

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/3615962>

An algorithm for automated printed circuit board layout and routing evaluation

Conference Paper · September 1993

DOI: 10.1109/ISEMC.1993.473720 · Source: IEEE Xplore

CITATIONS

3

READS

27

5 authors, including:



[Todd H. Hubing](#)

Clemson University

313 PUBLICATIONS 3,626 CITATIONS

[SEE PROFILE](#)



[Thomas Van Doren](#)

Missouri University of Science and Technology

103 PUBLICATIONS 2,367 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



material chemistry [View project](#)

AN ALGORITHM FOR AUTOMATED PRINTED CIRCUIT BOARD LAYOUT AND ROUTING EVALUATION

T. Hubing, P. Grover, T. Van Doren, J. Drewniak, and L. Hill
Department of Electrical Engineering
University of Missouri-Rolla
Rolla, Missouri 65401

Abstract

An algorithm has been developed to evaluate printed circuit boards that are designed using automated board layout and routing software. The algorithm analyzes aspects of component placement and trace routing while searching for violations of basic EMC design principles. The algorithm is implemented in a code designed to work with a widely used board layout and routing program. This code can help novice and experienced circuit board designers to avoid mistakes that may result in serious electromagnetic compatibility problems.

Introduction

Automated printed circuit board layout and routing software is widely used in the electronics industry to develop complex multilayer printed circuit board designs. Unfortunately, board layout and routing software does not enforce basic EMC design rules and procedures. Unless the user of this software is familiar with good design practices, boards designed in this manner may have serious EMC problems.

One advantage of using automated board layout tools is that a complete description of all aspects of the board design is stored in computer files. Various design evaluation utilities can access the information in these files and evaluate the board design without requiring the user to provide this information manually.

This paper describes a computer code, called EMlcheck, that was developed to perform an EMC evaluation of printed circuit boards laid out using the Mentor Graphics Boardstation software. The computer code analyzes aspects of the component placement and trace routing while searching for violations of basic EMC design rules. This easy-to-use code helps novice board designers avoid layout and routing errors that can adversely affect the electromagnetic compatibility of the design. Although this paper describes a specific implementation of a rule checking algorithm, the techniques employed are very general and can be adapted to work with most board layout and routing software.

EMlcheck is not a numerical modeling code nor does it employ numerical electromagnetic modeling techniques. It is a rule checker intended to be used by board designers or board design reviewers who may have little or no knowledge of electromagnetics or EMC. The program will run from beginning to end without any user supplied input.

EMlcheck scans the entire design as it is stored in the board layout description files looking for violations of basic EMC design guidelines. Each violation encountered is ranked and written to an output file. Violations can be displayed along with simple recommendations for correcting any problems found.

Algorithm Structure

The flowchart in Figure 1 provides an overview of the algorithm. The first step in evaluating printed circuit board designs is to assemble all the relevant information. EMlcheck scans the Mentor Graphics board description files to determine the names and location of every net¹ and component on the board. It then classifies each net based upon the devices that the net connects. Classification is necessary because the design rules pertaining to a particular net will depend on the type of

