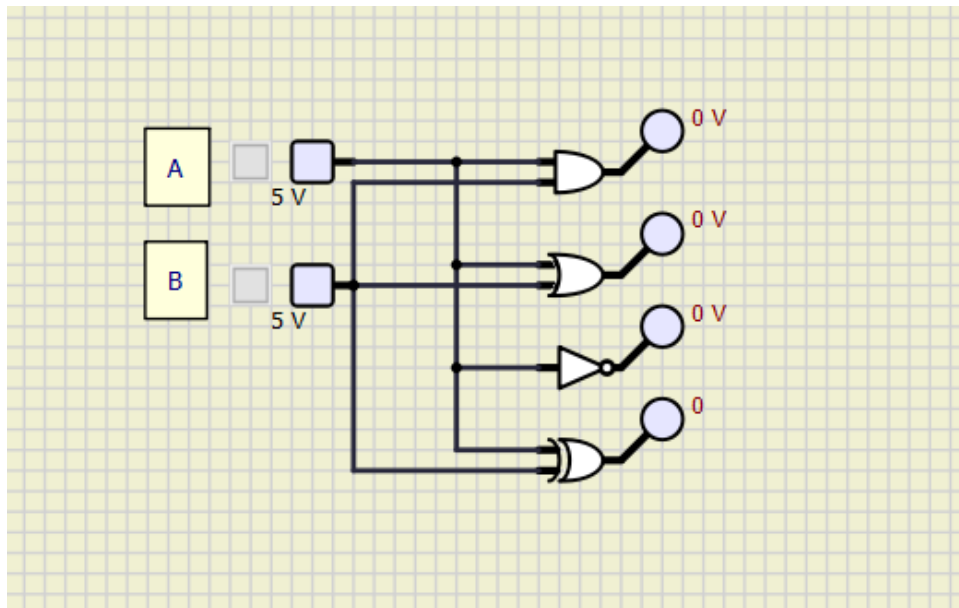


**Kamil Kałwelis 16943/IT/2020/NS**

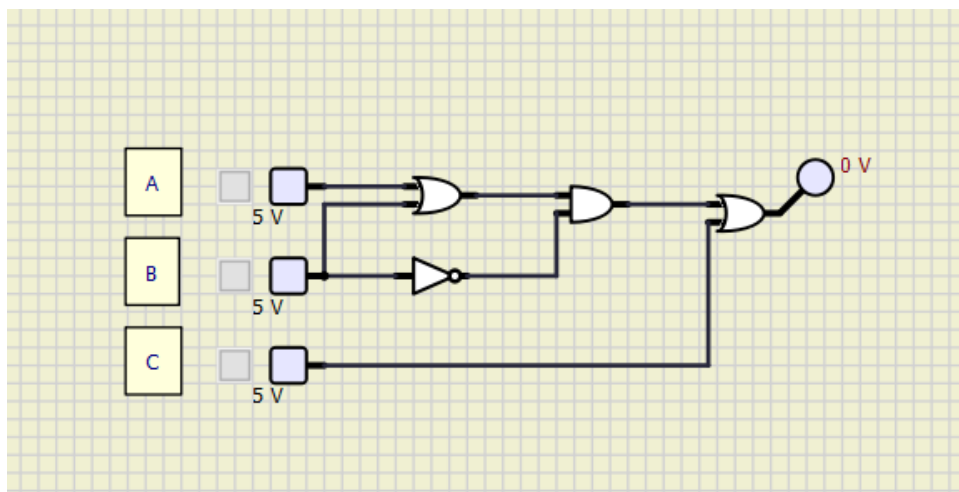
**ARCHITEKTURA  
KOMPUTERÓW  
LAB2**

**Kamil Kałwelis 16943/IT/2020/NS**

### 5.4.1.

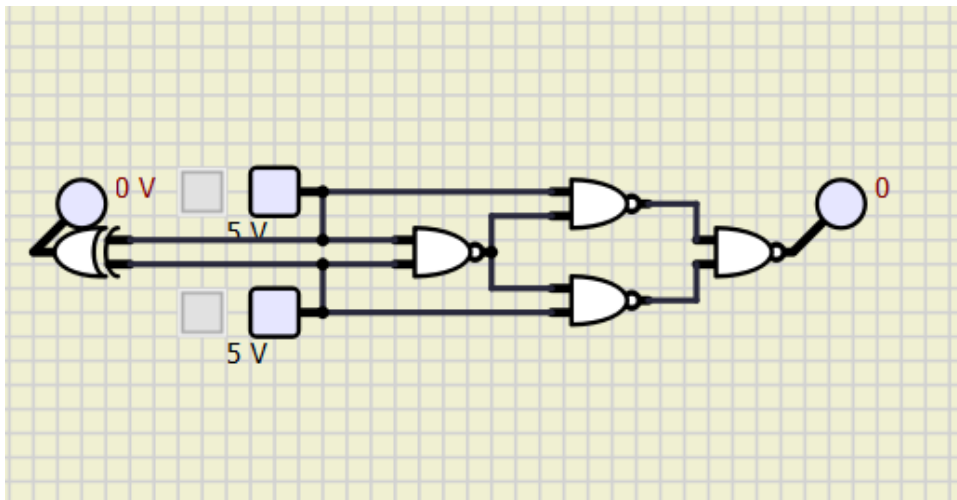


### 5.4.2.



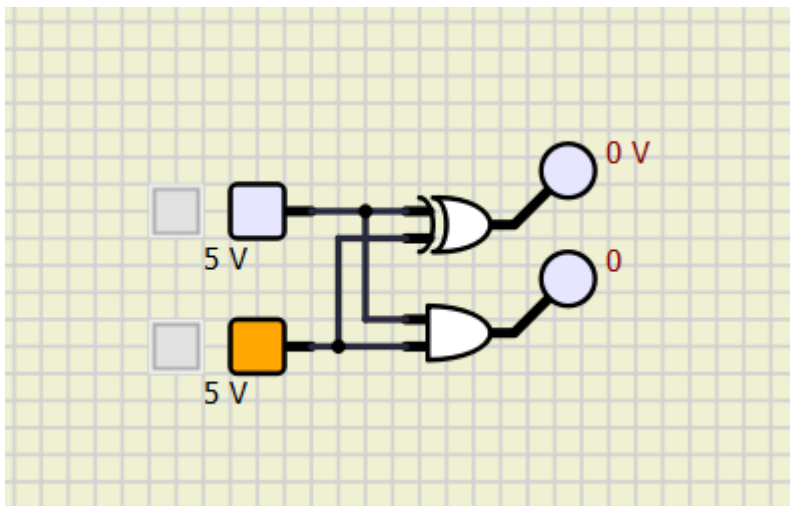
