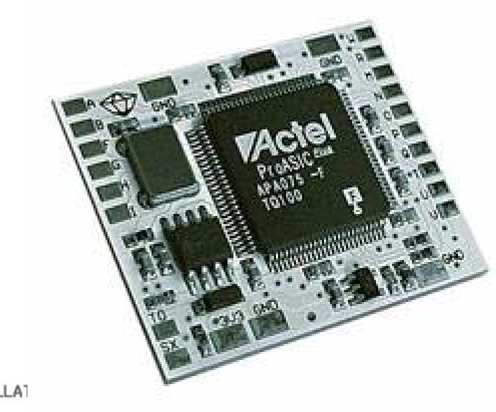
Implementing Logic Circuits With DNA

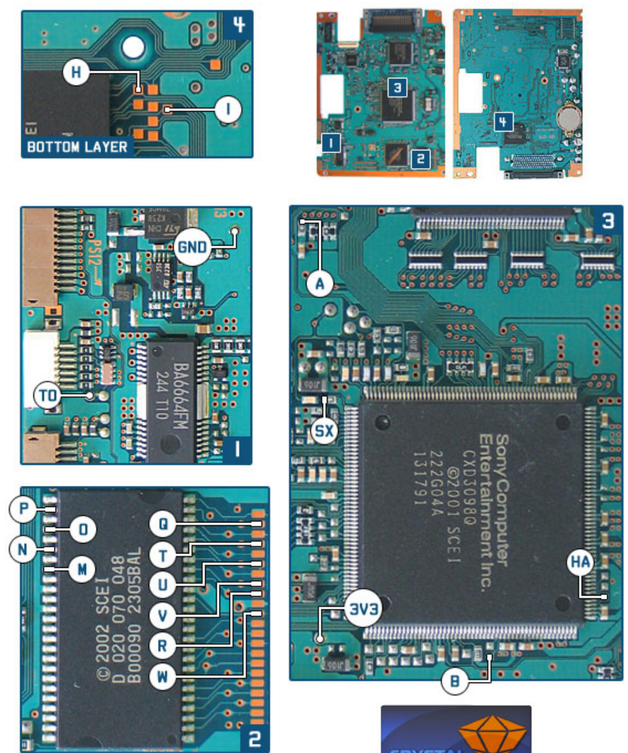
By Cancan Shi

Where can we find logic circuits?

- Logic circuits can be found in most consumer electronics
  - TVs

– game controllers





What are Logic Circuits?

- Also called digital circuits
- Uses digital signal instead of analog signal
- The most common “fundamental unit” is the logic gate
- By combining numerous gates, more complex system can be created

continued

- The complex system of digital electronics is collectively referred to as a digital circuit.
- Digital circuits are the basis of all digital electronics, such as: computers, digital cameras, mobile phones, etc.

What are logic gates?

- Performs a logical operation on one or more logic inputs and produces a single logic output
- Logic inputs and outputs are two levels, example: 0/1, high/low, true/false

Logic Gates

- There are several basic logic gates:

  AND, OR, NOT, NAND, NOR, XOR and XNOR
- NAND and NOR logic gates are the two pillars of logic, in that all other types of Boolean logic gates can be created from proper combinations of them.

DNA Computer VS. Silicon Computer

- Silicon computer:
  - According to Moore’s law and the miniaturization limitations of silicon, microprocessors made of silicon will eventually reach their limits of speed and size
- DNA computer:
  - DNA's key advantage is that it will make computers smaller than any computer that has come before them, while at the same time holding more data

continued

- Silicon Computer:
  - Toxic material are used to make silicon microprocessors
  - Conventional computers operate linearly, taking on tasks one at a time.
- DNA computer:
  - DNA biochips can be made cleanly
  - DNA computers perform calculations parallel to other calculations

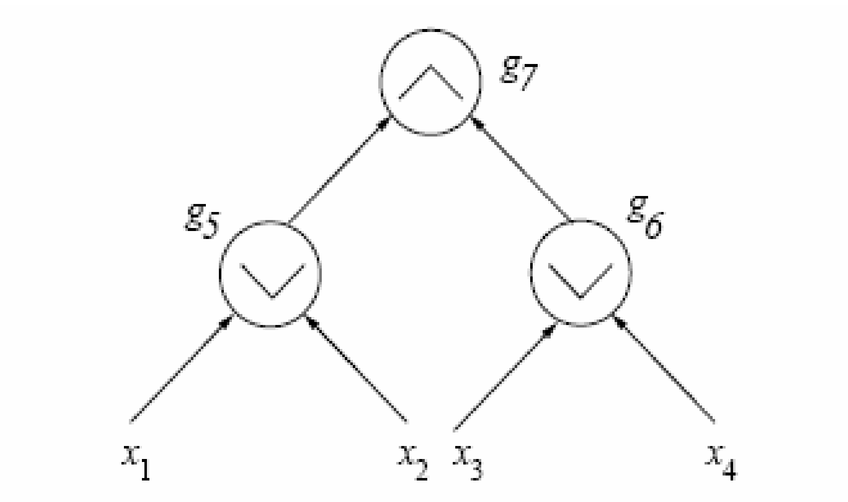
continued

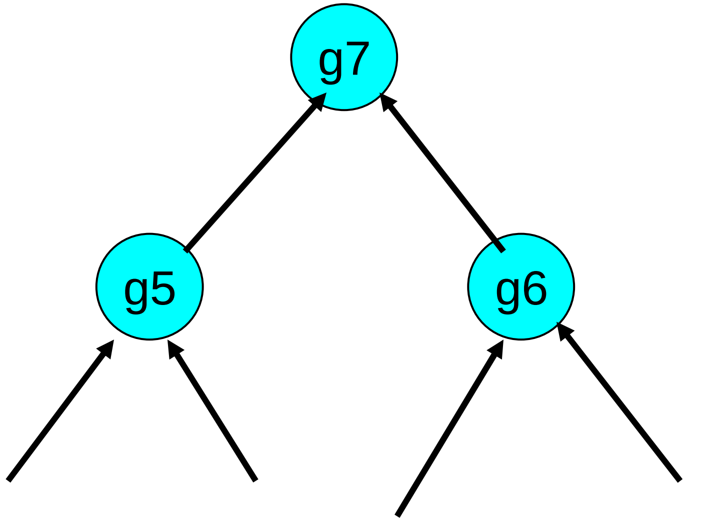
- Silicon computer:
  - Competitively not as affordable as DNA
- DNA computer:
  - Large supply / affordable source for DNA computers

Building a DNA computer

- The first step of building a DNA computer is simulating the function of logic gates (as used in silicon computers) using DNA strands.

Logic circuit can be viewed as directed acyclic graph





Review the directed Hamilton Path problem

- Use DNA sequence to build graph
- Solve the problem by utilizing the characteristics of DNA sequence
- Complementary, melt, anneal

Since logic circuits can be viewed as a directed acyclic graph, can we borrow concepts/ideas from the way we solve the “Hamilton Directed Graph” problem, to solve logic circuits

?

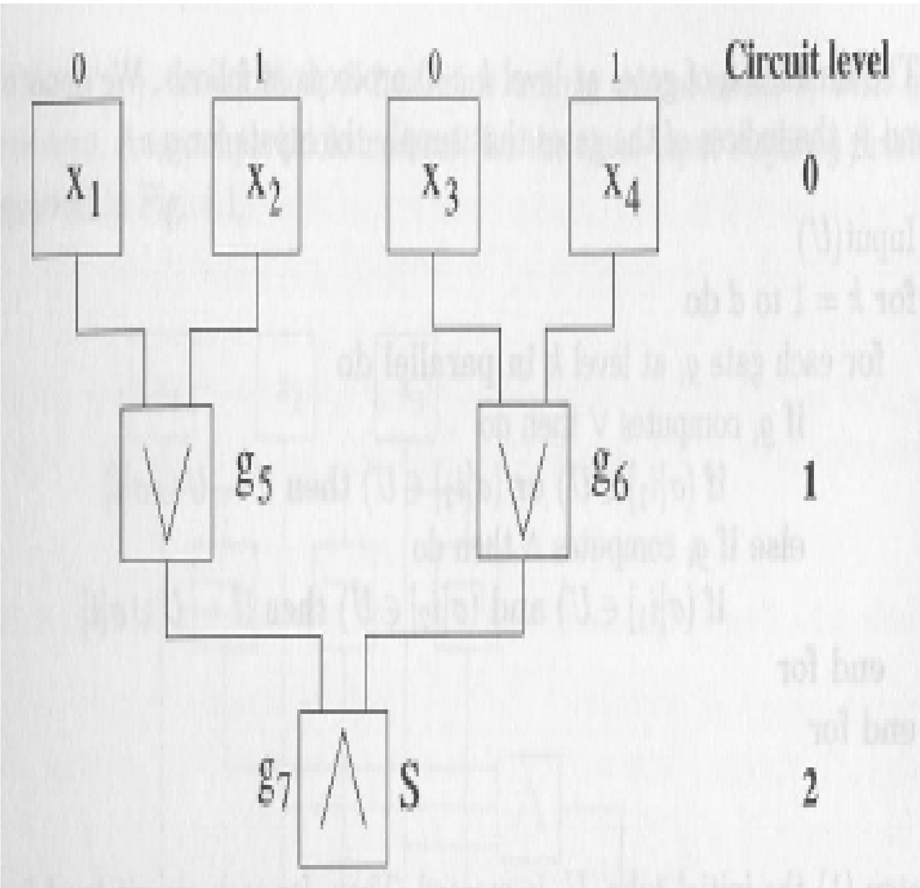
The Answer is “YES”

- In 1997 researchers (Ogihara and Ray) at the University of Rochester developed logic gates made of DNA. –using ideas from the Hamilton Directed Graph

Major Characteristics of DNA Gates

- Instead of relying on electrical signals, DNA Gates rely on DNA codes
- DNA Gates detect specific fragments of the genetic blueprint as input, then splice together the fragments to form a single output.

