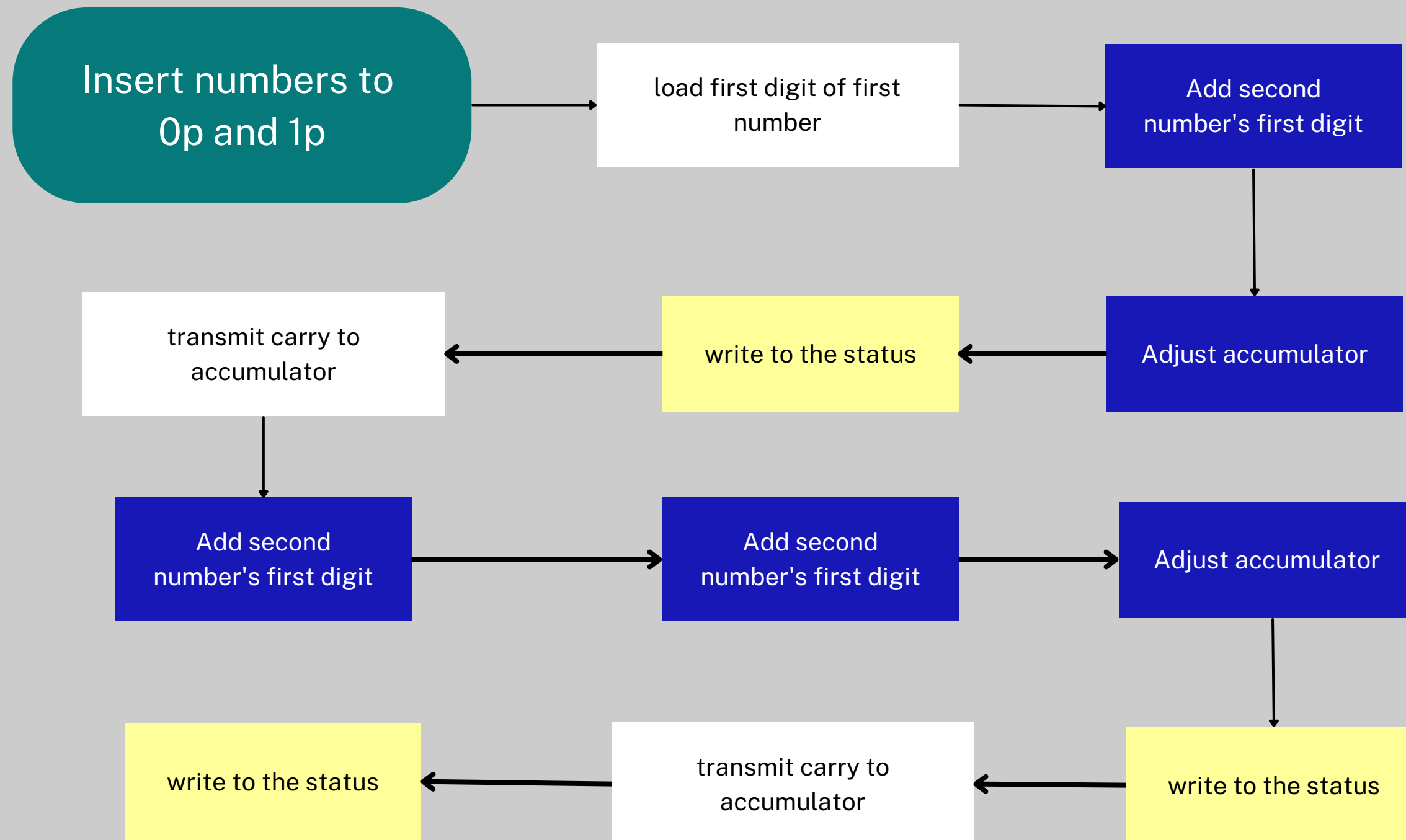
Algorithm

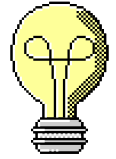
Common features

- Insert numbers into the first and second register pairs.
- Store result in chip 0, bank 0, Register 0, 4-bit Status nibbles.

Multiplication is done by addition.

Division is done by subtraction.