A	B	C	wynik
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

### 5.4.3.

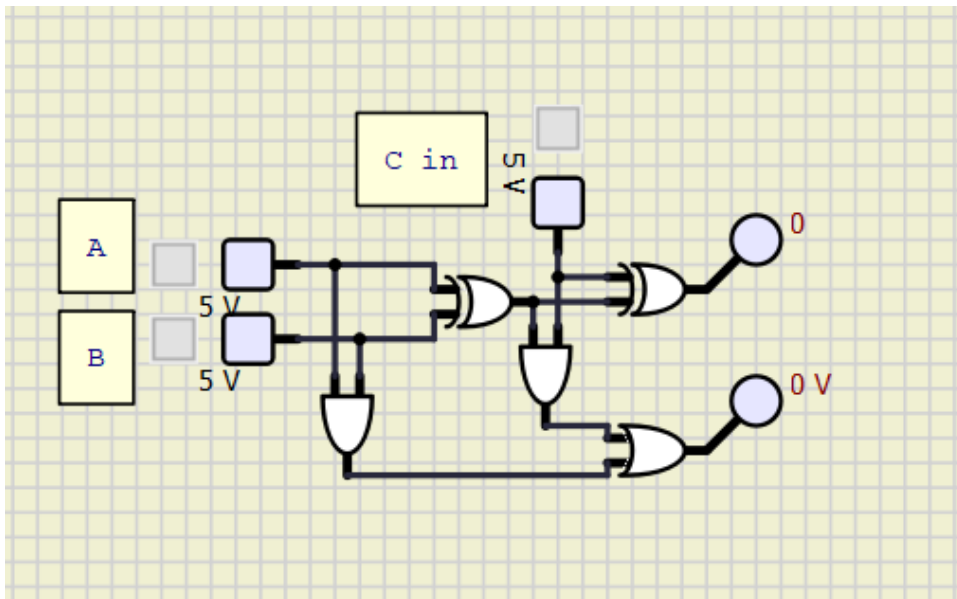


\*bramka XOR po lewej stronie schematu, została dodana w celu zobrazowania identycznego sposobu działania, co układ zbudowany po prawej stronie

