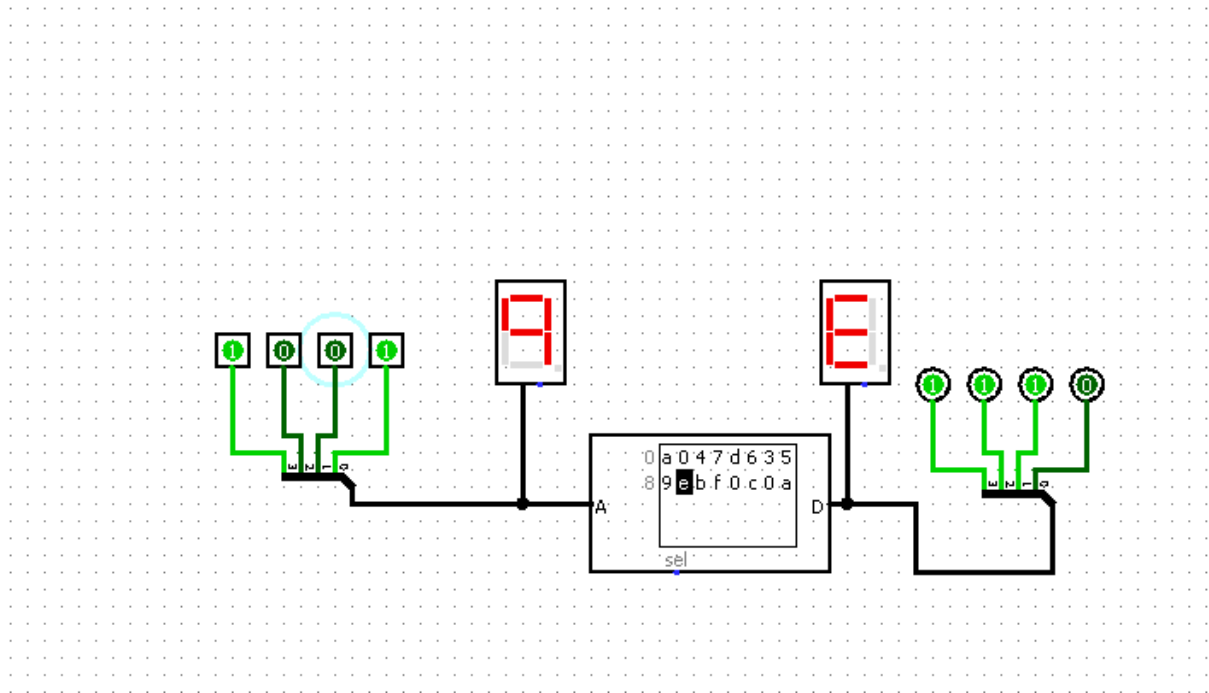


**Realizacja układów kombinacyjnych z wykorzystaniem multipleksera i pamięci ROM.  
Układy arytmetyczne.**

Zadanie 2:Zadanie 4d:

A	B	C	Różnica	Pożyczka
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Różnica				
A\BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Pożyczka				
A\BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

Różnica:  $ab'c' + a'b'c + abc + a'bc'$

Pożyczka:  $a'c + bc + a'b$

### Zadanie 7:

Output:

usa 000F ua 0000000F ulla 0000000000000000F

sa 000F a 0000000F lla 0000000000000000F

usb FFFB ub 0000FFFB ullb 000000000000FFFB

sb FFFB b FFFFFFFB llb FFFFFFFF

### Zadanie 8:

- a) Bin: 0000 0000 0000 0101, hex: 0x0005
- b) Bin: 1111 1111 1111 1011, hex: 0xFFFB
- c) Bin: 1111 1111 1111 1111, hex: 0xFFFF
- d) Bin: 0111 1111 1111 1111, hex: 0x7FFF
- e) Bin: 1000 0000 0000 0000, hex: 0x8000
- f) Największa: 32767, Najmniejsza: -32768

### Zadanie 9:

Typ short w języku C jest typem całkowitym, który (zazwyczaj) zajmuje 16 bitów w pamięci. Oznacza to, że może on przechowywać liczby w zakresie [-32768, 32767].

$121 + 34 = 155$  – wynik poprawny, suma w zakresie [-32768, 32767]

$20000 + 1000 = 21000$  – wynik poprawny, suma w zakresie [-32768, 32767]

$20000 + 30000 = -15536$  – wynik niepoprawny, przepełnienie zakresu (overflow), suma wynosi 50000, więc suma poza zakresem [-32768, 32767]

$-20000 + (-3000) = -23000$  – wynik poprawny, suma w zakresie [-32768, 32767]

$-20000 + (-25000) = -45000$  – wynik niepoprawny, przepełnienie zakresu (overflow), suma wynosi -45000, więc suma poza zakresem [-32768, 32767]