Addition

After every addition, we need to normalize the number in the accumulator to be sure that it's in the valid form before writing to the output or doing more calculations

INTEL 4004 SYSTEM EMULATOR (initial version)

4002 RAM: BANK #0 CHIP #0

MAIN MEMORY

0123456789ABCDEF

REGISTER 00000000000000000000

REGISTER 10000000000000000000

REGISTER 20000000000000000000

REGISTER 30000000000000000000

STATUS

0123

0021

0000

0000

0000

OUTPUTS:

Q0Q1Q2Q3

RESET

4004 CPU:

STACK

PC013

LEVEL 1000

LEVEL 2000

LEVEL 3000

REGISTERS

R0R107R8R900

R2R314RARB00

R4R500RCRD00

R6R700RERF00

ACCUMULATOR: 0 [0000]

CARRY:

TEST:

013 40 13 JUN \$013

CYCLES: 27

STEP

ANIMATE

RUN

STOP

RESET

4002 RAM: BANK #0 CHIP #1

MAIN MEMORY

0123456789ABCDEF

REGISTER 00000000000000000000

REGISTER 10000000000000000000

REGISTER 20000000000000000000

REGISTER 30000000000000000000

STATUS

0123

0000

0000

0000

0000

OUTPUTS:

Q0Q1Q2Q3

RESET

CONFIGURATION:

General

Intel 4001-0009 hardware (link)

MCS-4/40 hardware (link)

BreakPoints

RESET All

ROM block:

000 F0 CLB

001 FD DCL

002 20 07 FIM P0,\$07

004 22 14 FIM P1,\$14

006 28 00 FIM P4,\$00

008 A1 LD R1

009 83 ADD R3

00A FB DAA

00B E7 WR3

00C F7 TCC

00D 80 ADD R0

00E 82 ADD R2

00F FB DAA

010 E6 WR2

011 F7 TCC

012 FF WR3

LOAD

RESET

P1

P2

P3

Assembling code

ASCII

4001 ROM:

I/O PORT:

Q0Q1Q2Q3

INTEL 4004 SYSTEM EMULATOR (initial version)

4002 RAM: BANK #0 CHIP #0

MAIN MEMORY

0123456789ABCDEF

STATUS0123

REGISTER 000000000000000000000198

REGISTER 100000000000000000000000

REGISTER 200000000000000000000000

REGISTER 300000000000000000000000

OUTPUTS:Q0Q1Q2Q3

RESET

4004 CPU:

STACK

PC013

LEVEL 1000

LEVEL 2000

LEVEL 3000

REGISTERS

R0R199R8R900

R2R399RARB00

R4R500RCRD00

R6R700RERF00

ACCUMULATOR: 1 [0001]

CARRY:

TEST:

013 40 13 JUN \$013

CYCLES: 51

STEP

ANIMATE

RUN

STOP

RESET

4002 RAM: BANK #0 CHIP #1

MAIN MEMORY

0123456789ABCDEF

STATUS0123

REGISTER 000000000000000000000000

REGISTER 100000000000000000000000

REGISTER 200000000000000000000000

REGISTER 300000000000000000000000

OUTPUTS:Q0Q1Q2Q3

RESET

CONFIGURATION:

General

Intel 4001-0009 hardware (link)

MCS-4/40 hardware (link)

BreakPoints

RESET All

4001 ROM:

I/O PORT:

Q0Q1Q2Q3

ROM block:

000 F0 CLB

001 FD DCL

002 20 99 FIM P0,\$99

004 22 99 FIM P1,\$99

006 28 00 FIM P4,\$00

008 A1 LD R1

009 83 ADD R3

00A FB DAA

00B E7 WR3

00C F7 TCC

00D 80 ADD R0

00E 82 ADD R2

00F FB DAA

010 E6 WR2

011 F7 TCC

012 FF WR3

LOAD

RESET

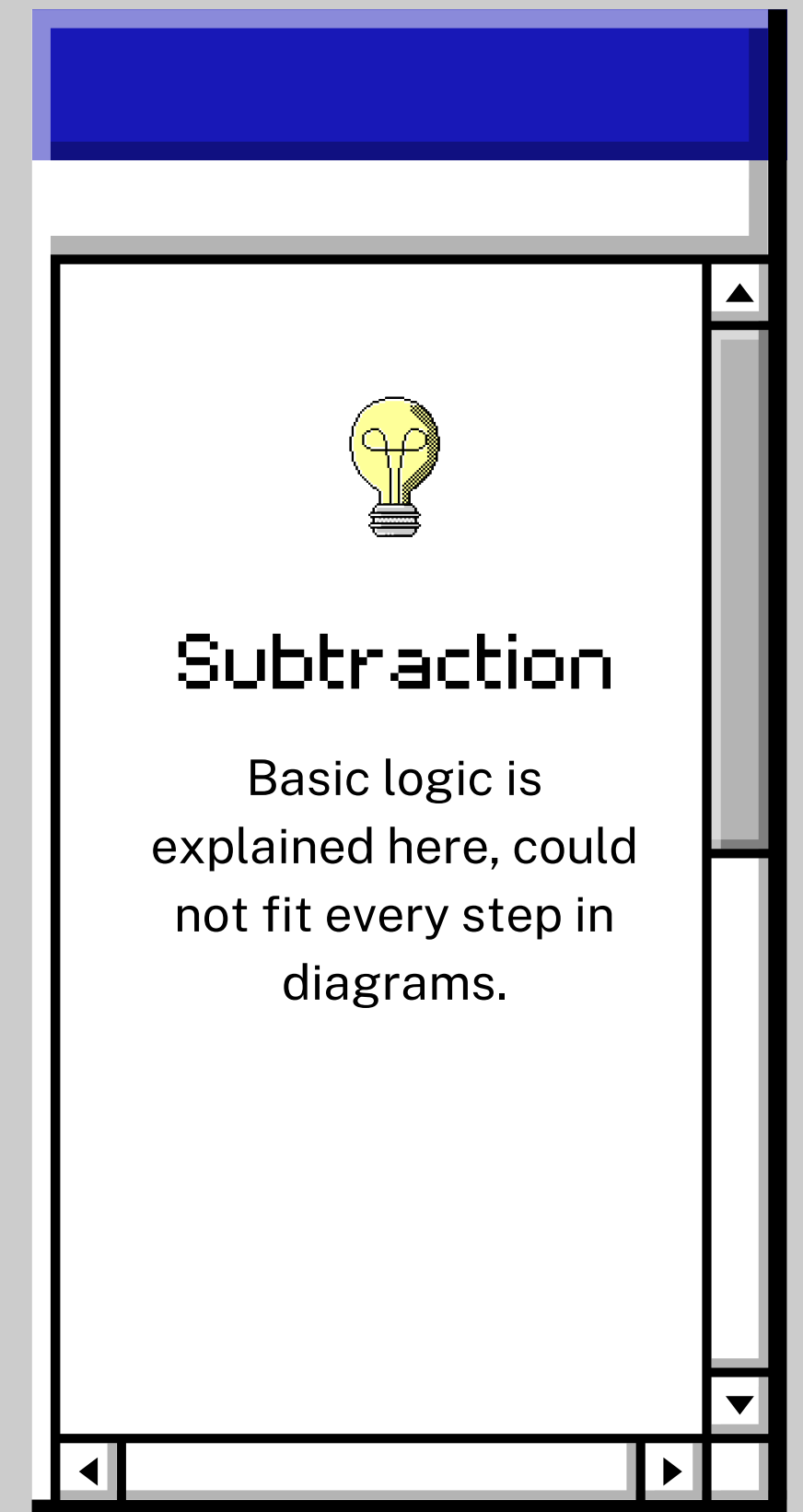
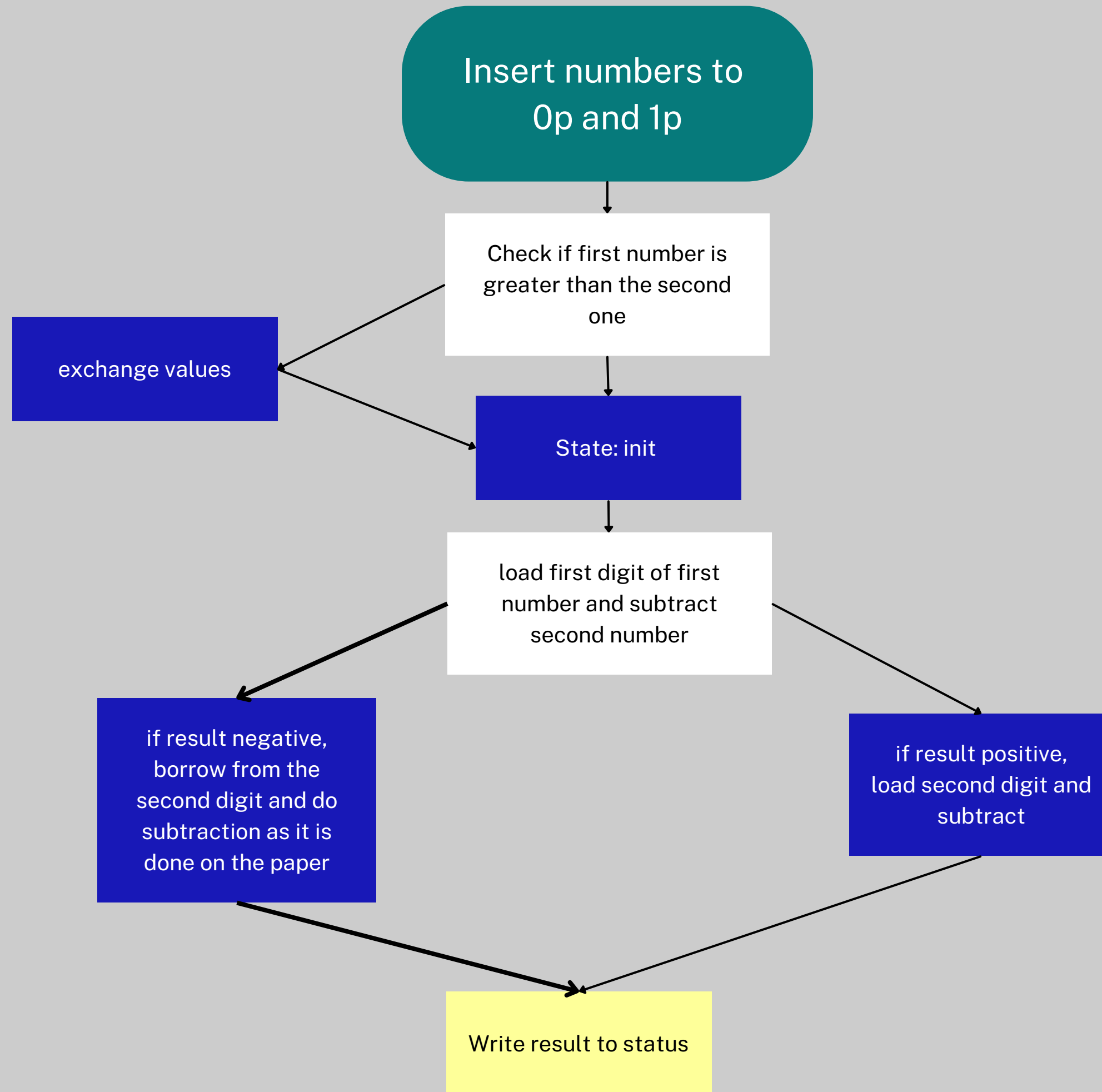
P1