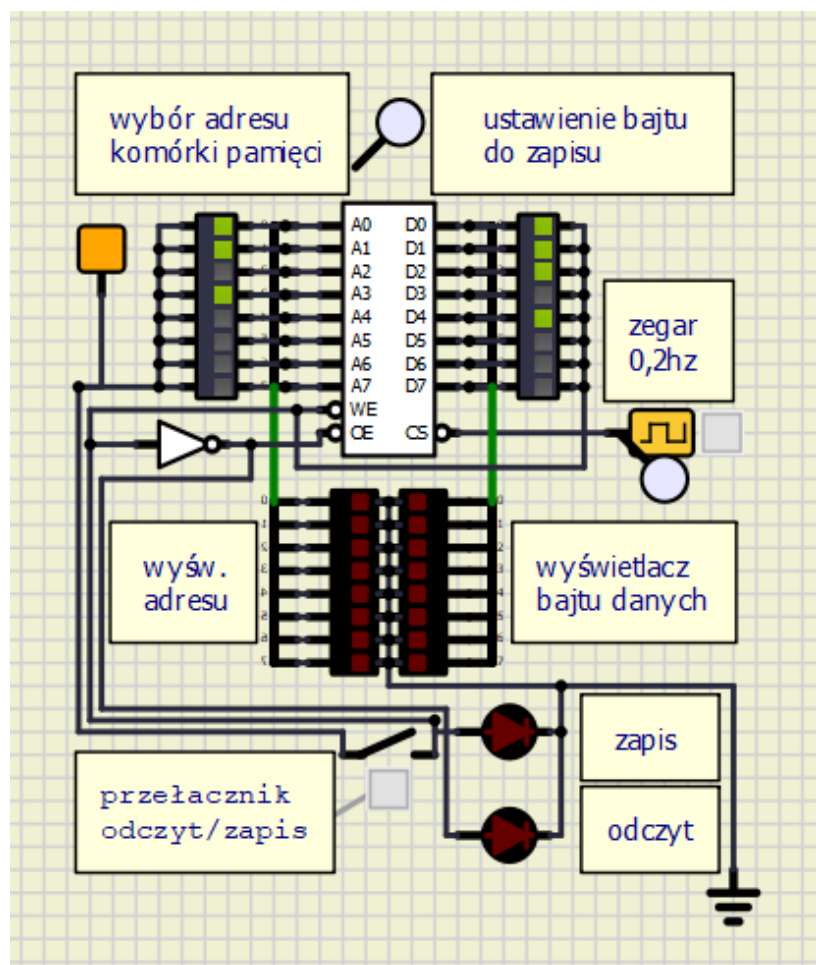
### 5.4.4.



5.4.5.



5.4.8.



# ZADANIA Z ASEMBLERA

## 9.5.2

Load me at

```
1  jmp start
2  start: nop
3  LDA 500h
4  MOV B,A
5  LDA 501h
6  STA 500h
7  MOV A,B
8  STA 501h
9  HLT
```

DataStackKeyPadMemory

Start 500h

Address (Hex)	Address	Data
0500	1280	3
0501	1281	7
0502	1282	0
0503	1283	0
0504	1284	0
0505	1285	0
0506	1286	0
0507	1287	0
0508	1288	0
0509	1289	0
050A	1290	0

Line No Assembler Message

0 Program assembled successfully

Load me at

```
1  jmp start
2  start: nop
3  LDA 500h
4  MOV B,A
5  LDA 501h
6  STA 500h
7  MOV A,B
8  STA 501h
9  HLT
```

DataStackKeyPadMemory

Start 500h

Address (Hex)	Address	Data
0500	1280	7
0501	1281	3
0502	1282	0
0503	1283	0
0504	1284	0
0505	1285	0
0506	1286	0
0507	1287	0
0508	1288	0
0509	1289	0
050A	1290	0