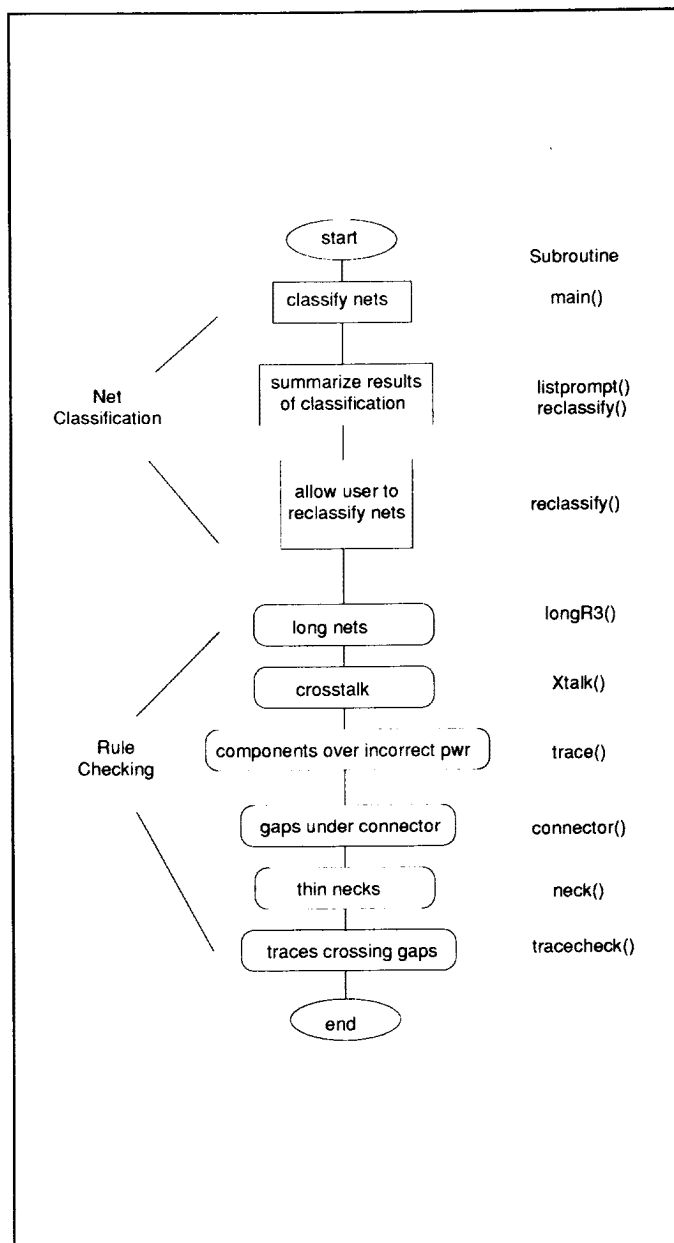


Figure 1: The EMlcheck algorithm

signal the net is likely to carry. For example, a trace that travels the full length of the board may not pose a radiation problem if it carries a low-power, low-frequency signal; but it may be a significant cause for concern if it is a high-speed clock trace. Rule checking software must make decisions about the type of signal likely to be found on each net based on the components that are connected to the net. Nets are classified according to their radiation potential, susceptibility, and D.C.

1 A net is a circuit board trace that connects two or more components. It is the physical equivalent of a node in a schematic diagram.

power reference. Nets that carry balanced signals and nets that are not confined to the card are also identified. A feature of this algorithm that allows it to be particularly intelligent in the way that nets are classified, is that it employs a *look-up table* that contains EMC-related information about components. Each pin of every component listed in the look-up table is assigned parameters that help determine the classification of the attached net.

After the program has classified each net, it gives the user an opportunity to change the classification of any net. This is important because often the board designer will be aware of information affecting the classification of a net that is not available to the rule checking software. Users unfamiliar with the board design or with the rule checking code can simply elect to use the default classifications.

Once all of the nets on the board have been classified, the code begins checking the design against a set of specific design rules. Each design rule is evaluated using a separate, independent subroutine. Rule checking subroutines read the necessary board geometry and net classification information, and each subroutine returns a ranked list of rule violations. The list of violations is stored in an output file and the user is given the option of viewing some or all of these violations.

Net Classification

Net classification is a critical step in the board layout evaluation process. Many of the design guidelines require a knowledge of the type of signal likely to be found on a net. Without a reasonably accurate evaluation of this signal, the value of some guidelines is greatly diminished. There are five variables assigned to each net for characterizing the type of signal the net is likely to carry:

	<u>Designation</u>	<u>Value</u>
1. A radiation potential	R1, R2, or R3	1,2, or 3
2. A susceptibility potential	S1, S2, or S3	1,2, or 3
3. The power bus associated with the signal	P5, P15, or P28	5,15, or 28
4. Is the signal balanced	UNB or BAL	0 or 1
5. Does the signal go off the card	YES or NO	1 or 0