P2

P3

Assembling code

ASCII



Conditional Jump in Assembly
jcn \$2 loop ; \$0010
inverted, A=1, C is set, TEST is set

INTEL 4004 SYSTEM EMULATOR (initial version)

4002 RAM: BANK #0 CHIP #0

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	5
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

4002 RAM: BANK #0 ▼ CHIP #1 ▼

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

4001 ROM:

I/O PORT:

Q0 Q1 Q2 Q3

4004 CPU:

	STACK	REGISTERS	
PC	0 2 F	R0 R1 4 8	R8 R9 0 5
LEVEL 1	0 0 0	R2 R3 1 3	RARB 0 0
LEVEL 2	0 0 0	R4 R5 0 9	RCRD 0 0
LEVEL 3	0 0 0	R6 R7 0 6	RERF 0 0

ACCUMULATOR: 3 [0011] CARRY: ☒ TEST: ☐

02F 40 2F JUN \$02F

CYCLES: 1837507

STEP

ANIMATE

RUN

STOP

RESET

CONFIGURATION:

- General

- Intel 4001-0009 hardware ([link](#))

- MCS-4/40 hardware ([link](#))

BreakPoints

RESET All

ROM block:

```
000 20 48 FIM P0,$48
002 22 13 FIM P1,$13
004 24 09 FIM P2,$09
006 26 06 FIM P3,$06
008 A0 LD R0
009 92 SUB R2
00A 12 12 JCN CZ,$12
00C A1 LD R1
00D B3 XCH R3
00E B1 XCH R1
00F A0 LD R0
010 B2 XCH R2
011 B0 XCH R0
012 F1 CLC
013 A1 LD R1
```

LOAD

RESET

P1

P2

P3

- ✓ **Assembling code**

- **ASCII**

INTEL 4004 SYSTEM EMULATOR (initial version)

4002 RAM: BANK #0 ▼ CHIP #0 ▼

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	2
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

4002 RAM: BANK #0 ▼ CHIP #1 ▼

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

4001 ROM:

I/O PORT:

Q0 Q1 Q2 Q3

p0: 7

p1: 89

Algorithm checked and exchanged
values of R0R1 and R2R3

4004 CPU:

	STACK	REGISTERS	
PC	0 2 F	R0 R1 8 9	R8 R9 0 2
LEVEL 1	0 0 0	R2 R3 0 7	RARB 0 0
LEVEL 2	0 0 0	R4 R5 0 9	RCRD 0 0
LEVEL 3	0 0 0	R6 R7 0 6	RERF 0 0

ACCUMULATOR: 8 [1000] CARRY: ☒ TEST: ☐

02F 40 2F JUN \$02F

CYCLES: 2650001

STEP

ANIMATE

RUN

STOP

RESET

CONFIGURATION:

- General

- Intel 4001-0009 hardware ([link](#))

- MCS-4/40 hardware ([link](#))

BreakPoints

RESET All

ROM block:

```
000 20 07 FIM P0,$07
002 22 89 FIM P1,$89
004 24 09 FIM P2,$09
006 26 06 FIM P3,$06
008 A0 LD R0
009 92 SUB R2
00A 12 12 JCN CZ,$12
00C A1 LD R1
00D B3 XCH R3
00E B1 XCH R1
00F A0 LD R0
010 B2 XCH R2
011 B0 XCH R0
012 F1 CLC
013 A1 LD R1
```

