Circuits and Proposition

Variable:	inputs to a circuit
Proposition:	output of a circuit
Literals:	variables and their negations
Term:	a literal or the disjunction or conjunction of a finite number of literals

Algorithm 1: To write the disjunctive form for a circuit with inputs $a_1, a_2, \ldots, a_n$ and output *f*.

1. If $f = 0$ for all assignments, the form is $a_1 \wedge a_1'$; if $f = 1$ for all assignments, the form is $a_1 \vee a_1'$. Otherwise, proceed.
2. For each assignment with $f = 1$ write
   $a_1^{e1} \wedge a_2^{e2} \wedge \ldots a_n^{en}$
   where a_i^{ei} is a_i if $a_i = 1$ and a_i^{ei} is a_i' if $a_i = 0$.
3. If only one term has been written, it is the form. Otherwise, the form is the disjunction of all terms written.

Principle of Substitution

In an equivalence any variable may be replaced by a proposition and any proposition may be replaced by an equivalent proposition.

Idempotent Laws	$x \wedge x \equiv x$
	$x \vee x \equiv x$
Commutative Laws	$x \wedge y \equiv y \wedge x$
	$x \vee y \equiv y \vee x$
Associative Laws	$x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
	$x \vee (y \vee z) \equiv (x \vee y) \vee z$
Distributive Laws	$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
	$x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$
Absorption Laws	$x \wedge (x \vee y) \equiv x$
	$x \vee (x \wedge y) \equiv x$
Bound Laws	$x \wedge 0 \equiv 0$
	$x \vee 1 \equiv 1$
	$x \vee 0 \equiv x$
	$x \wedge 1 \equiv x$
Complement Laws	$x \wedge x' \equiv 0$
	$x \vee x' \equiv 1$
Involution Laws	$(x')' \equiv x$
DeMorgan's Laws	$(x \wedge y)' \equiv x' \vee y'$
	$(x \vee y)' \equiv x' \wedge y'$

Algorithm 2 To write a Boolean expression in short disjunctive form.

1. Apply the DeMorgan and involution laws as many times as possible.
2. Apply the general first distributive law as many times as possible. The proposition is now in disjunctive form (d.f.)
3. Apply the idempotent, bound, and complement laws as many times as possible.
4. Apply the second absorption law as many times as possible. The proposition is now in short disjunctive form (s.d.f.).

Tautology: a proposition that has truth value 1 for all values of its component statements.
Contradiction: a proposition with truth value 0 for all of its component statements.
Implication: $x \Rightarrow y \equiv x' \vee y$

$x \Rightarrow (x \vee y)$ is a tautology

$x \wedge y \Rightarrow x$ is a tautology

$x \Rightarrow y$ could be written $y \Leftarrow x$, hence $(x \Rightarrow y) \wedge (x \Leftarrow y)$ is written $x \Leftrightarrow y$

Principle of Duality

Given any true statement of equivalence, the dual statement, obtained by switching $\vee$ and $\wedge$, and 0 and 1, is also true.

Minimal Form

A minimal form of a proposition is a disjunctive form such that
(1) no equivalent d.f. has fewer terms
(2) no equivalent d.f. with the same number of terms has fewer literals (counting repetitions).

Consensus Law

$$(x \wedge z) \vee (y \wedge z') \equiv (x \wedge z) \vee (y \wedge z') \vee (x \wedge y)$$

consensus literal: is the literal that is the reason for the consensus.
consensus term: is the conjunction of the two terms without the consensus literal and its negation.

Algorithm 3.3 To find the consensus expansion of a proposition *f* in short disjunctive form.

1. If there are two terms that have a consensus term that does not contain a term already in the form, then choose two such terms and add their consensus term to the form. Otherwise, go to 3.
2. Delete every previous term that contains the term just added; then go to 1.
3. STOP. You have the consensus expansion of *f*, that is, the disjunction of all terms that could occur in a minimal form of *f*.

Minterm: a term in a disjunctive form of a proposition if for each variable in the proposition, either that variable or its negation occurs in the term.

Algorithm 3.4 Selection of all minimal forms from the consensus expansion.

1. For each term in the consensus expansion, write the base ten codes of its minterm expansion.
2. Make a table with rows headed by the terms of the consensus expansion and columns headed by the base ten codes that appear in at least one minterm expansion. Place X's to indicate the minterm expansions of each term.
3. Circle each X that has no other X in its column. Then star each term in the consensus expansion that has a circle in its row. These are the essential terms of the original proposition. Then circle each column head that has an X in a starred row. If there are no uncircled column heads, then the only minimal form is the disjunction of the essential terms. Otherwise, proceed.
4. For each uncircled column head, write the disjunction of the terms heading row with X in that column. Then write the conjunction of the terms you have written. This proposition is called the Petrick proposition of the original proposition.
5. Rewrite the Petrick proposition in short disjunctive form; call it Π.
6. Retain only those terms of Π that involve the smallest number of minterms.
7. For each retained term of Π, count the number of literals (including repetitions) that occur in the terms of the consensus expansion that appear in that term of Π. Omit all retained terms of Π except those for which this count is smallest.
8. For each term of Π that still remains, the terms of the consensus expansion that appear in that term of Π, together with all essential terms found in Step 3, are the terms of a minimal form of the original proposition. Moreover, every minimal form is obtained in this way.