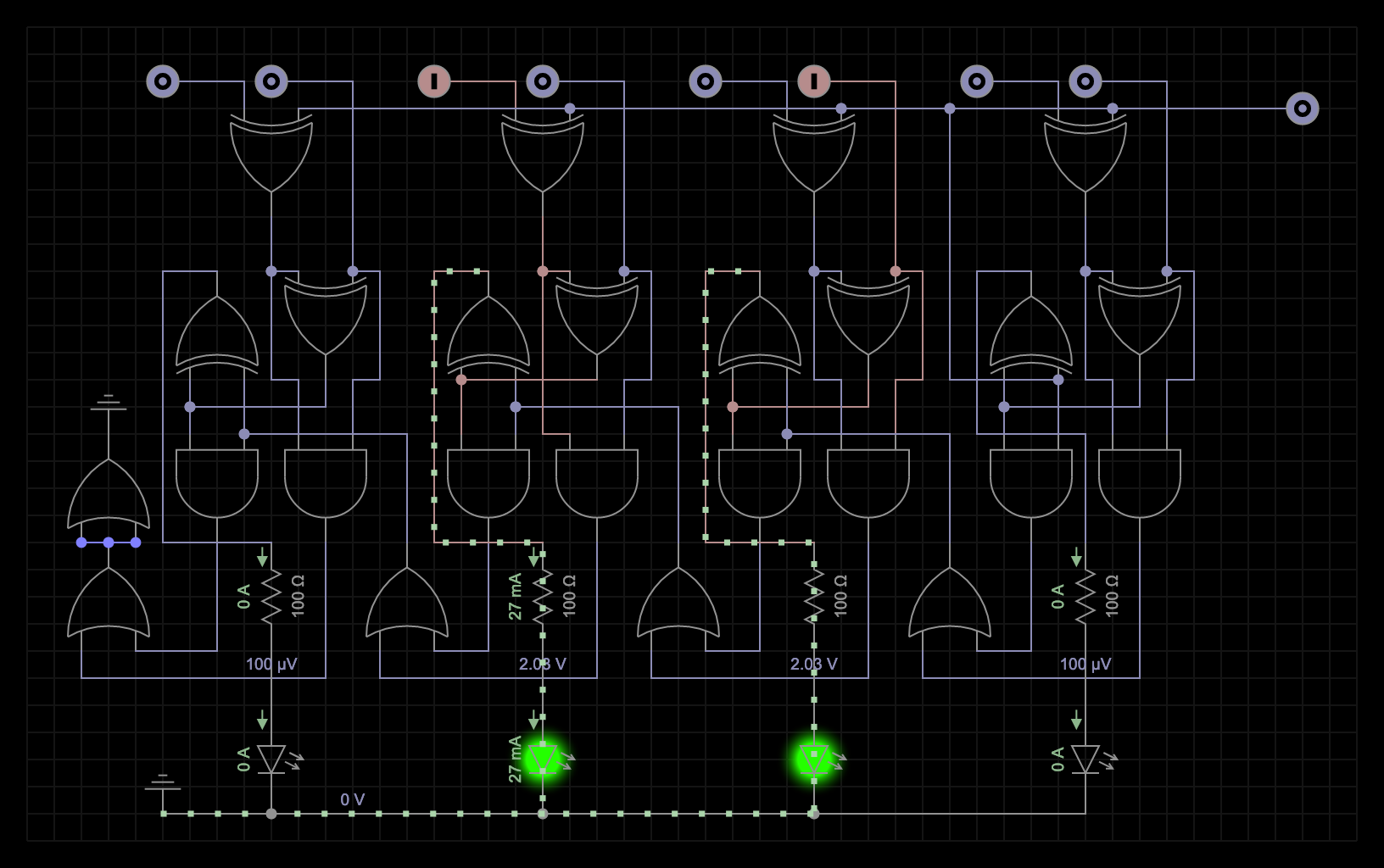
Oscar Kim

Part 1:

4 + 2 = 6

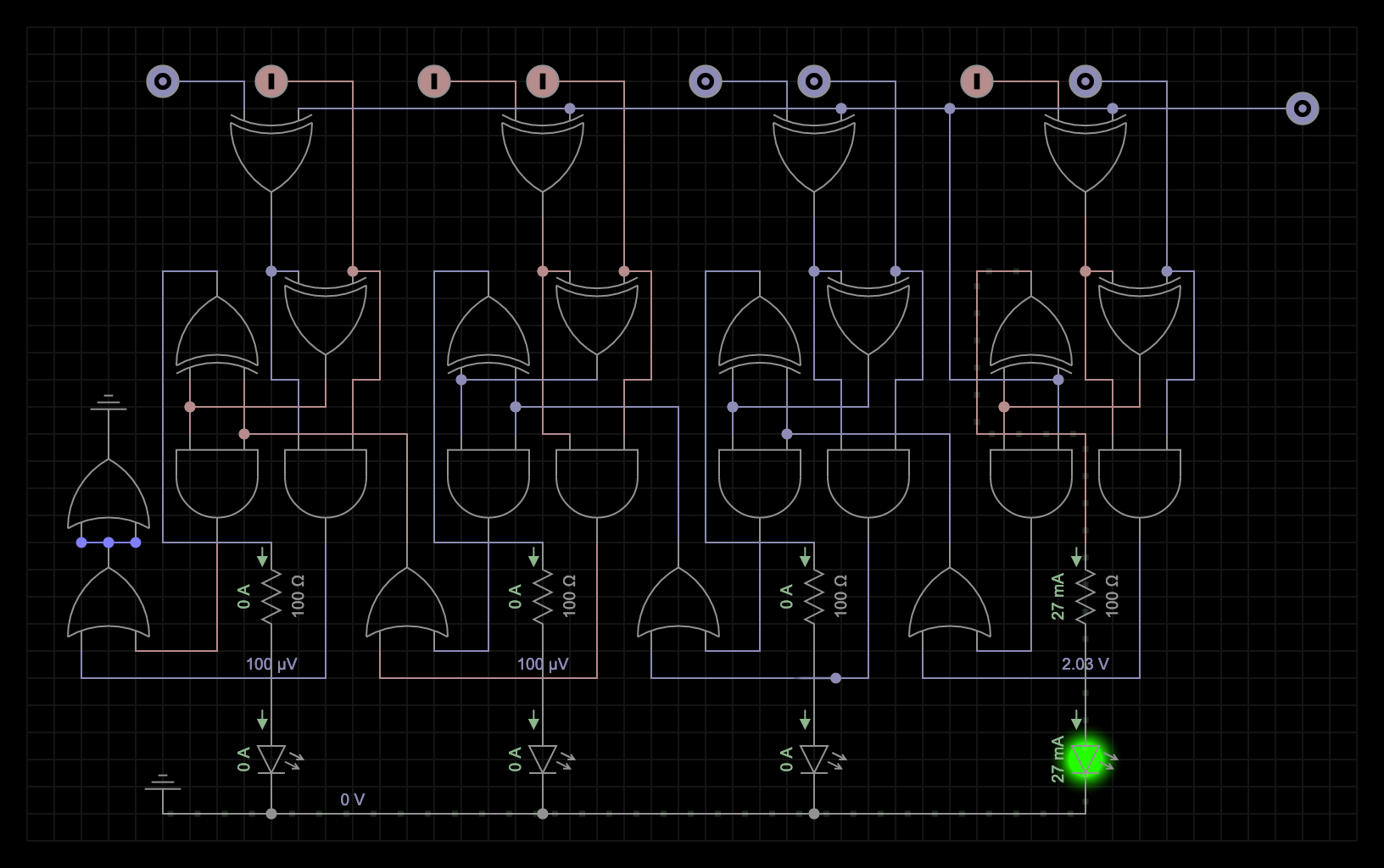
0100 + 0010 = 0110



5 - 4 = 1

0101 + ( - 0100 ) = 0001

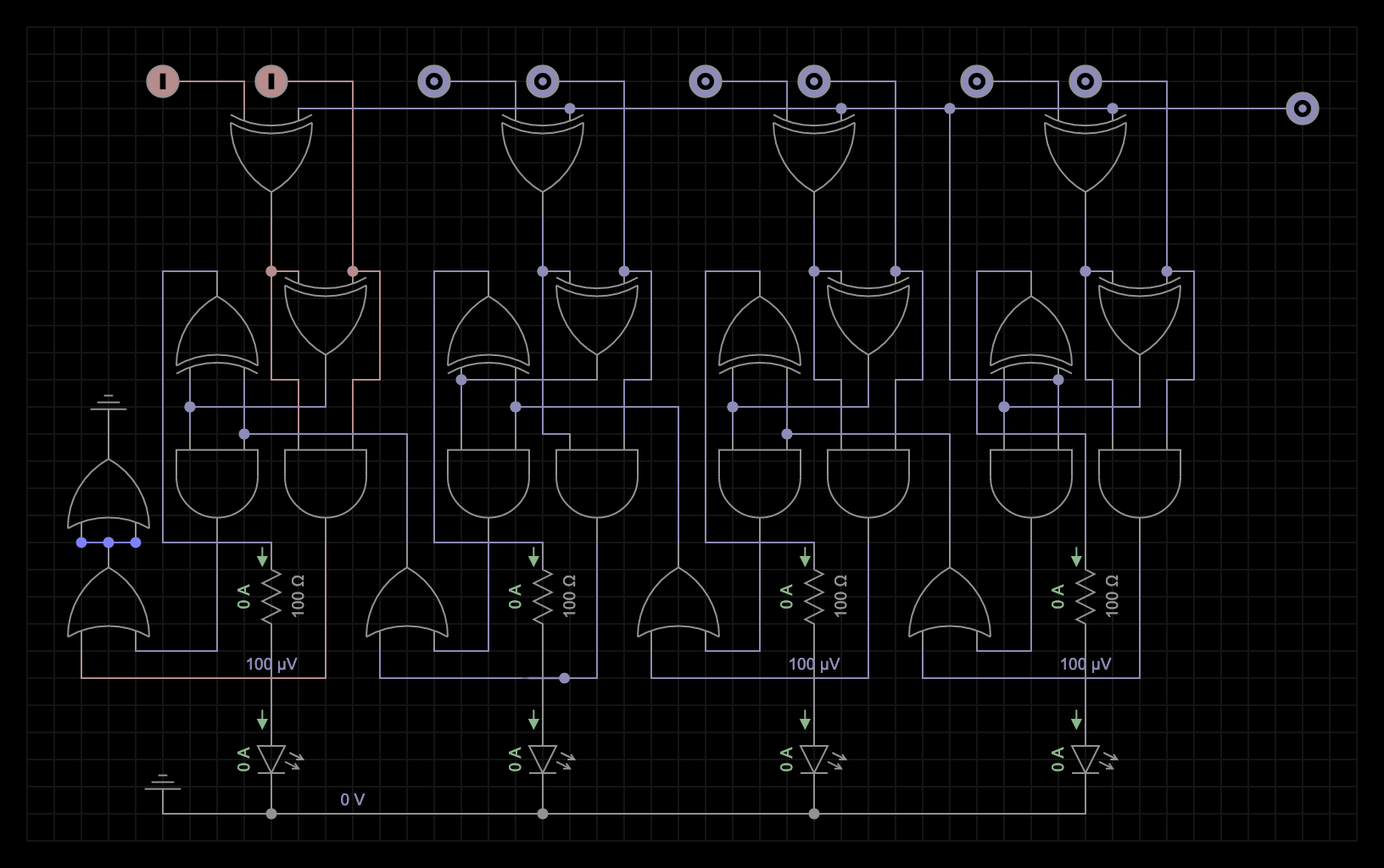
0101 + 1100 = 0001