LOAD

RESET

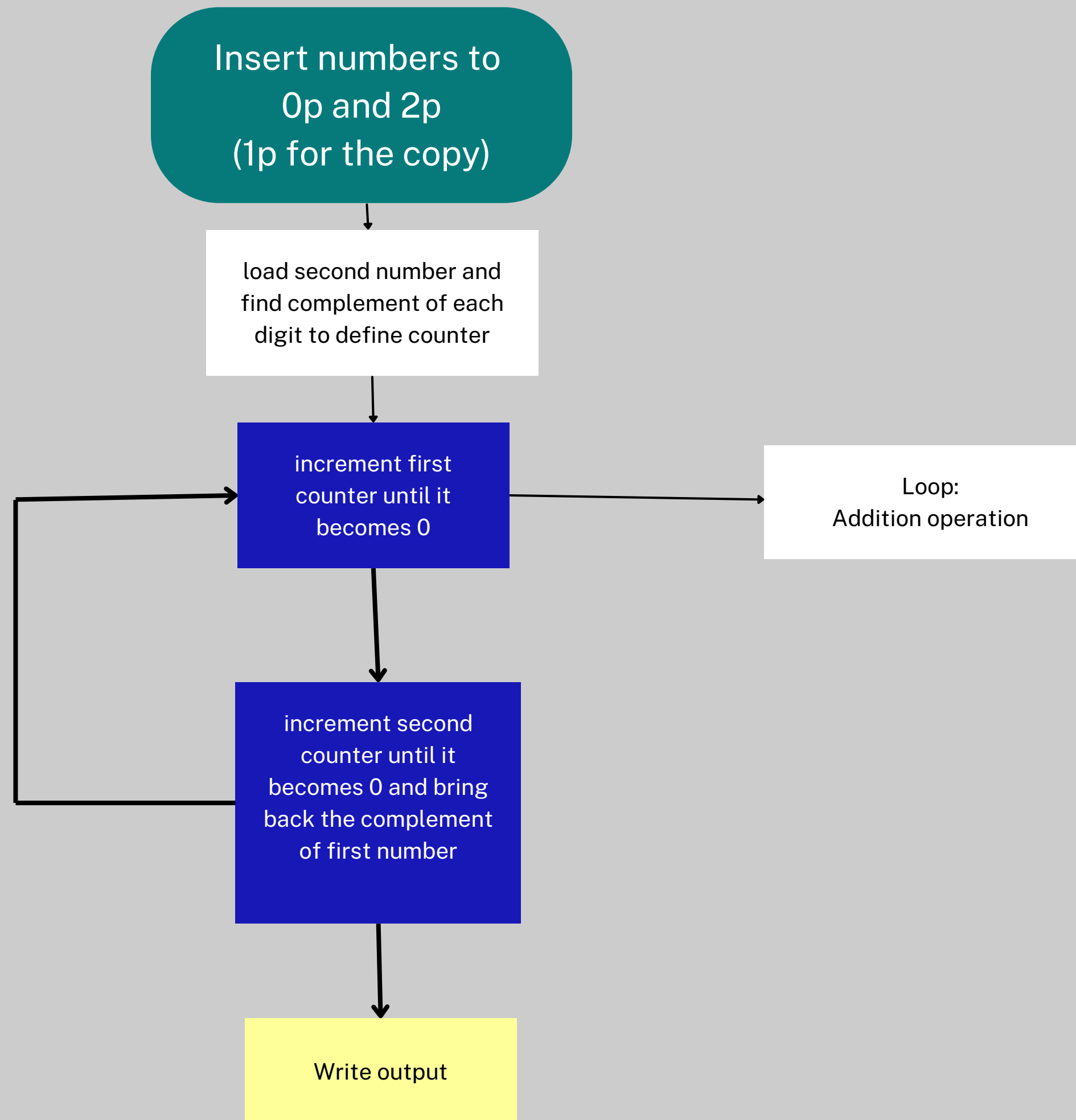
P1

P2

P3

- ✓ **Assembling code**

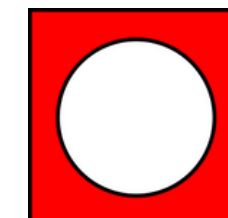
- **ASCII**



Multiplication

We re-use the addition algorithm here with implementing loops.

Complement:



INTEL 4004 SYSTEM EMULATOR (initial version)

19 x 2

4002 RAM: BANK #0 ▼ CHIP #0 ▼

4004 CPU:

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

	STACK	REGISTERS	
PC	0 2 C	R0 R1 3 8	R8 R9 0 0
LEVEL 1	0 0 0	R2 R3 1 9	RARB 0 0
LEVEL 2	0 0 0	R4 R5 0 2	RC RD 0 0
LEVEL 3	0 0 0	R6 R7 0 D	RERF 0 0

ACCUMULATOR: 3 [0011] CARRY: ☐ TEST: ☐

02C 40 2C JUN \$02C

CYCLES: 1587500

STEP

ANIMATE

RUN

STOP

RESET

4002 RAM: BANK #0 ▼ CHIP #1 ▼

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS: **Q0** **Q1** **Q2** **Q3**

RESET

CONFIGURATION:

- General

- Intel 4001-0009 hardware ([link](#))

- MCS-4/40 hardware ([link](#))

BreakPoints

RESET All

ROM block:

```
000 20 19 FIM P0,$19
002 22 19 FIM P1,$19
004 24 02 FIM P2,$02
006 26 00 FIM P3,$00
008 A5     LD R5
009 F4     CMA
00A F2     IAC
00B B7     XCH R7
00C A4     LD R4
00D F4     CMA
00E B6     XCH R6
00F D0     LDM 0
010 40 1F JUN $01F
012 A1     LD R1
013 83     ADD R3
```

LOAD

RESET

P1

P2

P3

- ✓ **Assembling code**

- **ASCII**

4001 ROM:

I/O PORT:

Q0 Q1 Q2 Q3

99 x 2

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	8
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS:	Q0	Q1	Q2	Q3
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

		STACK		REGISTERS		STEP	
PC	0 2 C	R0 R1	9 8	R8 R9	0 0		
LEVEL 1	0 0 0	R2 R3	9 9	RARB	0 0	ANIMATE	
LEVEL 2	0 0 0	R4 R5	0 2	RC RD	0 0		
LEVEL 3	0 0 0	R6 R7	0 D	RE RF	0 0	RUN	
ACCUMULATOR: 9 [1001]		CARRY: <input type="checkbox"/>		TEST: <input type="checkbox"/>			
02C 40 2C JUN \$02C				CYCLES: 308		STOP	
						RESET	