The radiation potential is a measure of the ability of a signal to cause a radiation problem. **R3** nets are most likely to be a source of radiation (e.g., clock lines, high-speed data lines). **R1** nets are least likely to radiate (e.g., control or reset lines). The susceptibility potential is a measure of the ease with which a signal on a net may be corrupted by external noise. **S3** nets are the most susceptible (e.g., traces attached to the input of a high-gain amplifier). **S1** nets are the most immune to interference (e.g., low-speed digital lines). The power bus associated with a particular net is recorded so that the EMCheck program can ensure that all components associated with a particular net derive their power from the same source. Any signals passed between components on different power buses must use components designed for this purpose (e.g., A/D converters). Balanced signals are exempt from some design guidelines. For example, traces carrying a balanced signal may cross over a gap in the ground plane. Therefore, any nets that carry a balanced signal must be identified during the net classification process. Finally, all nets attached to a connector pin are labeled I/O (Input/Output) nets. These nets are the link between the board and the external world. Since unwanted interference generally enters and exits the board through these traces, crosstalk between these nets and other nets on the board is a primary concern.

The look-up table

The same five variables that the EMCheck code uses to classify nets are assigned to each pin of every component listed in the look-up table. A portion of a typical look-up table is illustrated in Figure 2. Each line contains a component name, pin number, part number, and values for the five net classification variables. Typically the pins of passive components such as resistors and capacitors are classified as

R1,S1,?,?,?. This simply means there is not enough information to draw conclusions about the radiation, susceptibility, power bus, balance, or I/O status of traces connected to this component. On the other hand, the clock input pin to an HCMOS logic module would typically be classified **R3,S2,P5,UNB,NO**. This indicates that signals on traces connected to this pin are likely to be a radiation source, moderately susceptible, associated with the 5 volt power bus, unbalanced, and should not be associated with I/O.

```

CAPACITOR 1 200-15050 1 1 ? ? ?
CAPACITOR 2 200-15050 1 1 ? ? ?
CAPACITOR 1 200-15050 1 1 ? ? ?
.
CAP.POLAR 1 250-16020 1 1 5 ? ?
CAP.POLAR 2 250-16020 1 1 5 ? ?
RESISTOR 1 100-11052 1 1 ? ? ?
RESISTOR 2 100-11052 1 1 ? ? ?
CONN 1 550-14032 1 1 ? ? ? 1
CONN 2 550-14032 1 1 ? ? ? 1
CONN 3 550-14032 1 1 ? ? ? 1
CONN 4 550-14032 1 1 ? ? ? 1
CONN 5 550-14032 1 1 ? ? ? 1
CONN 6 550-14032 1 1 ? ? ? 1
.
.
FERR_BEAD_LEADED 1 980-00022 1 1 ? ? ?
FERR_BEAD_LEADED 2 980-00022 1 1 ? ? ?
74HC74 1 910-10074 3 1 1 5 0 ?
74HC74 2 910-10074 3 1 1 5 0 ?
74HC74 3 910-10074 3 1 1 5 0 ?
CRYSTAL 1 918-40012 3 1 5 1 0
CRYSTAL 2 918-40012 3 1 5 1 0
.
800-P4647 1 800-P4647 1 2 5 0 ?
800-P4647 2 800-P4647 3 2 5 0 ?
800-P4647 3 800-P4647 1 2 5 0 ?
800-P4647 4 800-P4647 1 2 5 0 ?
900-14647-001 7 900-14647-001 1 2 5 0 ?
900-14647-001 8 900-14647-001 1 2 5 0 ?
.
.
85C30 41 900-12471-405 1 1 5 0 ?
85C30 42 900-12471-405 1 1 5 0 ?
85C30 43 900-12471-405 1 1 5 0 ?
85C30 44 900-12471-405 1 1 5 0 ?
691 1 900-12471-410 1 1 5 0 ?
691 2 900-12471-410 1 1 5 0 ?
691 3 900-12471-410 1 1 5 0 ?
691 4 900-12471-410 1 1 5 0 ?
691 5 900-12471-410 1 1 5 0 ?
691 6 900-12471-410 1 1 5 0 ?

