Designing XOR from NAND gates

Symbols to be used are

OR $\vee$ AND $\wedge$ NOR $\downarrow$ NAND $\uparrow$

XOR $\veebar$ XAND $\barwedge$

XNOR $\barwedge$ XNAND $\barwedge$

$\uparrow$ is called Sheffer's arrow.

$\downarrow$ is Goodhand's arrow.

with an underline to indicate X~

(One day I hope to persuade the world to abandon its myriad use of "illogical" representation and adopt a simple and "logical" system.)

Truth Tables are

p	**q**	**p∨q**
F	F	F
F	T	T
T	F	T
T	T	T

p	**q**	**p∧q**
F	F	F
F	T	F
T	F	F
T	T	T

p	**q**	**p↓q**
F	F	T
F	T	F
T	F	F
T	T	F

p	**q**	**p↑q**
F	F	T
F	T	T
T	F	T
T	T	F

p	**q**	**p⊻q**
F	F	F
F	T	T
T	F	T
T	T	F

p	**q**	**p⊼q**
F	F	T
F	T	F
T	F	F
T	T	T

p	**q**	**p⊽q**
F	F	T
F	T	F
T	F	F
T	T	T

p	**q**	**p⊺q**
F	F	F
F	T	
T	F	F
T	T	T

So we discover

XAND ≡ XNOR and

XNAND ≡ XOR

Next we feed p into both inputs of a NAND gate

p	p↑p
F	T
T	F

It is apparent both the NAND and NOR gates can act as an inverters. So if we link two NAND gates in series we get

p	q	p↑q	(p↑q) ↑ (p↑q)
F	F	T	F
F	T	T	F
T	F	T	F
T	T	F	T

we produce an AND gate as expected.

To make an OR gate we invert p and q and then feed the inverses into a NAND gate.

p	q	¬p	¬q	¬p ↑ ¬q
F	F	T	T	F
F	T	T	F	T
T	F	F	T	T
T	T	F	F	T

I term this a **NAND triple**.

To produce a NOR gate we just add an inverter at the final output.

So from a single NAND gate we can now make AND, NAND, OR and NOR gates.

Making XOR from NANDs

Now we can use the NAND gates (or NOR gates) to create the XOR gate.

Step 1

From a Venn diagram the XOR gate is represented by

(P ∩ Q') ∪ (P' ∩ Q)

which in logic becomes

(p ∧ ¬q) ∨ (¬p ∧ q)

Step 2

Draw up the three simplest logical gates to create

① is (p ∧ ¬q) inverter + AND

② is (¬p ∧ q) inverter + AND

③ is ① ∨ ② an OR gate

- a total of 5 gates.

Step 3

In ① replace AND with NAND + inverter. ① is now 3 gates

In ② replace AND with NAND + inverter. ② is now 3 gates

Replace ③ with a NAND triple.

We now have nine gates.

Step 4

As ¬¬p = p eliminate the two occurrences of two inverters in series. We now have 5 gates representing

p	q	¬p	¬q	p↑¬q	¬p↑q
F	F	T	T	T	T
F	T	T	F	T	F
T	F	F	T	F	T
T	T	F	F	T	T

And we feed these two outputs into the final NAND gate to produce

p	q	(p↑¬q) ↑ (¬p↑q)
F	F	F
F	T	T
T	F	T
T	T	F

which is indeed an XOR gate.

If we add an inverter to the output we produce an XNOR gate.

From our initial analysis we can also substitute for XAND and XNAND though these are rarely specified.

In fact there is a simpler solution using only 4 NAND gates as follows

p	q	p↑q	(p↑q)↑p	(p↑q)↑q
F	F	T	T	T
F	T	T	T	F
T	F	T	F	T
T	T	F	T	T

and these two outputs are fed into another NAND gate to produce

p	q	{ (p↑q)↑p } ↑ { (p↑q)↑q }
F	F	F
F	T	T
T	F	T
T	T	F

producing a XOR gate as before.

It was my lecturer's ambition to create an algorithm that would identify the minimal gate solution. Does it exist?

Robert Goodhand

The Apollo 11 moon lander had a guidance computer consisting of 4500 NOR gates.

Designing XOR from NAND gates

Symbols to be used are

OR $\vee$ AND $\wedge$ NOR $\downarrow$ NAND $\uparrow$

XOR $\underline{\vee}$ XAND $\underline{\wedge}$

XNOR $\underline{\downarrow}$ XNAND $\underline{\uparrow}$

$\uparrow$ is called Sheffer's arrow.