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS:	Q0	Q1	Q2	Q3
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

General

Intel 4001-0009 hardware ([link](#))

MCS-4/40 hardware ([link](#))

BreakPoints

RESET All

I/O PORT:

Q0	Q1	Q2	Q3
●	●	●	●

000 20 99 FIM P0,\$99

002 22 99 FIM P1,\$99

004 24 02 FIM P2,\$02

006 26 00 FIM P3,\$00

008 A5 LD R5

009 F4 CMA

00A F2 IAC

00B B7 XCH R7

00C A4 LD R4

00D F4 CMA

00E B6 XCH R6

00F D0 LDM 0

010 40 1F JUN \$01F

012 A1 LD R1

013 83 ADD R3

LOAD

RESET

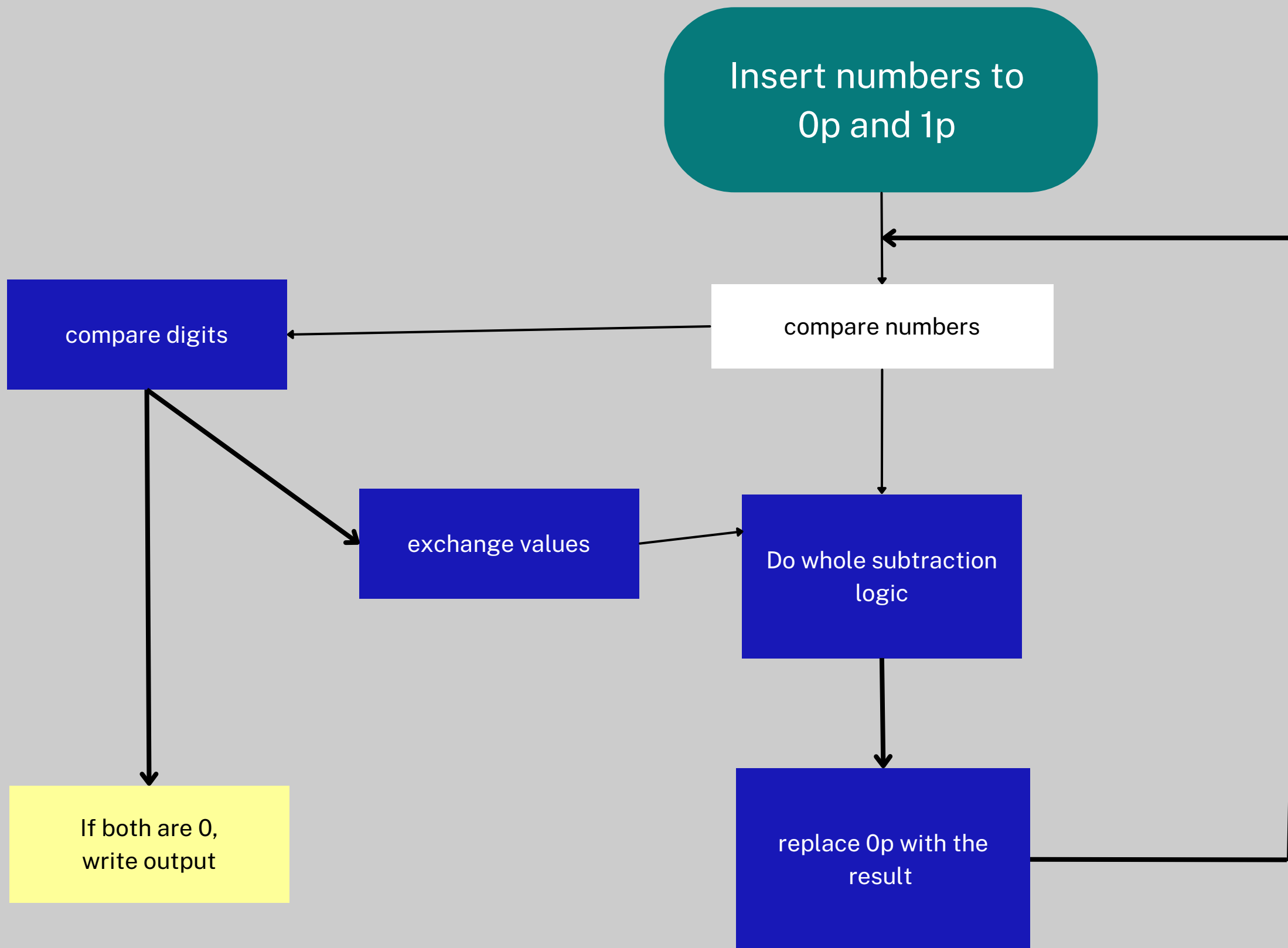
P1

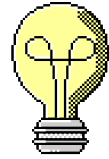
P2

P3

☒ Assembling code

☐ ASCII





Division

We re-use the subtraction algorithm here with implementing loops and conditional jumps.