8 - 8 = 0

1000 + (- 1000) = 0000

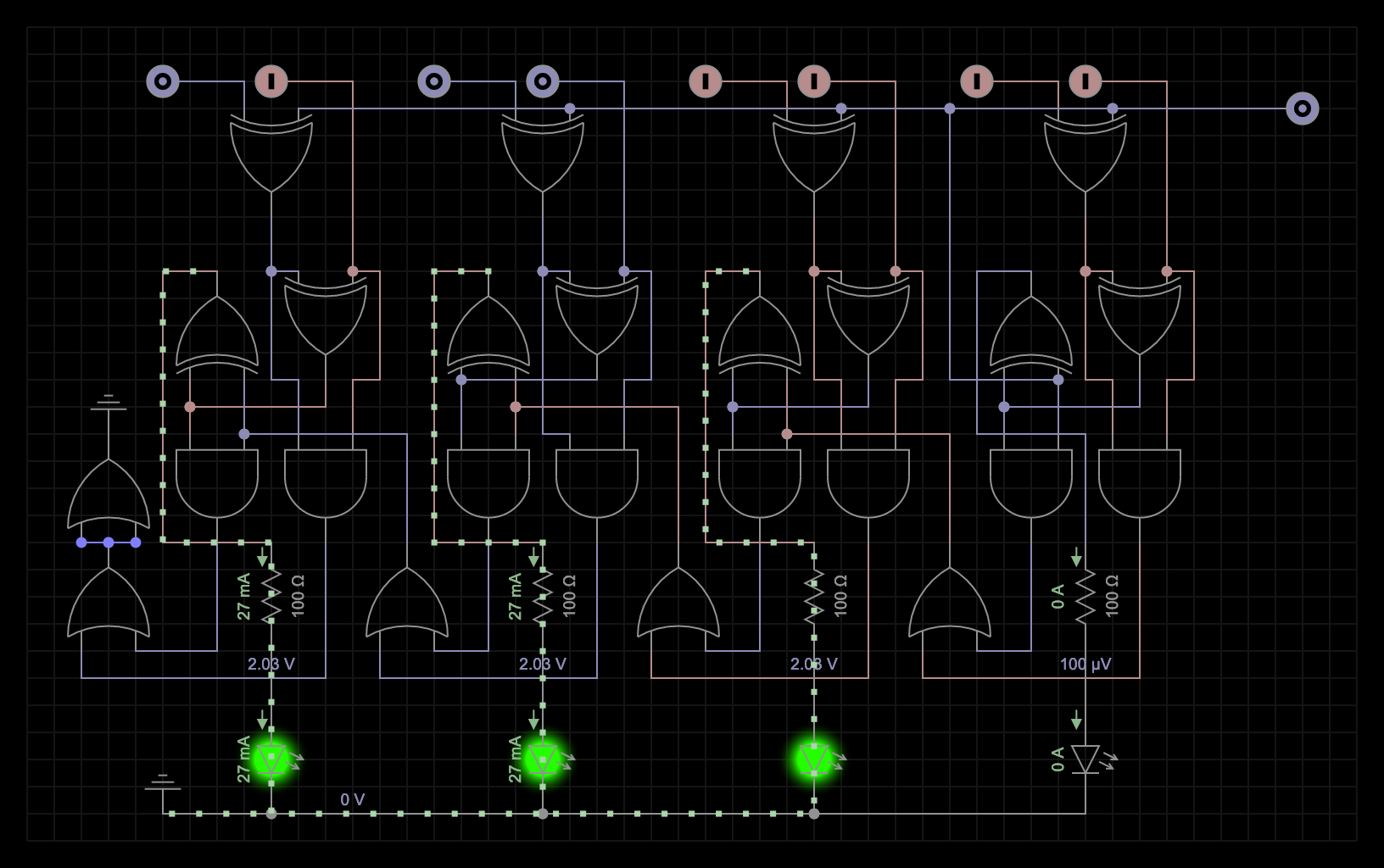
1000 + 1000 = 0000



3 - 5 = -2

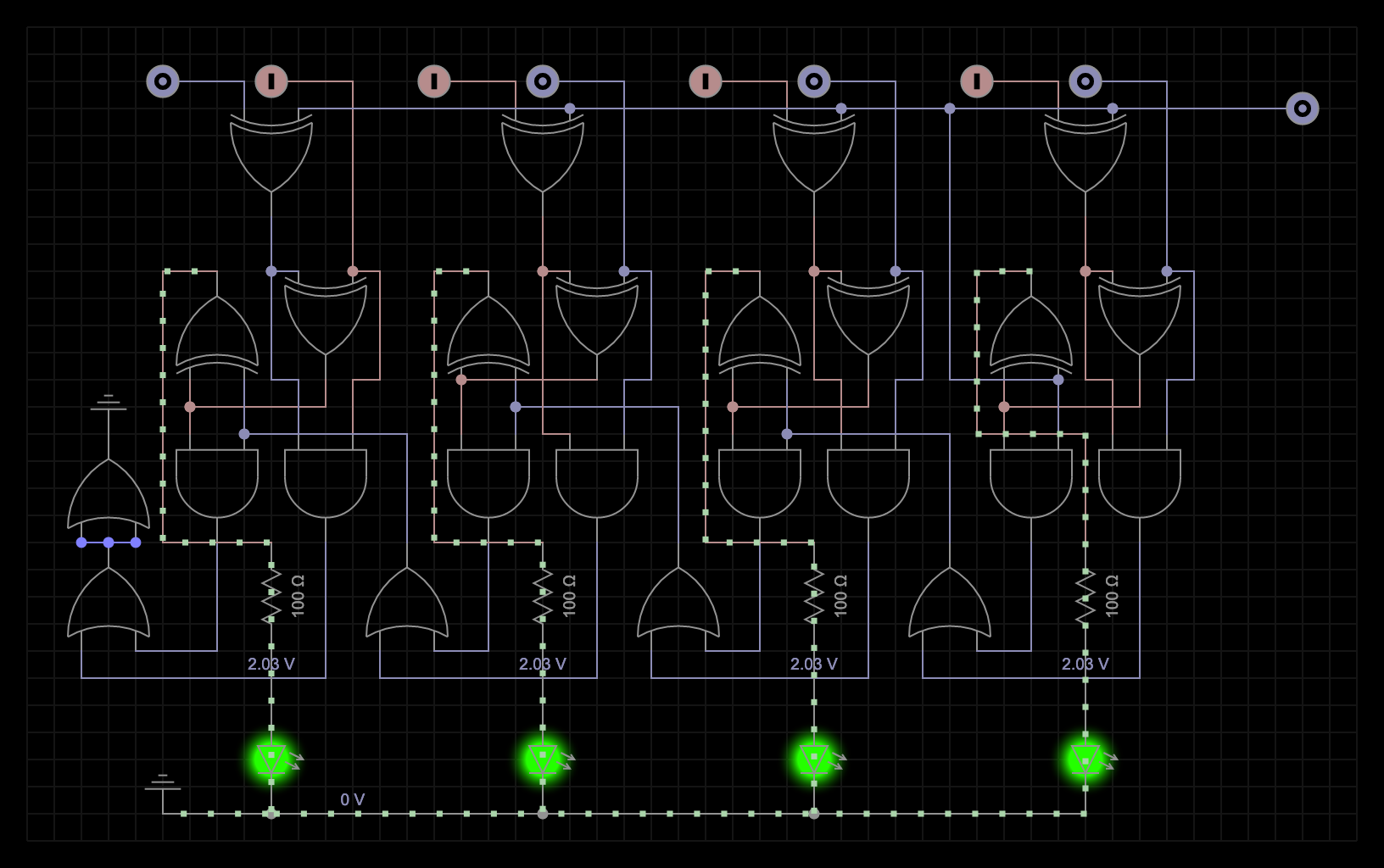
0011 + (- 0101) = - 0010

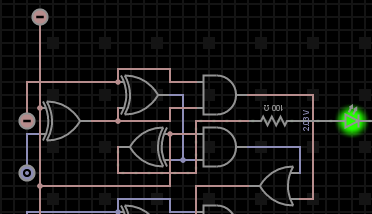
0011 + 1011 = 1110



7 + 8 = 15

0111 + 1000 = 1111





| A | B | A xor B |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 |  |
| 0 | 1 | 1 | this is the one! |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

The A bit is the blue bit on the bottom, and the B bit is the red bit above it.