Terminology for Logic Circuits



- Size: The number of gates in the circuit
- Depth: The number of gates in the longest path connecting an input vertex to output gate
- Circuit Level
- Boolean tables

Gate AND

- Boolean table of logic gate “AND”

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Gate OR

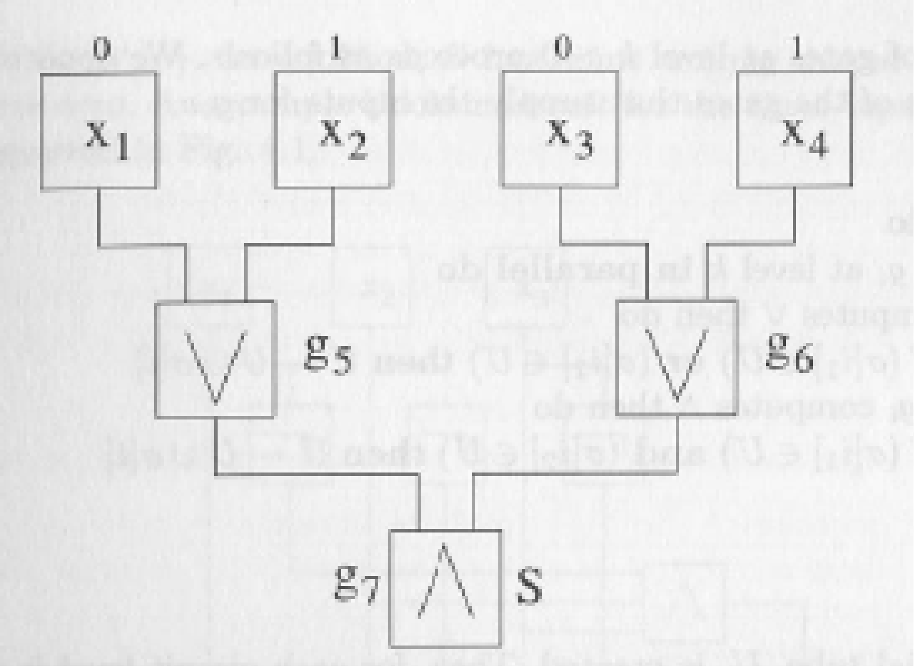
- Boolean table of gate “OR“

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

3 phases to simulate logic gate

- Set-up
- Level simulation
- Final read-out of output gates

Set Up

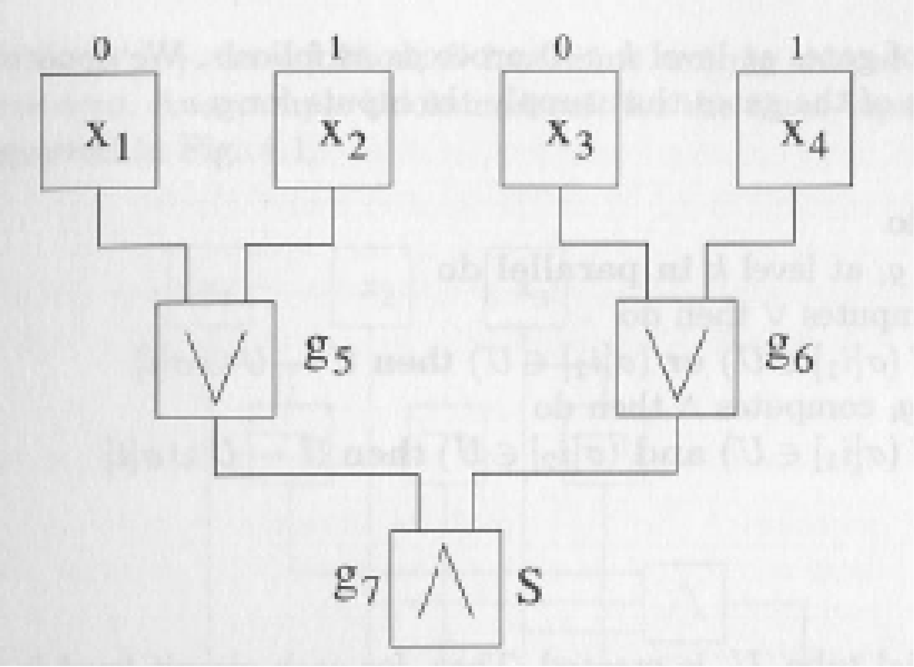


- Set up:

  1. Assign a DNA strand s[i], to input Xi, if Xi = 1. Length is l

  2. Assign a DNA strand to each gate. Length l.

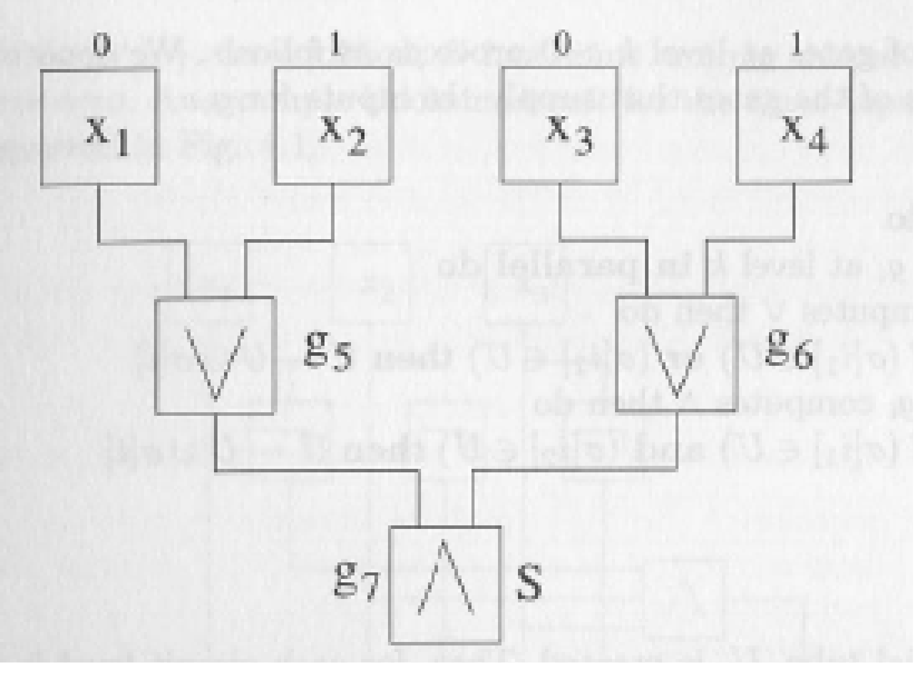
continued



3. for each edge, also assign a sequence, length l

However, this sequence has two parts, each of them is half part of the connecting gate's complementary sequence.

continued



- For example:

  X2 is

  CCCTAGTACGGG

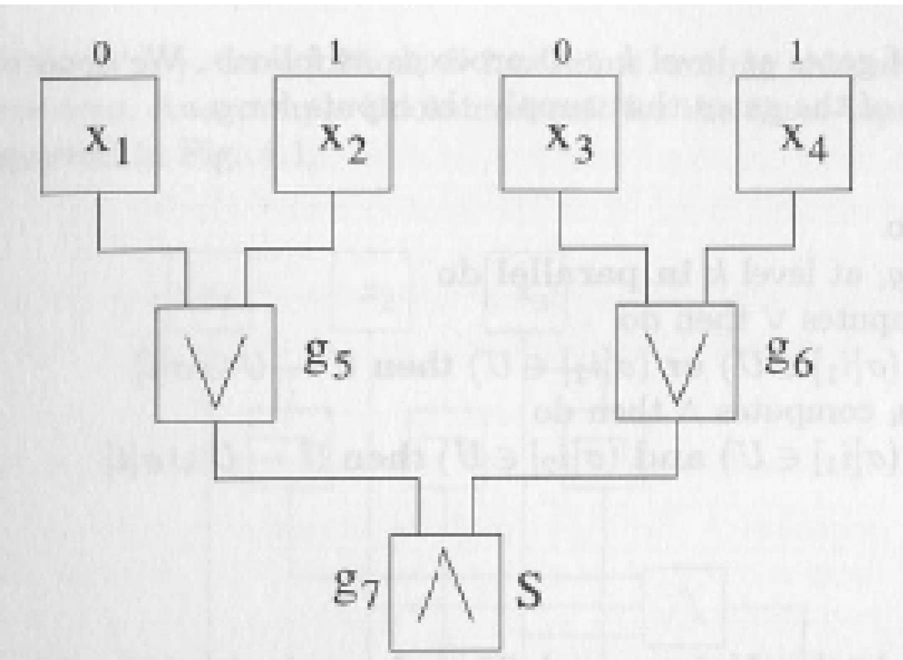
  g5 is

  CCCGATGCACCC

  Therefore, e25 is:

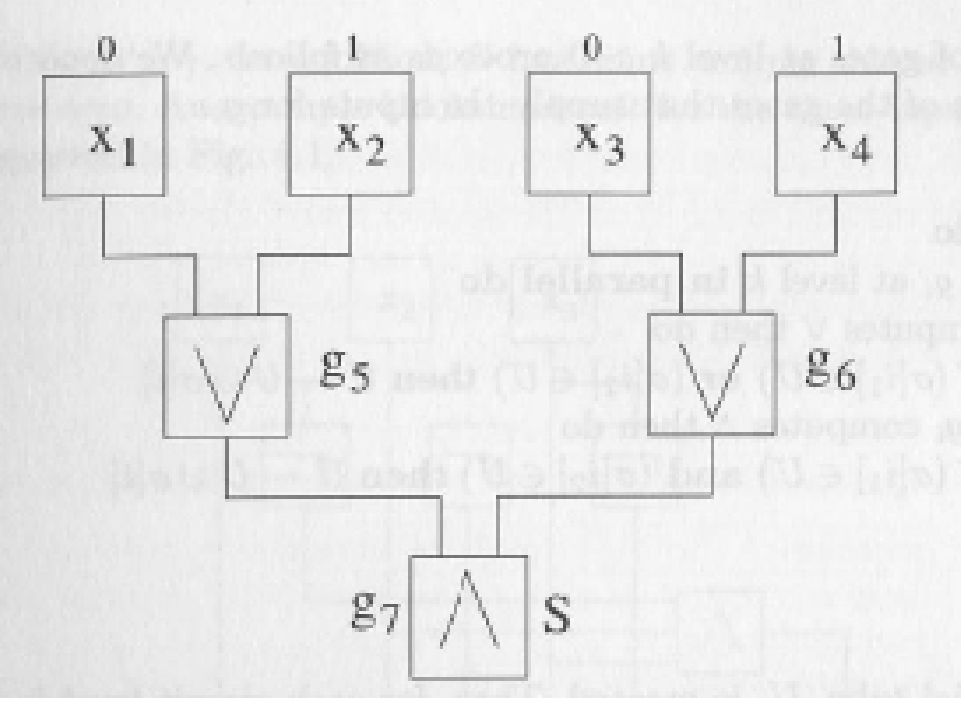
  ATGCCCGGGCTA

Level Simulation



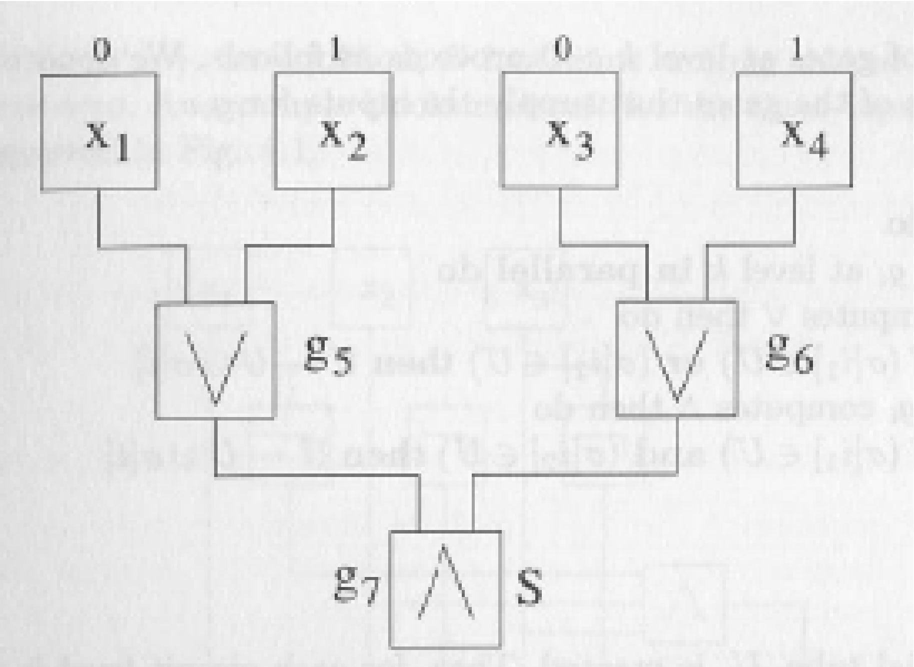
- First pour DNA strands Xi into an empty tube T0.
- Then pour T0 into T1
- T1 originally has g5 and g6

Continued



- In tube T1, there are DNA strands of X2, g5, e25, X4, g6 and e46
- Check level 1
- Two OR gates g5 and g6

Continued



- Sequence e25 anneal with X2 and g5.
- It creates a length 2l sequence.