Line No Assembler Message

0 Program assembled successfully

```
jmp start
start: nop
LDA 500h
MOV B,A
LDA 501h
STA 500h
MOV A,B
STA 501h
HLT
```

## 9.5.3

Load me at

```
1
2 ;9.5.3
3
4 jmp start
5
6 ;data
7 zm1: db 3
8 zm2: db 4
9
10 ;code
11 start: nop
12 mvi a, zm1
13 mvi b, zm2
14 sta zm2
15 mov a, b
16 sta zm1
17
18
19
20 hlt
```

Data				Stack	Keypad	Memory
Address	Variable	Value	Value (Decimal)			
4204	zm2	03h	3			
4203	zm1	04h	4			

Line No	Assembler Message
0	Program assembled successfully

;9.5.3

jmp start

;data

zm1: db 3

zm2: db 4

;code

start: nop

mvi a, zm1

mvi b, zm2

sta zm2

mov a, b

sta zm1

hlt

## 9.5.4

Load me at

```
1
2 ;9.5.4
3
4 jmp start
5
6 ;code
7 start: nop
8
9 lda 2050h
10 mov b, a
11 lda 2052h
12 add b
13 sta 2060h
14
15 lda 2051h
16 mov b, a
17 lda 2053h
18 adc b
19 sta 2061h
20
21
22 hlt
```

Data Stack KeyPad **Memory**

Start 2050h

Address (Hex)	Address	Data
2050	8272	45
2051	8273	66
2052	8274	22
2053	8275	33
2054	8276	0
2055	8277	0
2056	8278	0
2057	8279	0
2058	8280	0
2059	8281	0
205A	8282	0

Line No Assembler Message

0 Program assembled successfully

Load me at

```
1
2 ;9.5.4
3
4 jmp start
5
6 ;code
7 start: nop
8
9 lda 2050h
10 mov b, a
11 lda 2052h
12 add b
13 sta 2060h
14
15 lda 2051h
16 mov b, a
17 lda 2053h
18 adc b
19 sta 2061h
20
21
22 hlt
```

Data Stack KeyPad **Memory**

Start 2050

Address (Hex)	Address	Data
205C	8284	0
205D	8285	0
205E	8286	0
205F	8287	0
2060	8288	67
2061	8289	99
2062	8290	0
2063	8291	0
2064	8292	0
2065	8293	0
2066	8294	0

Line No Assembler Message

0 Program assembled successfully

;9.5.4

jmp start

start: nop

lda 2050h

mov b, a

lda 2052h

add b

sta 2060h

lda 2051h

mov b, a

lda 2053h



```
adc b
sta 2061h
hlt
```

## 9.5.5

Load me at

```
1
2 ;9.5.5
3
4 jmp start
5
6 ;data
7 arg1: db 2
8 arg2: db 4
9 wynik: db 0
10
11 ;code
12 start: nop
13 lda arg1
14 sta wynik
15 mov b, a
16 lda arg2
17 cmp b
18 jc skip
19 lda arg2
20 sta wynik
21 skip: nop
22 hlt
```

Address	Variable	Value	Value (Decimal)
4205	wynik	04h	4
4203	arg1	02h	2
4204	arg2	04h	4

Line No	Assembler Message
0	Program assembled successfully




```
;9.5.5
jmp start
;data
arg1: db 2
arg2: db 4
wynik: db 0

;code
start: nop
lda arg1
sta wynik
mov b, a
lda arg2
cmp b
jc skip
lda arg2
sta wynik
```

skip: nop

Hlt

## 9.5.4

 Data  Stack  KeyPad Memory			
Address	Variable	Value	Value (Decimal)
420D	pozycja	05h	5
▶ 4203	tab	10h	16
420C	maks	12h	18
420B	ile	08h	8

;9.5.6

jmp start

;data

tab: db 16,10,2,3,18,5,6,0

ile: db 8

maks: db 0

pozycja: db 0

licznik: db 1

;code

start: nop

lxi h, tab

lda tab

sta maks

mov b, a

lda licznik

sta pozycja

back: inx h

lda licznik

inr a

sta licznik

mov a, m

cmp b

jc back

jz skip

jnc store

store: sta maks

mov b, a

lda licznik

sta pozycja

jnz back

skip: nop

hlt