$\downarrow$ is Goodhand's arrow.

with an underline to indicate X~

(One day I hope to persuade the world to abandon its myriad use of "illogical" representation and adopt a simple and "logical" system.)

Truth Tables are

p	**q**	**p∨q**	**p**	**q**	**p∧q**
F	F	F	F	F	F
F	T	T	F	T	F
T	F	T	T	F	F
T	T	T	T	T	T

p	**q**	**p↓q**	**p**	**q**	**p↑q**
F	F	T	F	F	T
F	T	F	F	T	T
T	F	F	T	F	T
T	T	F	T	T	F

p	**q**	**p$\underline{\vee}$q**	**p**	**q**	**p$\underline{\wedge}$q**
F	F	F	F	F	T
F	T	T	F	T	F
T	F	T	T	F	F
T	T	F	T	T	T

p	**q**	**p$\underline{\downarrow}$q**	**p**	**q**	**p$\underline{\uparrow}$q**
F	F	T	F	F	F
F	T	F	F	T	T
T	F	F	T	F	F
T	T	T	T	T	T

So we discover

XAND ≡ XNOR and

XNAND ≡ XOR

Next we feed p into both inputs of a NAND gate

p	p↑p
F	T
T	F

It is apparent both the NAND and NOR gates can act as an inverters. So if we link two NAND gates in series we get

p	q	p↑q	(p↑q) ↑ (p↑q)
F	F	T	F
F	T	T	F
T	F	T	F
T	T	F	T

we produce an AND gate as expected.

To make an OR gate we invert p and q and then feed the inverses into a NAND gate.

p	q	¬p	¬q	¬p ↑ ¬q
F	F	T	T	F
F	T	T	F	T
T	F	F	T	T
T	T	F	F	T

I term this a **NAND triple**.

To produce a NOR gate we just add an inverter at the final output.

So from a single NAND gate we can now make AND, NAND, OR and NOR gates.

Making XOR from NANDs

Now we can use the NAND gates (or NOR gates) to create the XOR gate.

Step 1

From a Venn diagram the XOR gate is represented by

$(P \cap Q') \cup (P' \cap Q)$

which in logic becomes

$(p \wedge \neg q) \vee (\neg p \wedge q)$

Step 2

Draw up the three simplest logical gates to create

① is $(p \wedge \neg q)$ inverter + AND

② is $(\neg p \wedge q)$ inverter + AND

③ is ① $\vee$ ② an OR gate

- a total of 5 gates.

Step 3

In ① replace AND with NAND + inverter. ① is now 3 gates

In ② replace AND with NAND + inverter. ② is now 3 gates

Replace ③ with a NAND triple.

We now have nine gates.

Step 4

As $\neg\neg p = p$ eliminate the two occurrences of two inverters in series. We now have 5 gates representing

p	q	$\neg p$	$\neg q$	$p \uparrow \neg q$	$\neg p \uparrow q$
F	F	T	T	T	T
F	T	T	F	T	F
T	F	F	T	F	T
T	T	F	F	T	T

And we feed these two outputs into the final NAND gate to produce

p	q	$(p \uparrow \neg q) \uparrow (\neg p \uparrow q)$
F	F	F
F	T	T
T	F	T
T	T	F

which is indeed an XOR gate.

If we add an inverter to the output we produce an XNOR gate.

From our initial analysis we can also substitute for XAND and XNAND though these are rarely specified.

In fact there is a simpler solution using only 4 NAND gates as follows

p	q	$p \uparrow q$	$(p \uparrow q) \uparrow p$	$(p \uparrow q) \uparrow q$
F	F	T	T	T
F	T	T	T	F
T	F	T	F	T
T	T	F	T	T

and these two outputs are fed into another NAND gate to produce

p	q	$\{ (p \uparrow q) \uparrow p \} \uparrow \{ (p \uparrow q) \uparrow q \}$
F	F	F
F	T	T
T	F	T
T	T	F

producing a XOR gate as before.

It was my lecturer's ambition to create an algorithm that would identify the minimal gate solution. Does it exist?

Robert Goodhand

The Apollo 11 moon lander had a guidance computer consisting of 4500 NOR gates. An iphone contains over 9 billion logic gates.