X2 g5

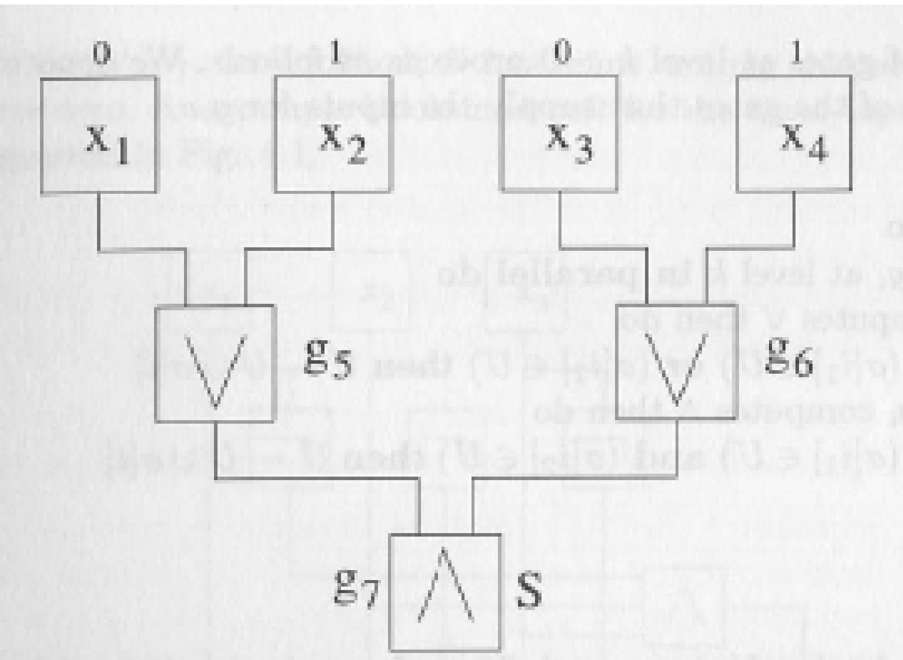
CCCTAGTACGGG|CCCGATGCACCC

| || | | | | | || ||

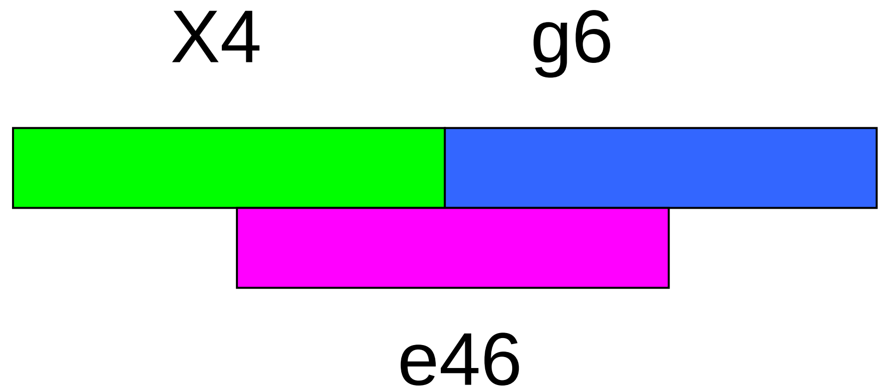
ATGCCC GGGCTA

e25

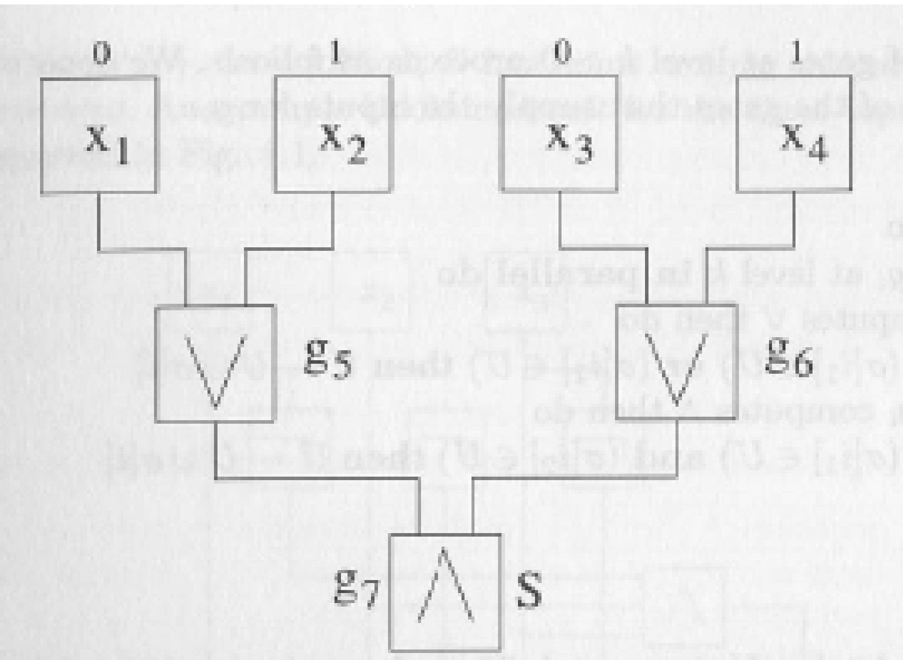
Continued



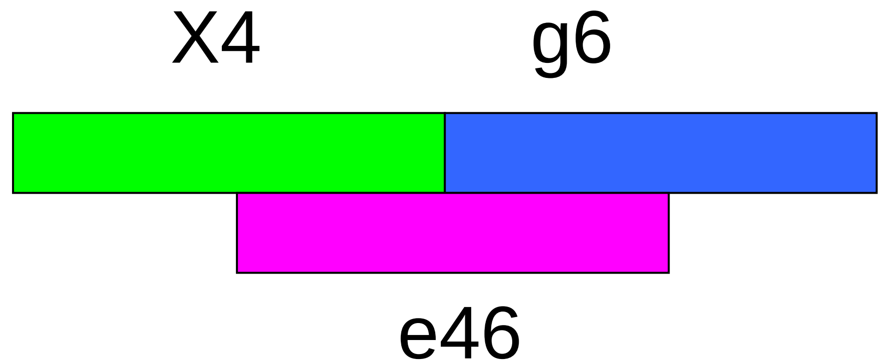
- Same as gate g6 and X4



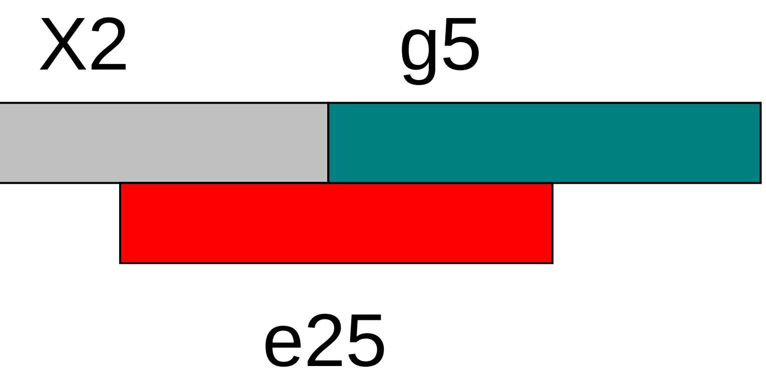
Continued



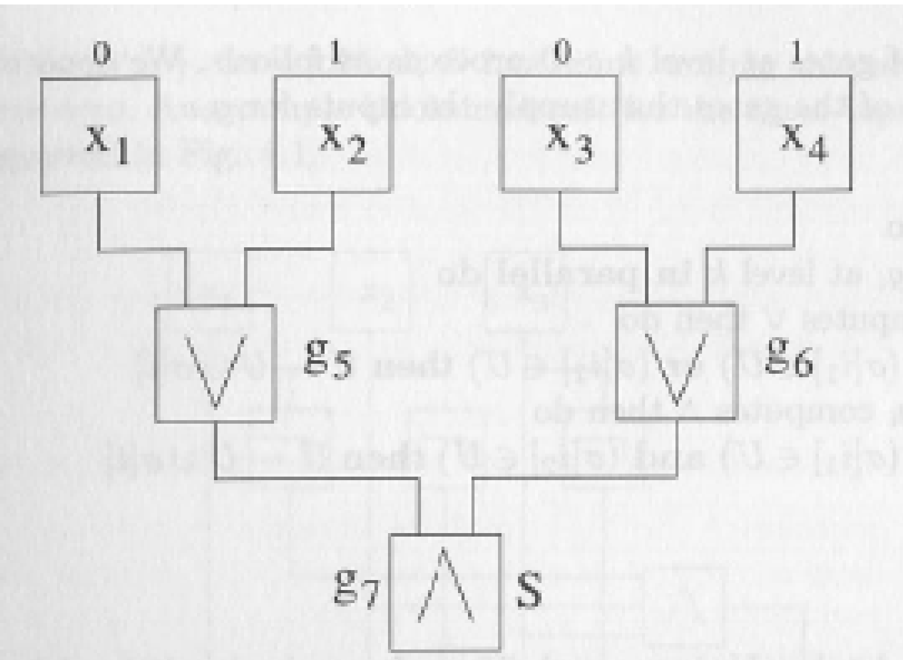
- Now in tube T1 there are



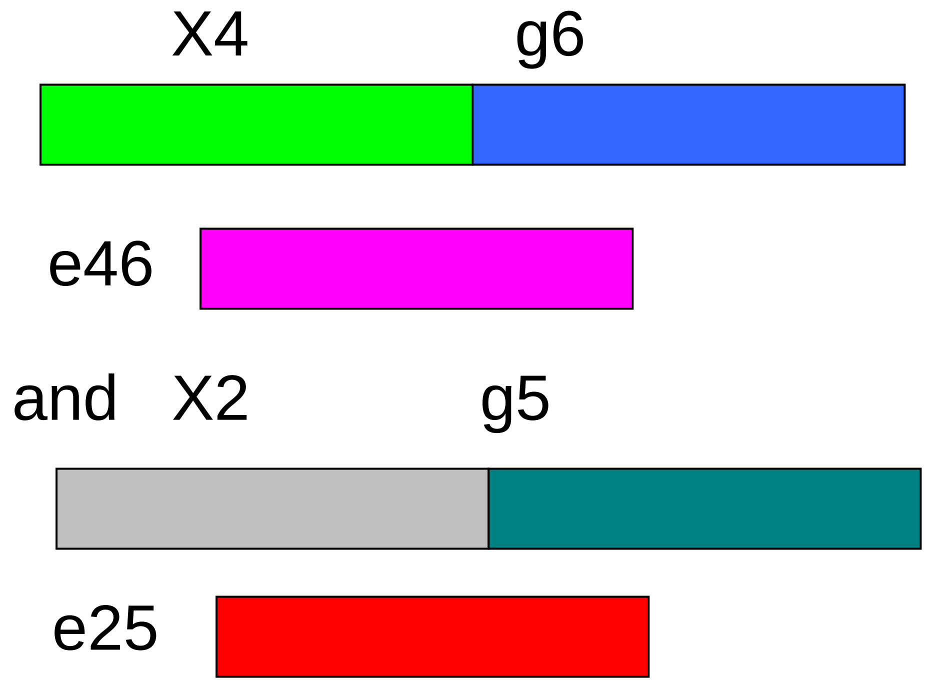
and



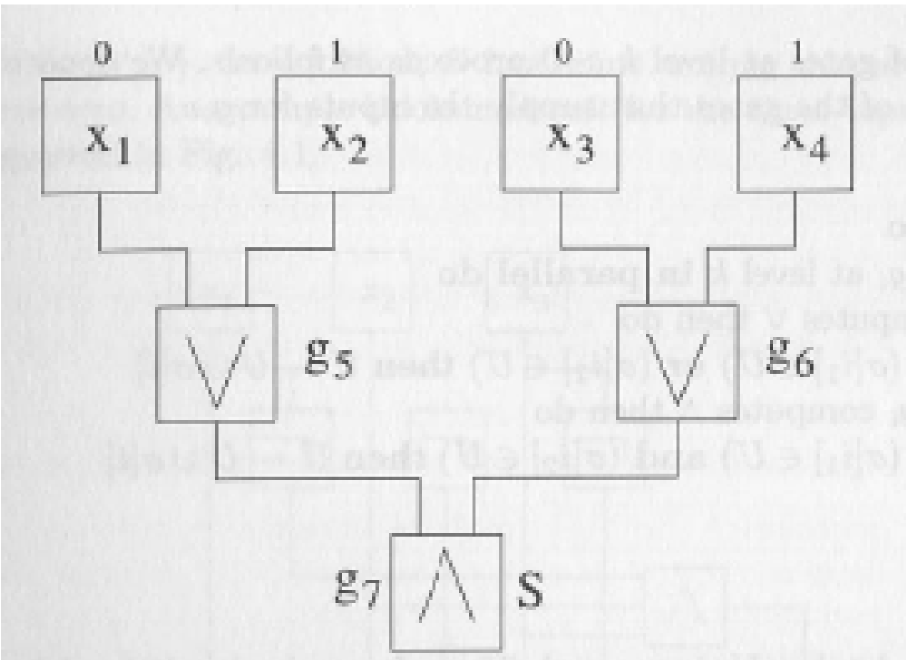
Continued



- By running the solution on a polyacrylamide gel, the strands are separated as:

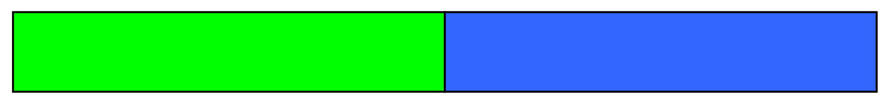


Continued



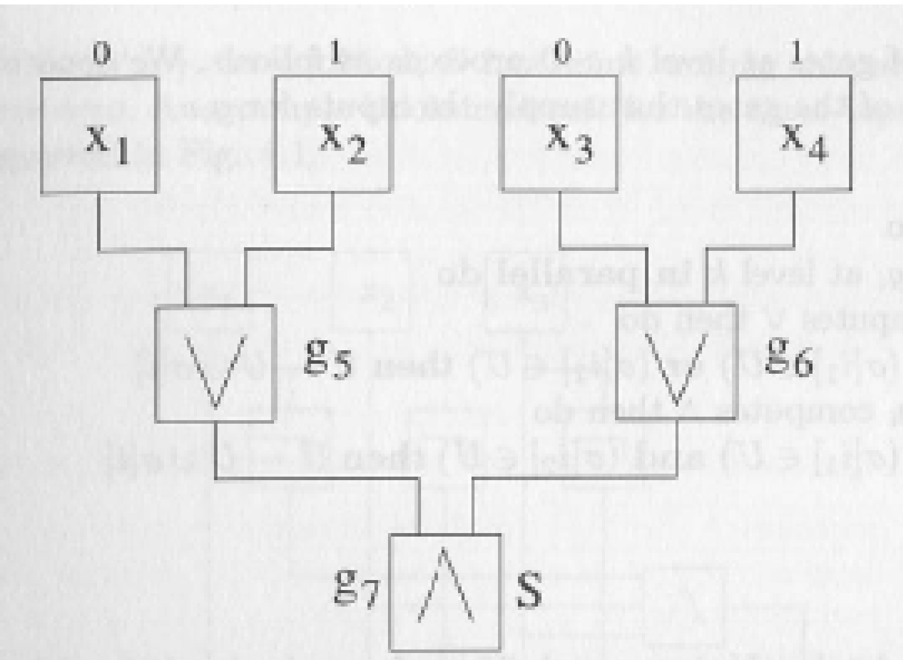
- Wash out the string which has length l, only length 2l strands left in the T1