```

Figure 2: The lookup table

The following guidelines are applied when classifying component pins:

- An **R3** classification is assigned to all clock inputs and data outputs when the data is likely to be high speed continuous data.
- An **R2** classification is assigned to all digital logic outputs that are not **R3**. This includes the output pins of generic digital components such as NAND gates or inverters.
- An **R1** classification is assigned to any pins that are neither **R3** nor **R2**.
- An **S3** classification is assigned to any component pins that are likely to be highly sensitive to noise. This includes inputs to operational amplifiers, optical receivers, analog comparators, and precision A/D converters. Inputs that are not particularly sensitive, but which may be especially important to the proper operation of the system, such as the reset input on a microprocessor are also classified as **S3**.
- An **S2** classification is assigned to input pins on active devices that are not classified as **S3**.
- An **S1** classification is assigned to any pin that is neither **S3** nor **S2**.
- Most digital devices are designed to work with a supply voltage at or near 5 volts. All pins on such a device are given a power bus classification of **P5**. Pins on devices designed to work with other supply voltages are given a **Pxx** classification, where **xx** is the absolute value of the supply voltage.
- Passive devices and devices that work with a range of supply voltages are given a power bus classification of **?**.
- On devices with more than one supply voltage, some pins will often be clearly associated with one voltage or the other. Pins without a clearly defined supply are assigned a **?**.
- Balanced signals do not occur by accident and pins on components designed to send or receive balanced signals should be easily identified. Most digital logic inputs and outputs are classified as unbalanced (i.e., **UNB**) while the pins of most passive components (other than baluns) are classified as **?**.
- The general rule for classifying pins as I/O is very simple. If the component is a connector, all the pins are **YES**. If the component is not a connector, all of the pins are assigned a **?**. Any component pins that should not normally be connected to I/O are assigned a **NO** for this variable. The EMCheck code will flag any connection between a **NO** and a **YES** alerting the user to the potential problem.

Any components on the board being evaluated that are not in the look-up table will be classified as **R1S1?2?2** by default. If there are too many unknown components on the board, the value of the rule checking software will be greatly diminished. Therefore, it is important to maintain an up-to-date look-up table. Adding new components to the look-up table is very straight forward. Ideally the EMC engineer responsible for a particular product design would ensure that the look-up table includes all components of interest to the board designers.

Assigning net variables

The EMCheck code classifies nets by looking at all of the component pins attached to the net. Radiation and susceptibility classifications are assigned on a worst-case basis. For example, a net connecting four **R2** pins and an **R3** pin would receive an **R3** classification. The power bus classification of the net is the same as the power bus classification of the pins it connects. In the event that a net connects two or more pins with different power bus ratings, the user is warned of a potential layout error. The I/O and balanced signal classifications also must be unani-

mous. A net connected to pins with **BAL** (or **YES**) and **?** classifications will be assigned a **BAL** (or **YES**). All **UNBs** (or **NOs**) and **?**s will cause the net to be classified as a **UNB** (or **NO**). A net with all **?**s will receive a balanced signal classification of **?** or an I/O classification of **NO**. A net connecting a **BAL** (or **YES**) pin to a **UNB** (or **NO**) pin will cause a warning message to be displayed.

The entire net classification process occurs automatically once the program is started and requires no user input. Once the nets are classified, the user is given an opportunity to view any or all of the nets and their classifications. The user can then opt to change the classification of any nets, if desired, before proceeding to the rule checking portion of the algorithm.

Rule Checking

The rule checking portion of the software is very flexible so that specific design guidelines enforced by the algorithm can be easily modified to suit the needs of the user. Design rules for the EMCheck code are checked in subroutines that run independently. Input for each subroutine is provided by the net classification file and/or by passing specific parameters in the call from the main program.