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	C	3
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS:	Q0	Q1	Q2	Q3
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

		STACK		REGISTERS		STEP	
PC	0 3 B	R0 R1	0 3	R8 R9	0 3		
LEVEL 1	0 0 0	R2 R3	0 5	RARB	0 0	ANIMATE	
LEVEL 2	0 0 0	R4 R5	0 9	RC RD	0 0		
LEVEL 3	0 0 0	R6 R7	0 3	RERF	0 0	RUN	
ACCUMULATOR: 3 [0011]		CARRY: <input type="checkbox"/>		TEST: <input type="checkbox"/>		STOP	
03B 40 3B JUN \$03B		CYCLES: 1950001				RESET	

	MAIN MEMORY																STATUS			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
REGISTER 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REGISTER 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUTPUTS:	Q0	Q1	Q2	Q3
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

General

Intel 4001-0009 hardware ([link](#))

MCS-4/40 hardware ([link](#))

BreakPoints

RESET All

I/O PORT:

Q0	Q1	Q2	Q3
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The screenshot shows the 6502 Assembler interface. On the left, a list of assembly instructions is displayed in a monospaced font, with addresses, hex values, and mnemonics. The instructions are:

- 000 20 18 FIM P0,\$18
- 002 22 05 FIM P1,\$05
- 004 24 09 FIM P2,\$09
- 006 26 00 FIM P3,\$00
- 008 A0 LD R0
- 009 92 SUB R2
- 00A 12 18 JCN CZ,\$18
- 00C A1 LD R1
- 00D 93 SUB R3
- 00E 12 18 JCN CZ,\$18
- 010 A7 LD R7
- 011 E5 WR1
- 012 DC LDM 12
- 013 E6 WR2
- 014 A1 LD R1

On the right side, there are four buttons: "LOAD", "RESET", "P1", and "P3". At the bottom, there are two checkboxes: "Assembling code" (checked) and "ASCII" (unchecked).

27 / 13

4004 CPU:

STACK		REGISTERS	
PC	0 3 B	R0 R1	0 1 R8 R9 0 1
LEVEL 1	0 0 0	R2 R3	1 3 RARB 0 0
LEVEL 2	0 0 0	R4 R5	0 9 RCRD 0 0
LEVEL 3	0 0 0	R6 R7	0 2 RERF 0 0

ACCUMULATOR: 1 [0001] CARRY: ☐ TEST: ☐

03B 40 3B JUN \$03B CYCLES: 1725000

STEP

ANIMATE

RUN

STOP

RESET

CONFIGURATION:

☒ General

☐ Intel 4001-0009 hardware ([link](#))

☐ MCS-4/40 hardware ([link](#))

BreakPoints

RESET ALL

ROM block:

The screenshot shows the 6502 Assembler interface. On the left, a list of assembly instructions is displayed in a monospaced font, with addresses and hex values in red and instructions in green. The instructions are: 000 20 27 FIM P0,\$27, 002 22 13 FIM P1,\$13, 004 24 09 FIM P2,\$09, 006 26 00 FIM P3,\$00, 008 A0 LD R0, 009 92 SUB R2, 00A 12 18 JCN CZ,\$18, 00C A1 LD R1, 00D 93 SUB R3, 00E 12 18 JCN CZ,\$18, 010 A7 LD R7, 011 E5 WR1, 012 DC LDM 12, 013 E6 WR2, and 014 A1 LD R1. To the right of the code list are four buttons: 'LOAD', 'RESET', 'P1', and 'P2'. At the bottom, there are two checkboxes: 'Assembling code' (checked) and 'ASCII' (unchecked).

Future improvements



Add more operations for
calculator



C code which asks for
numbers and operators
to generate assembly
code for intel4004



Thank you!