X4 g6

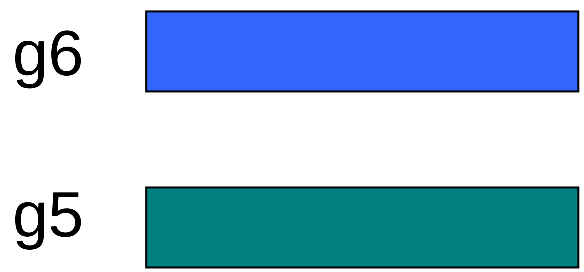


and X2 g5

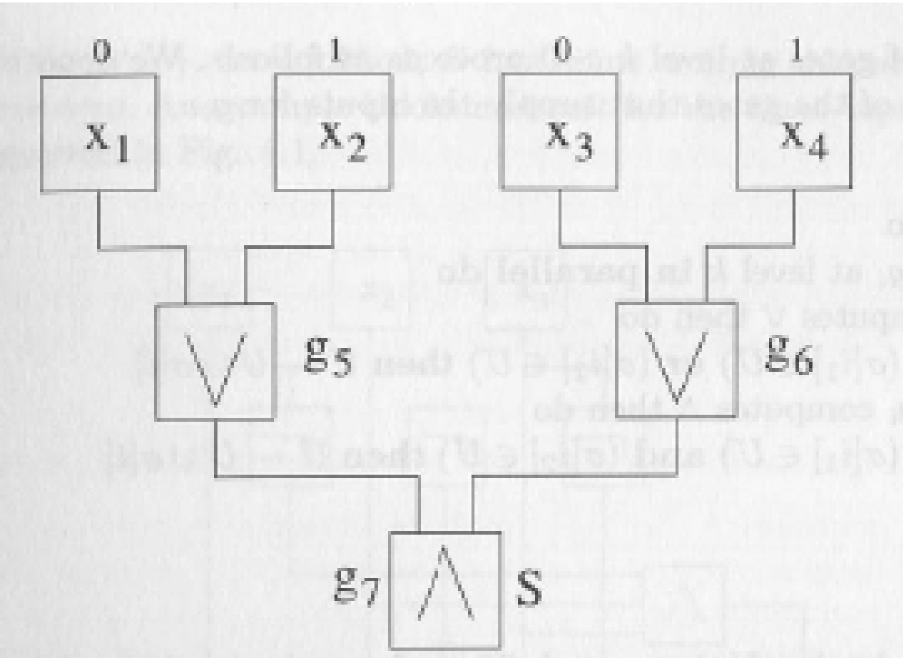
Continued



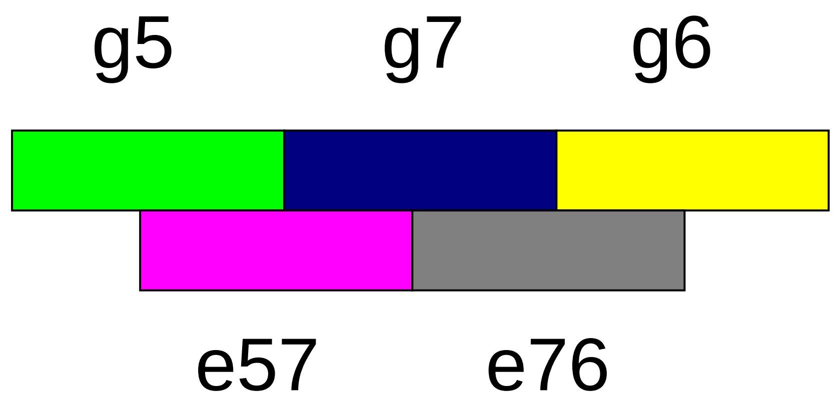
- These strands are then cut with a restriction enzyme recognizing sequence ε , only g5 and g6 left in T1



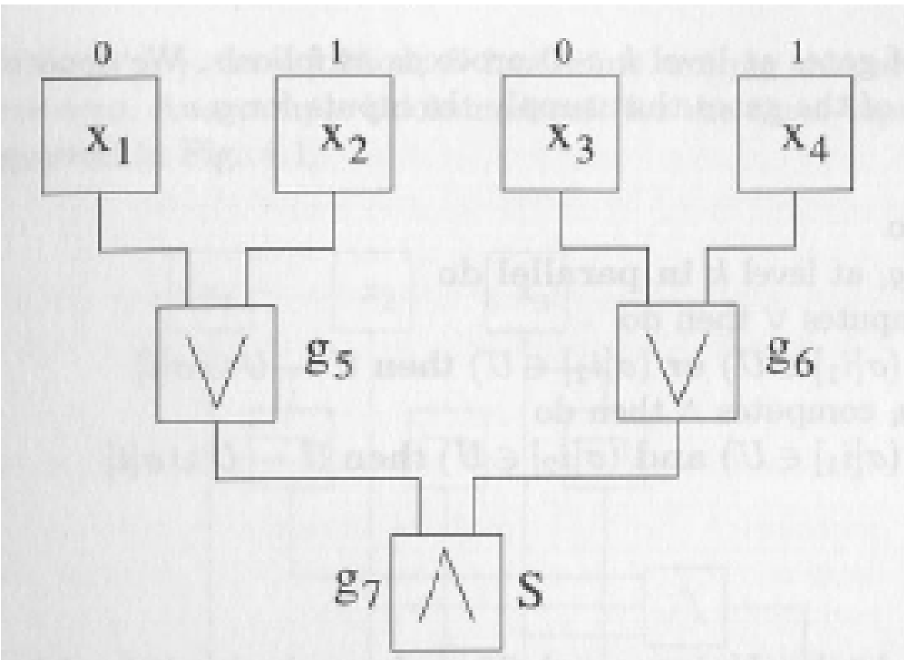
Continued

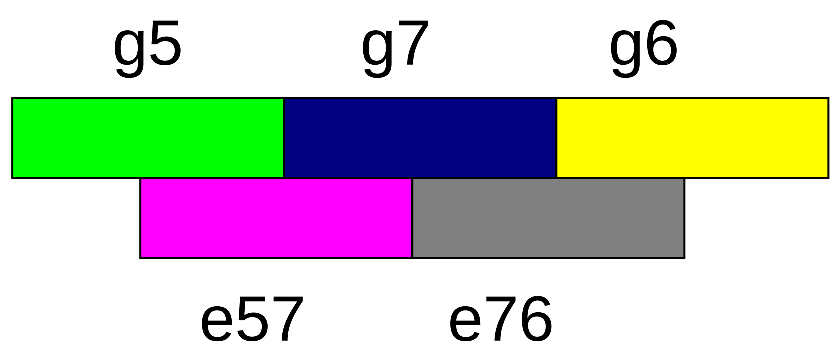


- Check level 2
- It's an AND gate
- Pour fluid from T1 to T2
- Same thing happens

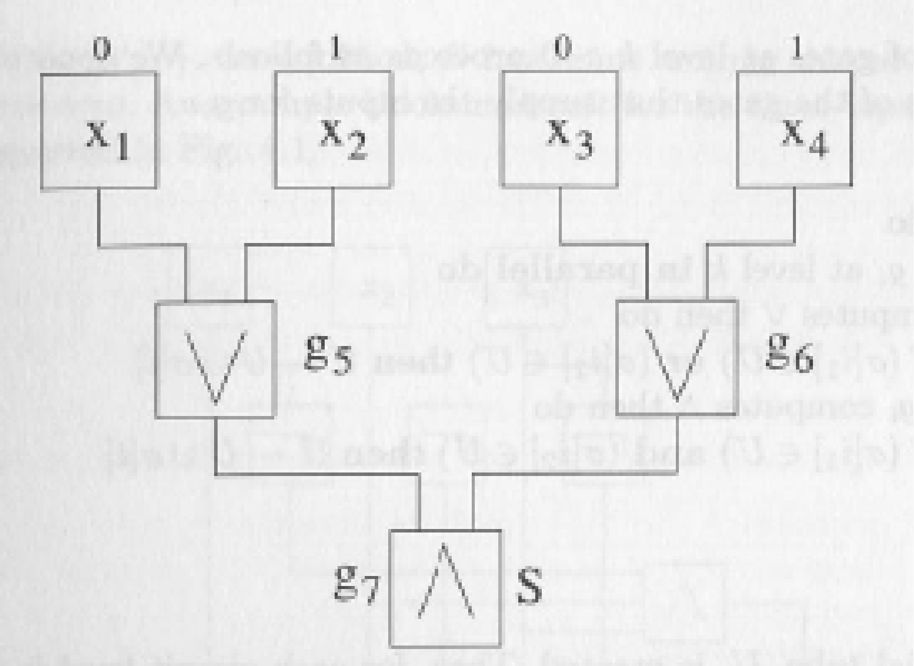


Final read-out



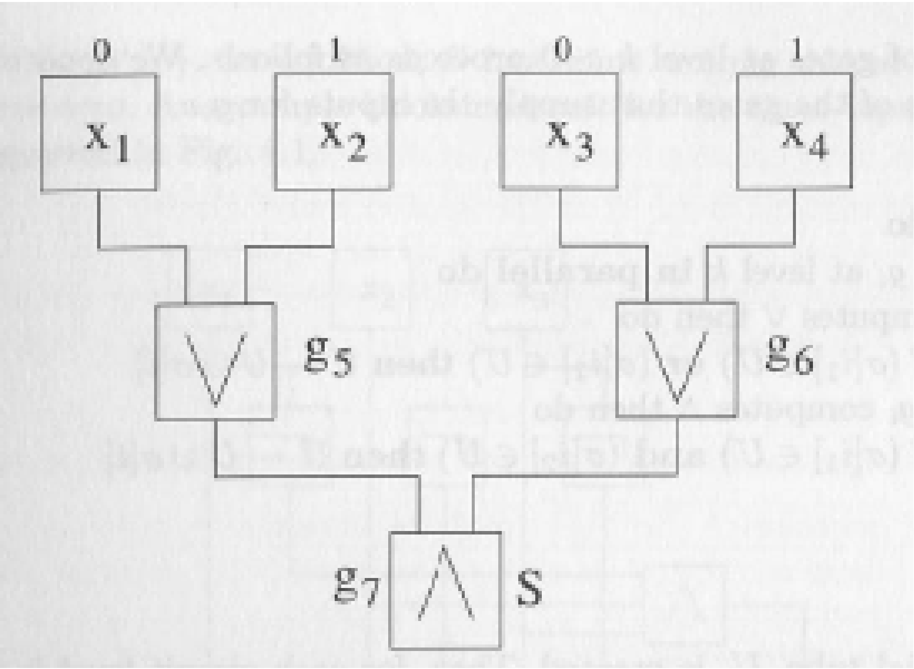
- Gate g7 is also the output gate, here we can find length 3l strands, that means this AND gate gives out “1”, otherwise gives out “0”.
- Therefore the output of this circuit is “1”
- 