The source code of each rule checking subroutine contains a small section set off by rows of asterisks where important variables are set or violation decisions are made. The purpose of the asterisks is to highlight places where the subroutines can be modified to suit the needs of the user. For example, the crosstalk subroutine contains a variable called **MAXR3S3**. If the calculated crosstalk factor between an **R3** and **S3** net is greater than the value of this variable, a violation is flagged. By raising or lowering the value of **MAXR3S3**, the user can adapt this rule to be more forgiving or more stringent.

The following sections describe each of the rule checking subroutines included in the initial version of EMCheck. These rules are intended to be applied to a specific class of multilayer printed circuit boards with specific design objectives. However, from the descriptions provided, the reader can gain an appreciation for the type of design problems that automated board evaluation software can identify.

Long nets

Nets that are most susceptible or most likely to radiate should be among the shortest nets on the board. The longR3S3 subroutine calculates the length of all **R3** and **S3** nets. Since nets may have many branches, the calculated length is actually the sum of the segment lengths and may not be the same as the end-to-end trace length. **R3** nets with a calculated length greater than the value of **MAXR3** are flagged as being in violation of this design rule. **MAXR3** is a function of the overall board dimensions. The formula provided for calculating **MAXR3** is

$$MAXR3 = 0.5\sqrt{Board\ Length \times Board\ Width}$$

This formula is set off by rows of asterisks in the longR3S3 subroutine and is easily modified by the user. The value of **MAXS3** puts an upper limit on the allowable length of **S3** nets and its value is calculated in a manner similar to that of **MAXR3**.

Crosstalk

When two nets have traces that are routed very close together, it is possible for the signal on one net to couple over to the other net. The crosstalk subroutine calculates a factor that gives the user an indication of the likelihood of having a crosstalk problem between two nets. The formula used to calculate the crosstalk factor is,

$$XFACTOR(Net1, Net2) = \sum_{i=1}^N \sum_{j=1}^M \frac{\sqrt{|Net1_i \cdot Net2_j|}}{d_{ij}}$$

where, $Net1_i$ is the vector in the plane of the board defined by segment i of NET1.

$Net2_j$ is the vector in the plane of the board defined by segment j of NET2.

d_{ij} is the distance from the center of segment i to the center of segment j .

XFACTOR is a unitless quantity that is greatest for nets with long, closely spaced, parallel segments. The crosstalk subroutine calculates XFACTOR for all possible combinations of R3, S3, and I/O nets. The variable MAXR3S3 contains the highest permissible value for the XFACTOR between an R3 and an S3 net. MAXR3IO and MAXS3IO are the corresponding variables for R3 to I/O and S3 to I/O XFACTORS.

Components over incorrect power plane

On dense multilayer boards, D.C. power will typically be supplied to every active component through power planes. Ideally, these planes are confined to designated layers in a multilayer board. This helps to ensure lateral separation between analog and digital components and provides low impedance power distribution, which is essential for the reliable operation of high-speed printed circuit boards. Traces connected to power or ground pins are an indication that the component is improperly positioned with respect to the power or ground planes. This subroutine checks the length of any trace segment connected to the power or ground pin of an active component. Any segment length greater than MAXPG (default = 1 cm) is considered a violation. The user is advised to relocate the component so that it is positioned above the correct power and ground planes.

Traces crossing gaps

Unbalanced signals use the ground plane of the board as a current return path. At frequencies above a few kilohertz, most of the return current flows directly beneath the signal trace. When unbalanced signal traces pass over a gap in the ground plane, return currents are forced to take a different (higher impedance) return path. This degrades the signal quality and makes it easier to couple unwanted electromagnetic interference into or out of the circuit. Since there is seldom a good reason to route an unbalanced signal trace over a gap in the ground plane, the code flags a violation every time this occurs. The offending net is named and the recommendation is to remove the gap or relocate the offending signal trace.

