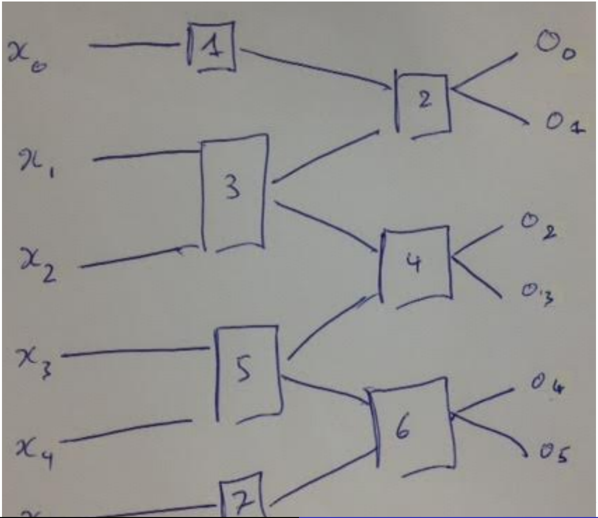
Table of contents

Does a 6 bit counter exists ?

Can we create a **deterministic** network that will iterate over all 6 bits string – in any order – ?

Does a 6 bit counter exists ?



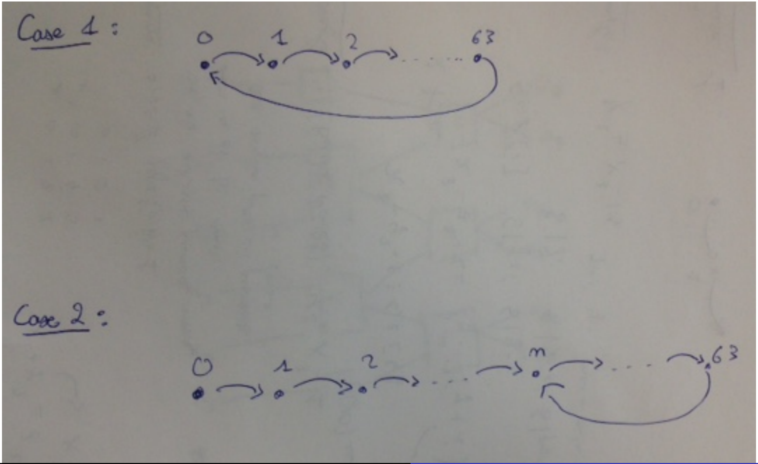
Can we bruteforce that problem ?

There are $\mathbf{2^{44}}$ differents networks which is approximatively 17000 billions.
We could hope for an answer in a few days.
But we can drastically restrict our search space with a **few** observations.

Main idea : we are going to bruteforce on **sub networks** and select those who respect a **necessary** condition.

How such a network's dynamic would look like ?

There are only two possibilities (up to any permutation):



Implication on the computed function by the network

The function that our counting network coumputes is either:

- **A bijection**
- **An almost-bijection** f i.e, $\exists! x_0, y_0 \in \{0,1\}^6$ such that f' is a bijection where:

$$\forall x \neq x_0 \quad f'(x) = f(x)$$
$$f'(x_0) = y_0$$

We are going to check on both cases.

The $4 * 16$ property

If we have a bijection it will **enumerate** all strings of $\{0,1\}^6$.
After reordordering it will look like:

```
0000000000000000000000000000000011111111111111111111111111111111
0000000000000000111111111111111100000000000000001111111111111111
0000000011111111000000001111111100000000111111110000000011111111
0000111100001111000011110000111100001111000011110000111100001111
0011001100110011001100110011001100110011001100110011001100110011
0101010101010101010101010101010101010101010101010101010101010101
```

The $4 * 16$ property

Let's focus on the first two bits, we can organise our sequences this way:

0000000000000000	0000000000000000	1111111111111111	1111111111111111
0000000000000000	1111111111111111	0000000000000000	1111111111111111
0000000011111111	0000000011111111	0000000011111111	0000000011111111
0000111100001111	0000111100001111	0000111100001111	0000111100001111
0011001100110011	0011001100110011	0011001100110011	0011001100110011
0101010101010101	0101010101010101	0101010101010101	0101010101010101

The $4 * 16$ property

Let get rid of the last 4 bits:

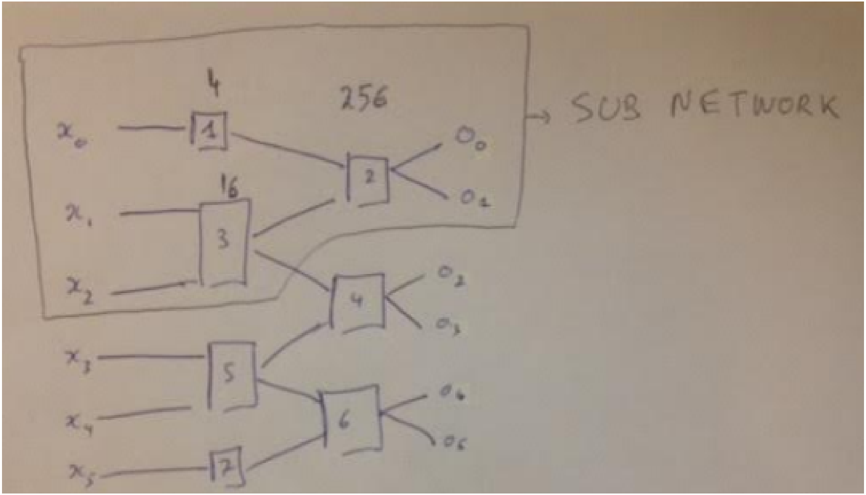
0000000000000000	0000000000000000	1111111111111111	1111111111111111
0000000000000000	1111111111111111	0000000000000000	1111111111111111

We see that each of the 2 bits patterns: **00, 01, 10, 11** occurs **16** times in this enumeration.
It's the **$4 * 16$ property**.

The $4 * 16$ property

Hence the sub network responsible for these 2 bits must have the **4*16** property to be eligible as a being a sub network of a 6 bits bijective counter network.

The $4 * 16$ property

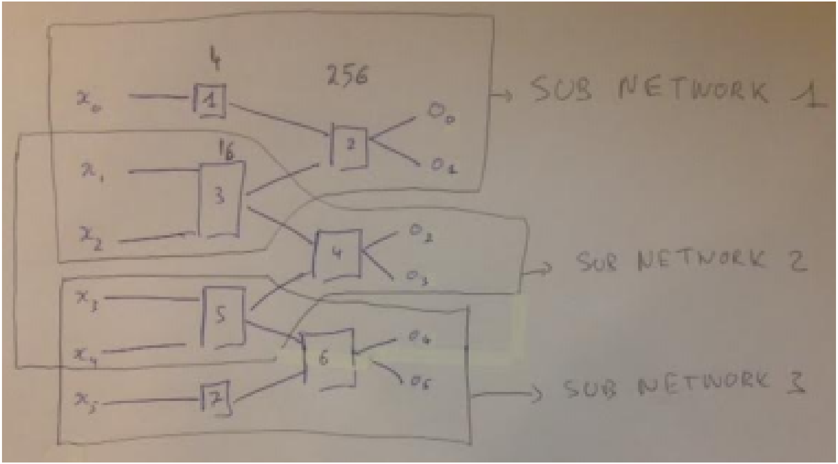


Compute all the $4 * 16$ sub networks

By exausthive search we find **288** (over $4 * 16 * 256 = 16384$) sub networks with this property.

By removing equivalent networks we are left with **72 circuits for the first 2 bits**.

4*16 holds for middle and ending bits



4*16 holds for middle and ending bits

- **72** networks for the first 2 bits
- **216** networks for the middle 2 bits
- **72** networks for the last 2 bits

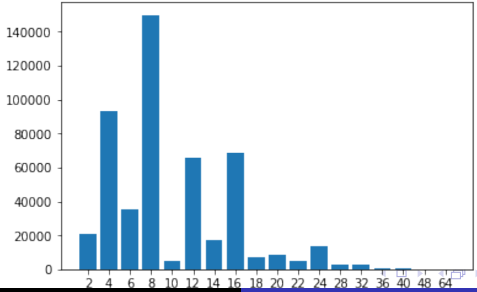
How to conclude I ? Combine it !

Hence by combining these we have $72 * 216 * 72 \simeq 10^6$ 6 bits networks to test.

How to conclude II ? Count orbits !

Over all these potential networks we count **497664** bijections. For each of these bijections we have to **count their orbits**, we have a winner **iif it has only 1 orbit**.

We do not find such a network, here there's the histogram of orbits:



Conclusion

Conclusion: There are no bijective 6bits counters.

2*161517 property

If we have an almost-bijection **all 6 bits sequences will be reached by our network but one**.

Also one bit string will be reached **twice**.

It means that **at least one** of our sub network will see:

- 2 patterns **16** times
- 1 pattern **15** times
- 1 pattern **17** times

2*161517 property

We call this the **2*161517 property**.

2*161517 does not occur !

By enumeration, **there exist no sub network with the 2*161517** property.
Hence, we cannot hope for an **almost-bijective counter**.

No 6bit counter network :'(

By case distinction, **there's no 6bit counter network**.
But there are **0 to 62** counters and this kind of methods helps to exhibit **at least 24**.

Sources ?

All our sources for these computations are available here:

`https://github.com/cosmo-sterin/ER_MolProg_Project2/tree/master/circuit_sim`