Check correctness



- To OR gate, if in the tube has length 2l strands, it means this gate gives out "1"
- To AND gate, if in the tube has length 3l strands, it means this gate gives out "1"

Continued

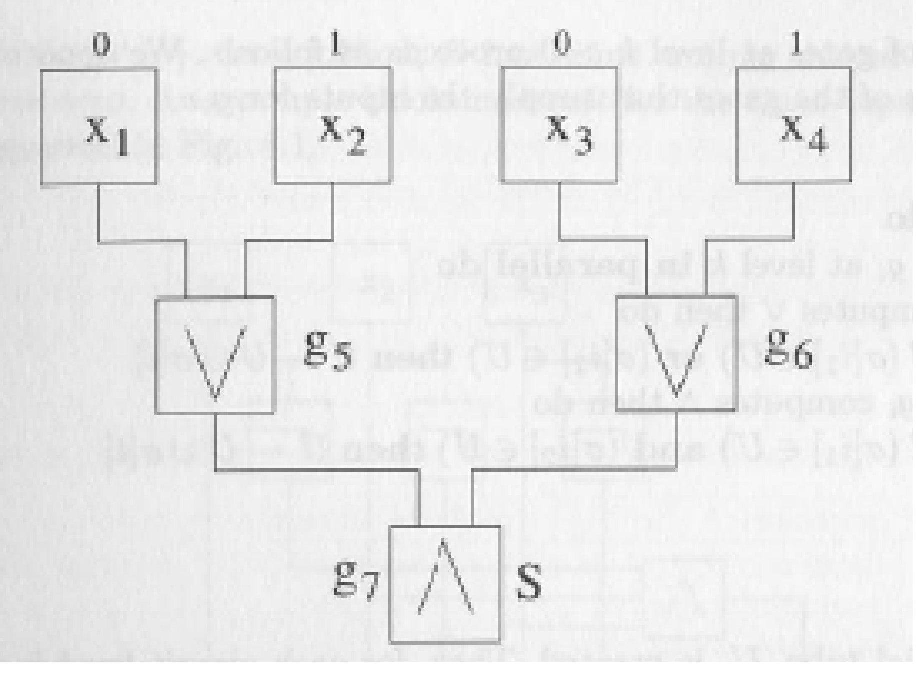


- Check g5:

  If X1=0 and X2=0, then we won't pour strands into the tube T0, therefore, in tube T1, it won't have length 2l string, g5 gives out "0"
- Check g6:

  same as g5

Continued



- Check g7:

  if g5 gives out “0”, and g6 gives out “1”, that means in tube T1, only g6 strand left. Therefore, after pouring T1 into T2, it won’t create length 3l string. Consequently, g7 gives out “0”.

Complexity

- Count the number of *pour* operation
- In each level k, there are three *pour* operations
  - Pour the fluid from Tk-1 into Tk
  - Pour the strands of each gate in level k into Tk
  - Pour the edge strands which connect gates in level k-1 and gates in level k into Tk

Continued

- Total number of *pour* is less than the summation of #gates and #edges
- Number of gates is size m
- Number of edges is $\geqslant$ 2m
  - As we know logic gate generally has at least 2 inputs
- Result: Complexity $\geqslant$ 3m

An important gate "NAND"

- Set up:

  1. A DNA strand with length l is assigned to any gate j in the level i denoted by g(i,j)

  Notice: Every strand corresponding to a gate starts and ends with a specific pattern as a restriction site.

Continued

2. for each intermediate gate g(i,j) with two inputs from gates g(i–1,p) and g(i–1,q) in (i– 1)th level, one strand with length 3l is also assigned and is called *link-strand*

Continued

- Link-strand:

  i) It's a 3l long strand

  ii) If g(i-1,p) = X, g(i-1,q) = Y, and g(i, j) = Z

  iii) Then the link-strand is $\overline{X}\ \overline{Y}\ \overline{Z}$

Continued

- For example, consider the strands

  z =5'–GGGAAGAGTCCC–3',

  x =5'–GGGTAGAAGCCC–3',

  y =5'–GGGTCTAGCCCC–3'

- Then the link strand is

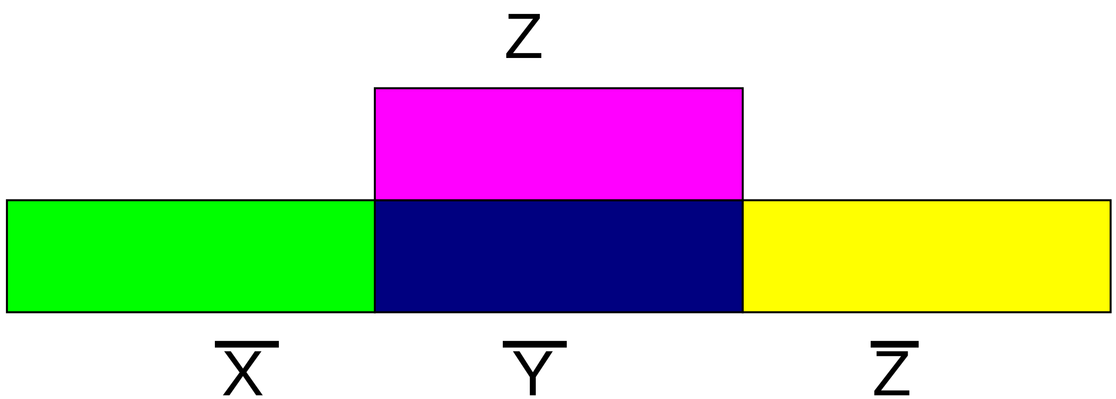
3'–CCCATCTTCGGGCCCTTCTCAGGGCCCAGATCGGGG–5'

Continued

- Now if we pour Z and link strand in to a test tube we get:

5’–GGGAAGAGTCCC–3’

3’–CCCATCTTCGGGCCCTTCTCAGGGCCCAGATCGGGG–5’

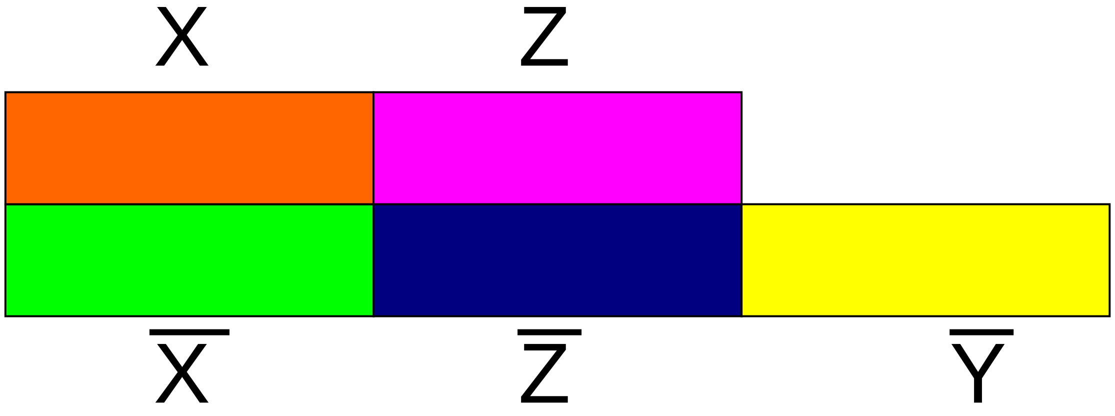


Continued

- If add strand x =5’−GGGTAGAAGCCC−3’ in the test tube the above strand is transformed to the following strand:

5’−GGGTAGAAGCCCGGGAAGAGTCCC−3’

3’−CCCATCTTCGGGCCCTTCTCAGGGCCCAGATCGGGG−5’



Continued

- If we add the restriction enzyme SmaI we get:

  5'-GGGTAGAAGCCC-3

  3'-CCCATC TTCGGG-5'

  And

5'-GGGAAGAGTCCC-3'

3'-CCC TTCTCAGGGCCCAGATCGGGG-5'

Level Simulation

- Tube T0 contains strands of length l each of which corresponds to only these input gates with value 1

- During the laboratory operations, the contents of Ti which corresponds to level i is added to the test tube Ti+1 of the i + 1th level.

Continued

- 1. Pour the contents of Tk−1 into Tk. The strands are annealed at the appropriate position by decreasing the temperature in the tube.
- 2. Add ligase enzyme to Tk in order for ligation between the double strands to occur.
- 3. All the complete double-stranded DNA sequences which show the zero value of output gates are eliminated from tube Tk by running on gelelectropherese.