The two bits are being passed through an XOR gate and the row highlighted in yellow is the row that corresponds to the picture above. The third column titled A xor B corresponds to the bit that is flashing green in the picture. The A bit is off and the B bit is on therefore the result is 1(turned on green).

Part 2:

I am all set up with visual studio.

Part 3:

I am all set up with masm.

Part 4:

1A.)

Convert 16 bit binary to base 10 integer:

int **bin\_dec**(**string** bin){

*// long long dec = 0;*

int dec = 0;

for(int i = 0 ; i < bin.**size**() ; i++){

if(bin**[**i**]** == '1'){

dec += **pow**(2 , (bin.**size**() - 1 - i));

}

}

return dec;

}

1B.)

Convert from hexadecimal to base 10

Step one: Convert from hex to binary;

string hex\_bin(string hex){

string reverse = “”;

string bin = “”;

int current = 0;

for(int i = 0 ; i < hex.size() ; i++){

//we are going to pretend like characters such as A and F in this pseudocode will

//just be integers but in the actual program i will of course

//convert them to integers.

current = hex[i]; // we assume each character is an int for this pseudocode

for(int c = 0 ; c < 4 ; c++){

reverse += to\_string(current % 2);

current /= 2;

}

for(int i = reverse.size() -1 ; i >=0 ; i--){

bin += reverse[i];

}

reverse = “”;

}

return bin;

}

Step two: convert from binary to base 10

int **bin\_dec**(**string** bin){

*// long long dec = 0;*

int dec = 0;

for(int i = 0 ; i < bin.**size**() ; i++){

if(bin**[**i**]** == '1'){

dec += **pow**(2 , (bin.**size**() - 1 - i));

}

}

return dec;

}

1C.)

Convert base 10 to binary

**string** **dec\_bin**(**string** dec){

int number = **stoi**(dec);

**string** reverse = "";

**string** bin = "";

while(number/2 !=0){

reverse **+=** **to\_string**(number % 2);

number /= 2;

}

reverse **+=** **to\_string**(number % 2);

for(int i = reverse.**size**() - 1 ; i >= 0 ; i --){

bin **+=** reverse**[**i**]**;

}

return bin;

}

1D.)

Step one: convert from base 10 to binary

**string** **dec\_bin**(**string** dec){

int number = **stoi**(dec);

**string** reverse = "";

**string** bin = "";

while(number/2 !=0){

reverse **+=** **to\_string**(number % 2);

number /= 2;

}

reverse **+=** **to\_string**(number % 2);

for(int i = reverse.**size**() - 1 ; i >= 0 ; i --){

bin **+=** reverse**[**i**]**;

}

return bin;

}

Step two: convert from binary to hexadecimal

string bin\_hex(string bin){

int start =4 - (bin.size() % 4); //if our binary size is not divisible by 4 we add zeros

//to the right until it is

string starting\_zeros = “”;

string hex = “”;

int value = 0;

for(int i = 0 ; i < start ; i++){

starting\_zeros += “0”;

}

bin = starting\_zeros + bin;

for(int i = 0 ; i < bin.size() ; i++){

value = 0 ;

if(bin[i] == ‘1’) value += 8;

i++;

if(bin[i] == ‘1’) value += 4;

i++;

if(bin[i] == ‘1’) value += 2;

i++;

if(bin[i] == ‘1’) value += 1;

i++;

hex += value; // we assume values like 10 or 12 become letters in

// this pseudocode but i will of course convert them

// to actual letters in the program

}

return hex;

}