XVIII. Logic and Bit Operations in Hardware

A. Each of the logical operations discussed previously (among others) can be implemented in computer hardware.

B. Since binary computers implement bit manipulations, the truth value ***true*** corresponds to the value $1B$ (the bit is turned on) and the truth value false corresponds to the value $0B$ (the bit is turned off).

C. Hardware Representation of Truth Values:

1. ***true*** corresponds to a switch that is turned on while ***false*** corresponds to a switch that is turned off.
2. ***true*** corresponds to a voltage of $+5$ volts while ***false*** corresponds to a voltage of -5 volts.
3. ***true*** corresponds to a binary 1, or one bit turned on, while ***false*** corresponds to a binary 0, or one bit turned off.

D. Bit Operations

		Disjunction	Conjunction	Exclusive Or
x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

E. Bit String Operations:

1. Normally these operations are accept strings of bits as operands rather than single bits.

2. Example: $219_{10} = 110011011_2$

$$= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0$$

$$= 64 + 16 + 8 + 2 + 1 = 219$$

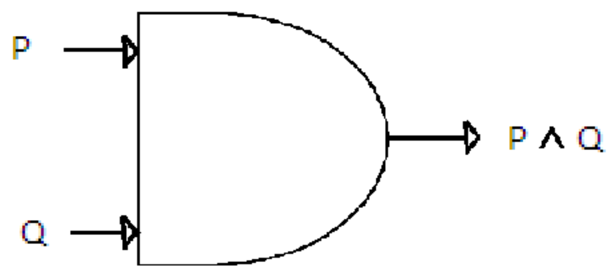
$$125_{10} = 001111101_2$$

3. Then: $219 \vee 125 = 110011011_2 \vee 001111101_2$
$= 11111111_2 = 255_{10}$

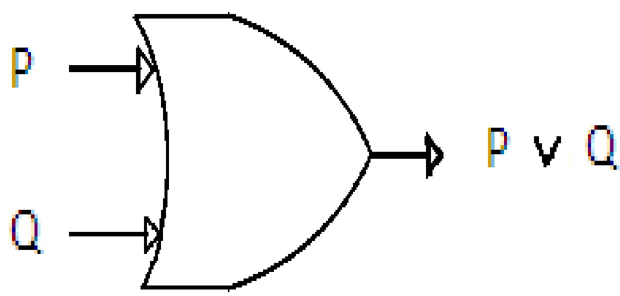
4. And: $219 \wedge 125 = 110011011_2 \wedge 001111101_2$
$= 000011001_2 = 25_{10}$

XIX. BitWise Compound Propositions Implemented in Hardware

A. The hardware symbol for the conjunction of P and Q is:



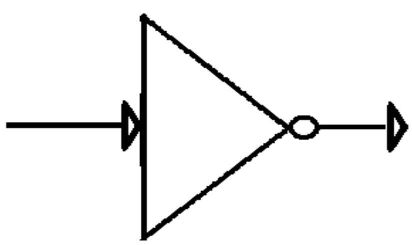
B. The hardware symbol for the disjunction of P and Q is:



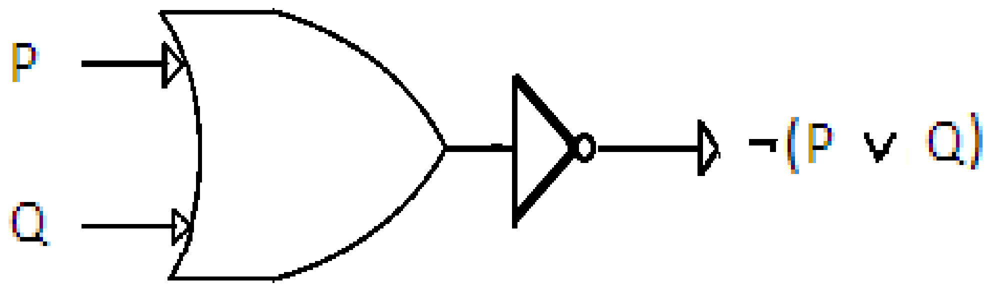
C. In both of the above the binary input for P, either $'1'B$ (true) or $'0'B$ (false), enters via the upper arrow (an electrical connection) while the the binary input for Q, either $'1'B$ (true) or $'0'B$ (false), enters via the upper arrow.

D. The output for either $P \wedge Q$ or $P \vee Q$, either $'1'B$ (true) or $'0'B$ (false) emerges from the right-most arrow.

H. The hardware symbol for the negation of P, or $\neg P$, is:



so $\neg(P \vee Q)$ would be represented in hardware as:

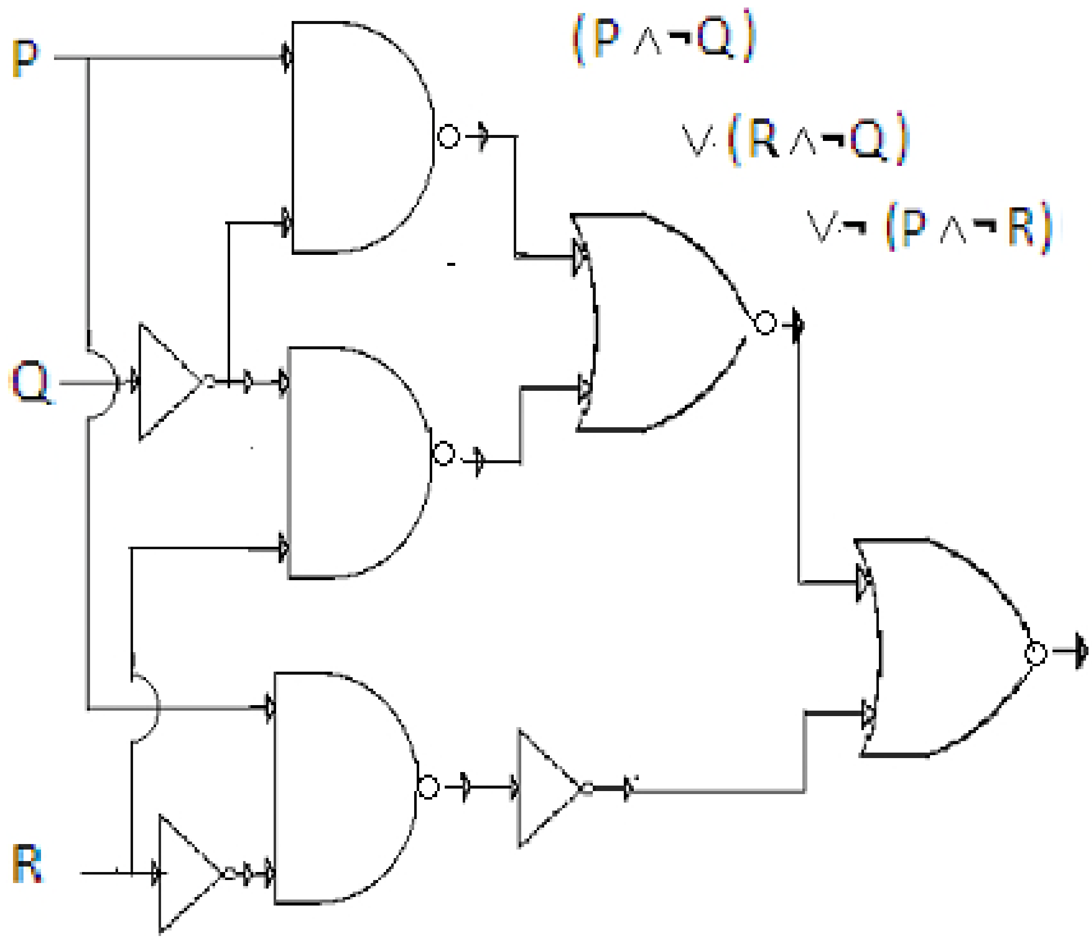


XIX. Representing Compound Logical Expressions in Hardware

A. Consider the logical expression:

$$(P \wedge \neg Q) \vee (R \wedge \neg Q) \vee \neg(P \wedge \neg R)$$

B. This expression can be represented in hardware as shown below:



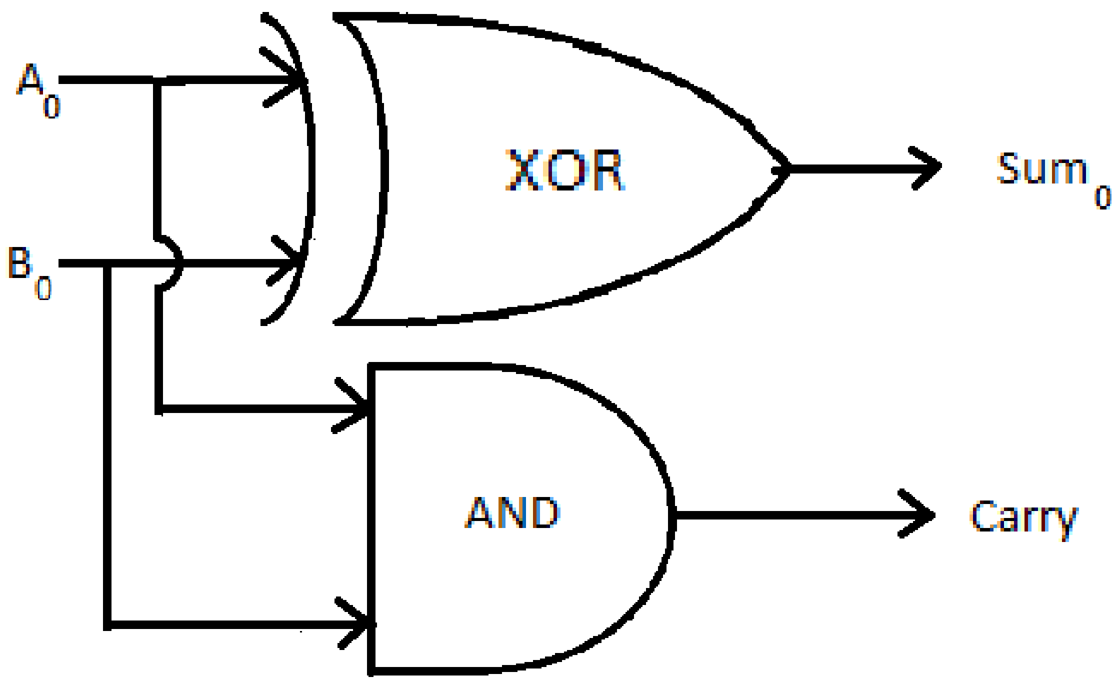
C. The left-most column of hardware gates represents, from top-to-bottom:
1. $P \wedge \neg Q$
2. $R \wedge \neg Q$
3. $P \wedge \neg R$

D. The center column of hardware gates represents, from top-to-bottom:
1. $(P \wedge \neg Q) \vee (R \wedge \neg Q)$
2. $\neg(P \wedge \neg R)$

E. The right-most column, a single gate, represents:
$((P \wedge \neg Q) \vee (R \wedge \neg Q)) \vee \neg(P \wedge \neg R)$

XX. Binary Addition Implemented in Hardware

A. Circuit



B. Bit Addition Definition

A	B	Sum	$Carry$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0