Gaps under connectors

One of the most fundamental and important design guidelines pertaining to high-speed digital circuit board ground, is that no matter how many different types of ground exist on the board, there can be only one ground reference at the I/O connectors. Consequently, it is undesirable for a gap in the ground plane to extend under a connector. This subroutine compares the location of every connector with that of any gaps in the ground plane. Any gap extending under a connector is flagged as a violation.

Thin-necked area fills

Area fills are used to define power and ground planes or to provide a low inductance signal current path. When the width of an area fill is substantially reduced in one place as illustrated in Figure 3, current flowing through the region of relatively high inductance results in a voltage that develops between the wider portions of the area fill. This voltage difference coupled with the high capacitance of the wide area fills represents a source of unwanted noise that should be avoided. This subroutine examines the shape of all area fills to ensure that they do not contain *thin necks*. The definition of a thin neck used by this subroutine is any section of the area fill with width, W , that connects two larger areas $A1$ and $A2$ such that

$$W \leq 0.10 \sqrt{\min(A1, A2)}$$

Summary

Rule checking codes look for circuit board design features that may result in electromagnetic compatibility problems, just as an experienced EMC engineer would. By flagging potential design flaws, a rule checking code can help board designers to avoid mistakes that could lead to serious EMC problems. Repeated application of a rule checking code to new designs helps teach board designers good EMC design practices.

Rule checking codes, like EMIcheck, do not use numerical modeling techniques and they cannot estimate the effect that a specific design change will have on the radiated field strengths. However, unlike numerical modeling codes, rule checking codes are easy to use, even by people with little or no EMC experience or training. A rule checking program can coexist on the same system as the board layout files and can be run by the board designer at any stage of the design. A well-written rule checking algorithm need not require any input from the user at all, since all of the board layout information is contained in the board layout files.

Rule checking software does not eliminate the need for a qualified EMC engineer to be involved in the design process. A qualified EMC engineer can ask pointed questions concerning the function and intended environment of a given board. Also, the EMC engineer with a specific design strategy in mind, may invent new rules or recommend changes that violate basic design rules in order to accomplish certain goals. Nevertheless, rule checking software can analyze many aspects of the board design that may not be apparent to the EMC engineer working with artwork and schematics. Also, because rule checking codes are fast and easy to use, a board design can be reviewed more frequently. For these reasons, rule checking software is likely to become a standard tool for board designers using automated board layout software in the future.

Acknowledgement

This work was supported by Boeing Commercial Avionics Systems.

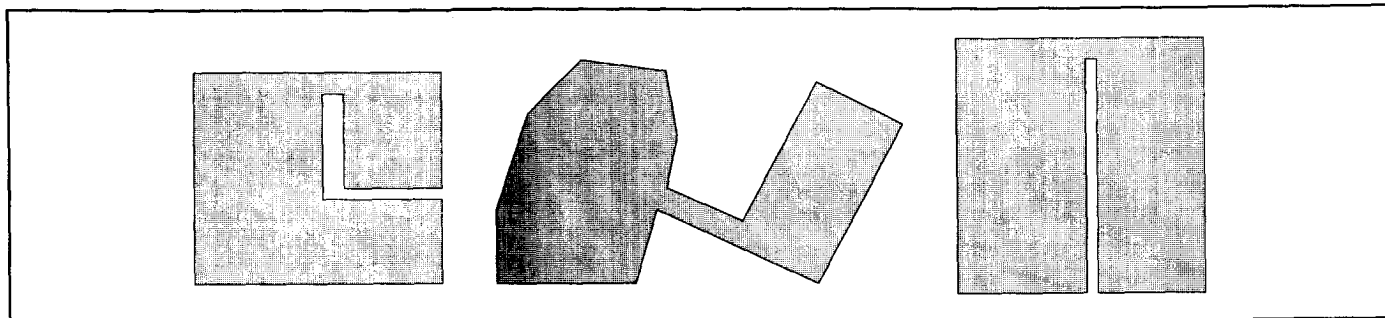


Figure 3: Examples of area fills with "thin necks"