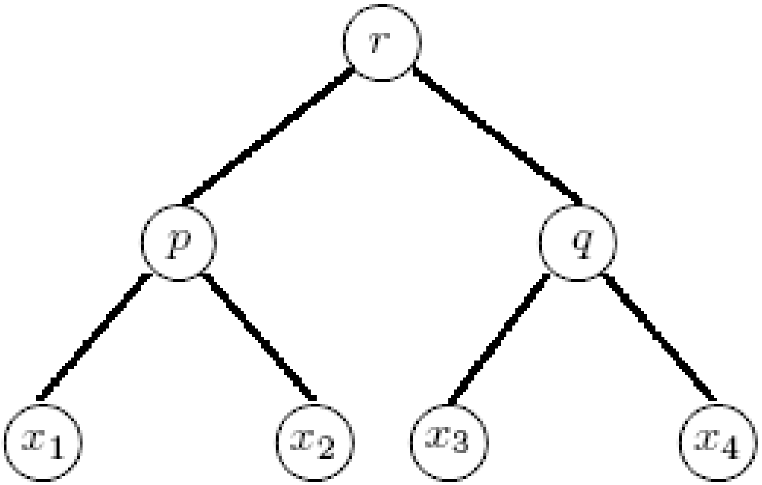
Continued

- 4. The incomplete DNA strands are cut by enzyme SmaI from the restriction site of this enzyme.
- 5. These strands are melted and the non-complement parts are kept. The other strands are ignored. This step can be performed by amplification of non-complement section of these strands by PCR.

Final read-out

- Eventually, after repeating the above operations for all the levels, if Td (the tube in last level) does not contain any complete double-stranded DNA, it can be induced that the final output for the circuits is one; otherwise is zero.

Example



x1=

5’-GGGGATTAACCC-3’,

x2=

5’-GGGAAATGTCCC-3’,

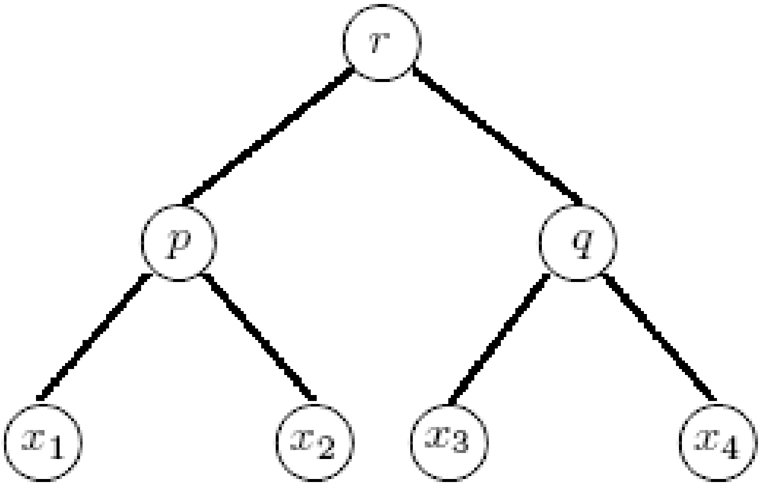
x3=

5’-GGGCAGCAGCCC-3’,

x4 =

5’- GGGTTTAGACCC-3’.

Continued



p=

5’-GGGTAGAAGCCC-3’,

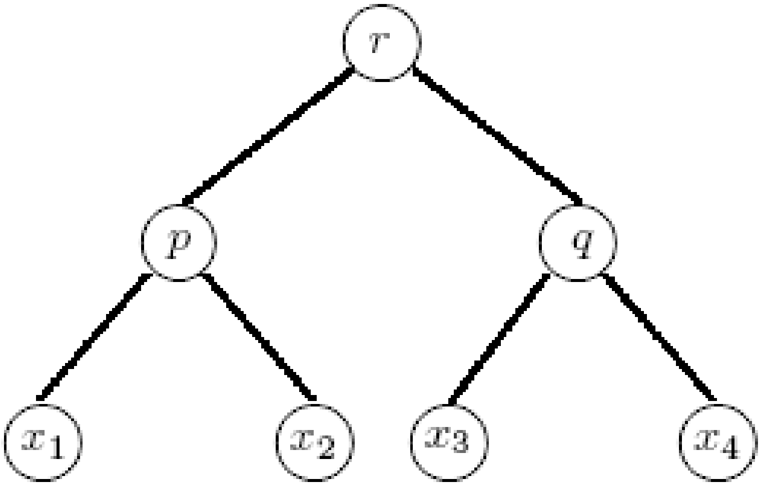
q=

5’-GGGTCTAGCCCC-3’,

r=

5’-GGGAAGAGTCCC-3’.

Continued



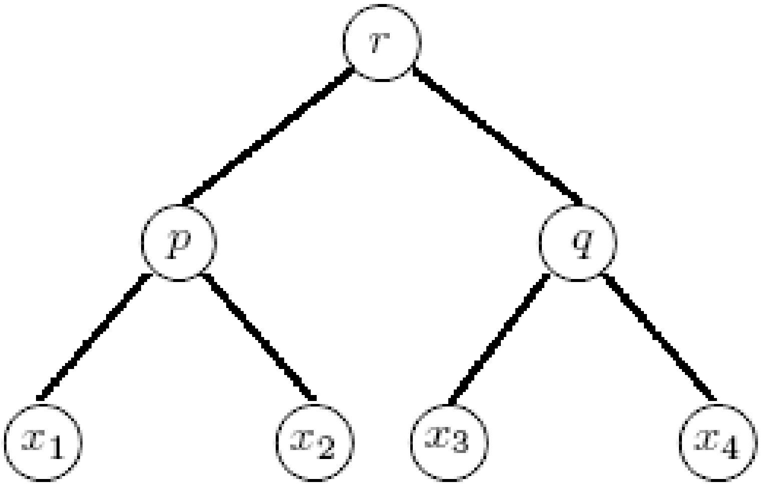
Set $x1 = 1$

$x2 = 1$

$x3 = 1$

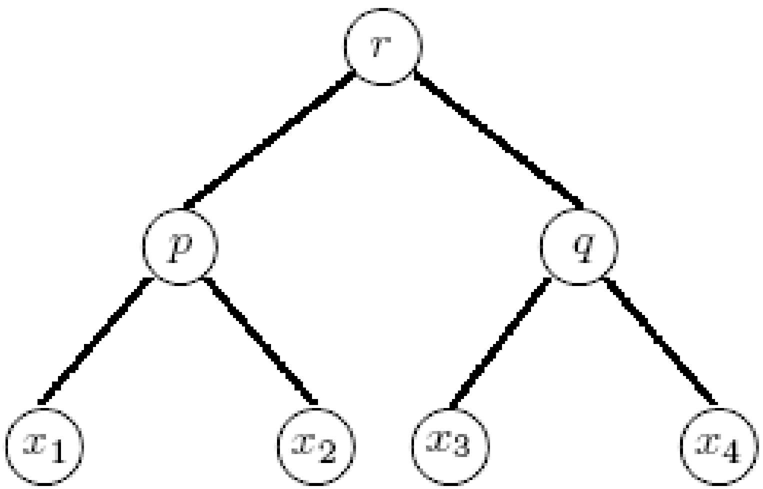
$x4 = 0$

Continued



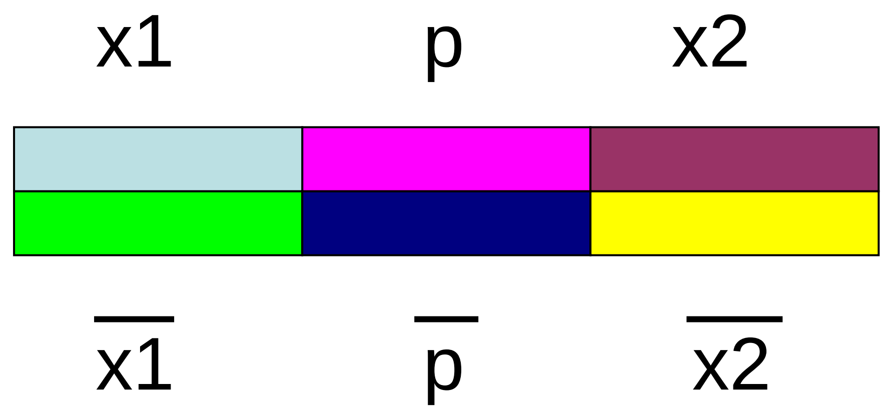
- In tube T0 has strands x1, x2, x3
- T1 contains x1, x2, x3, p, q, and link strands
- Pour T0 into T1

Continued



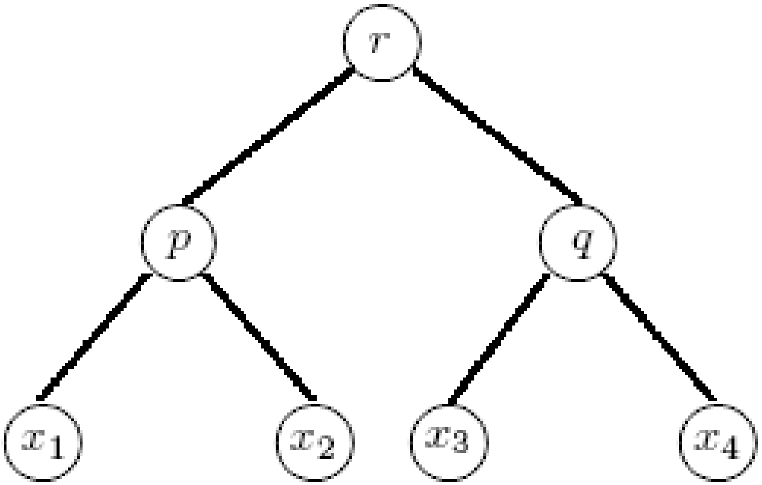
- After hybridization and ligation

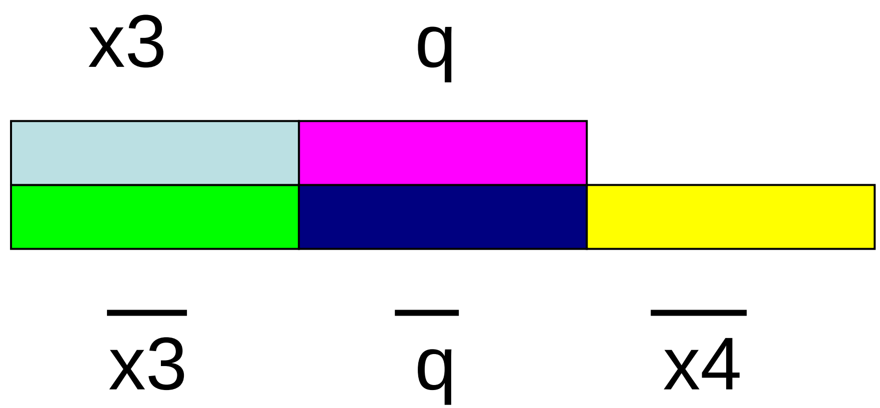
T1 contains:



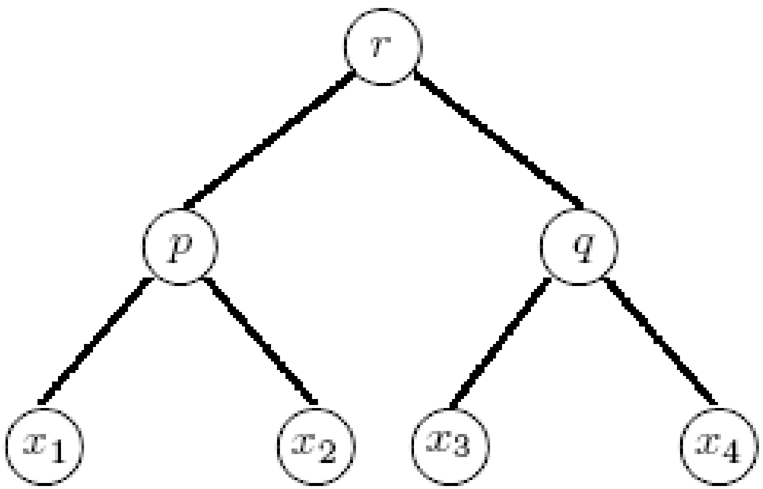
Continued

- And

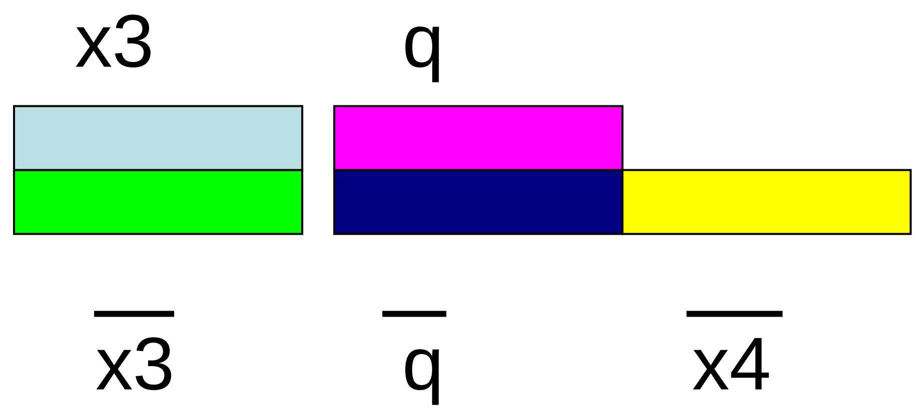




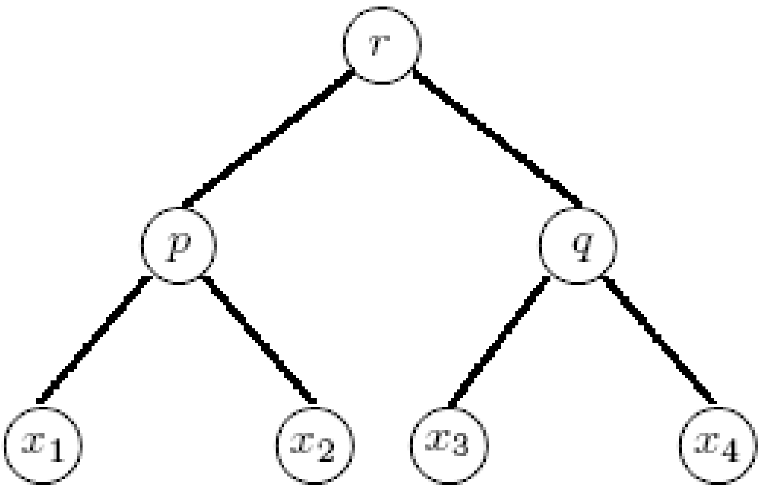
Continued



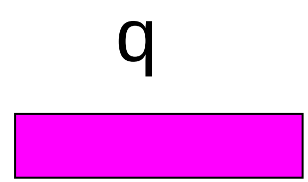
- After eliminate the complete double strand
- Adding enzyme SmaI into T1.



Continued

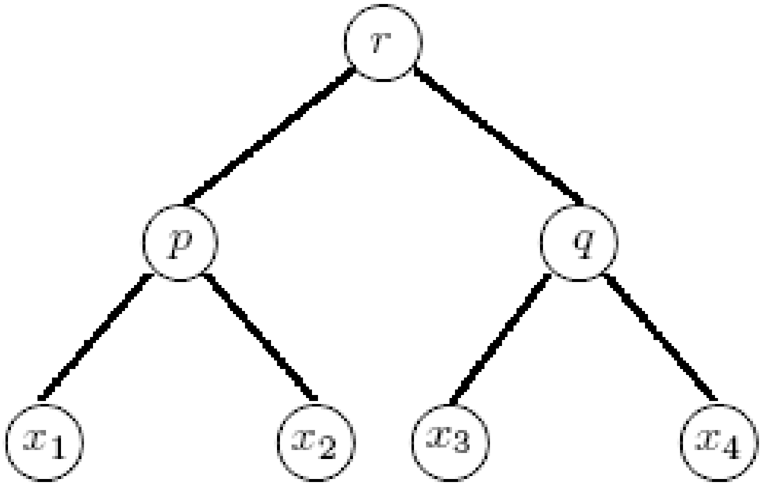


- After melting and ignoring the complement part, T1 only has

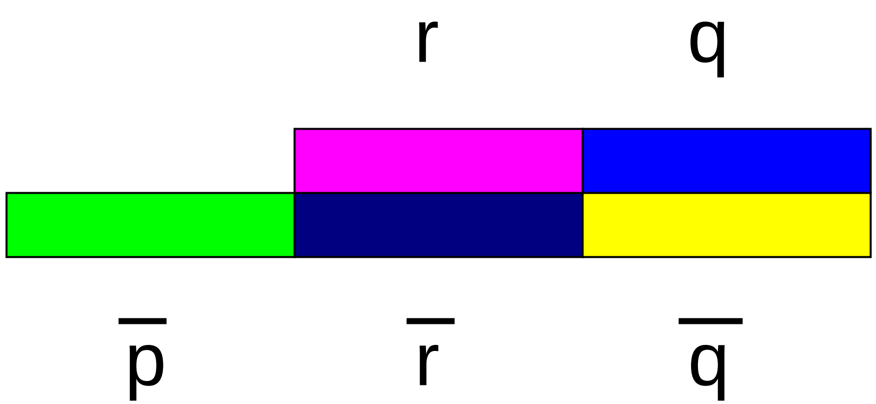


- Therefore, from T1 only q pour into T2

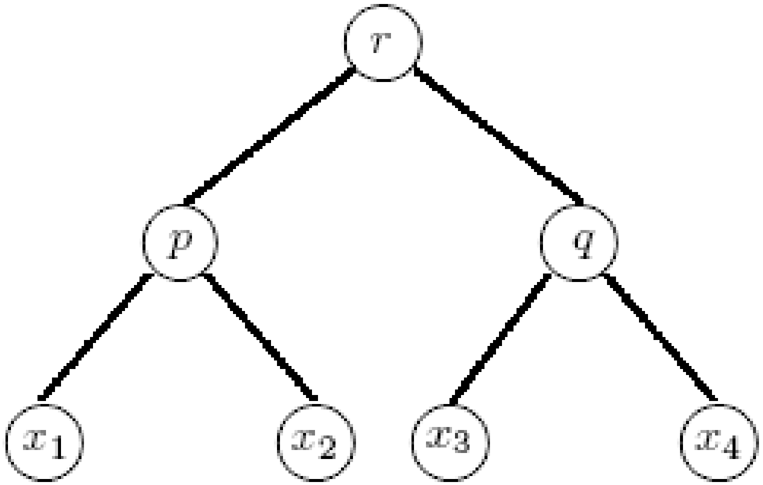
Continued



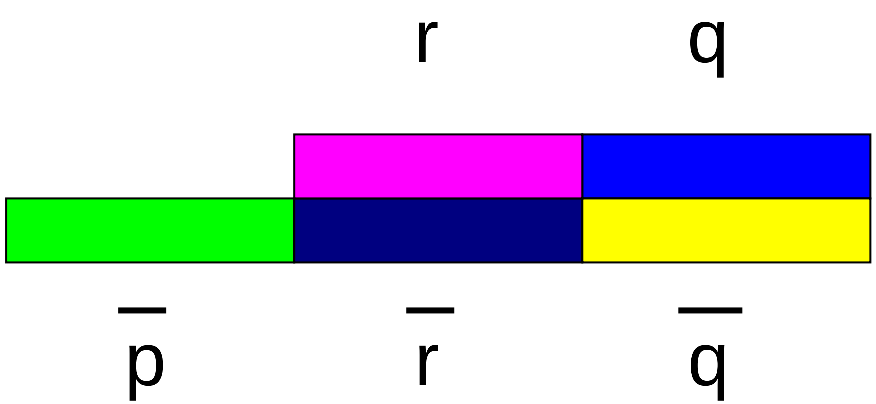
- After T1 pour into T2
- After hybridization and adding ligase enzyme T2 eventually contains



Continued



- Therefore this logic gate gives out 1



Question?

END