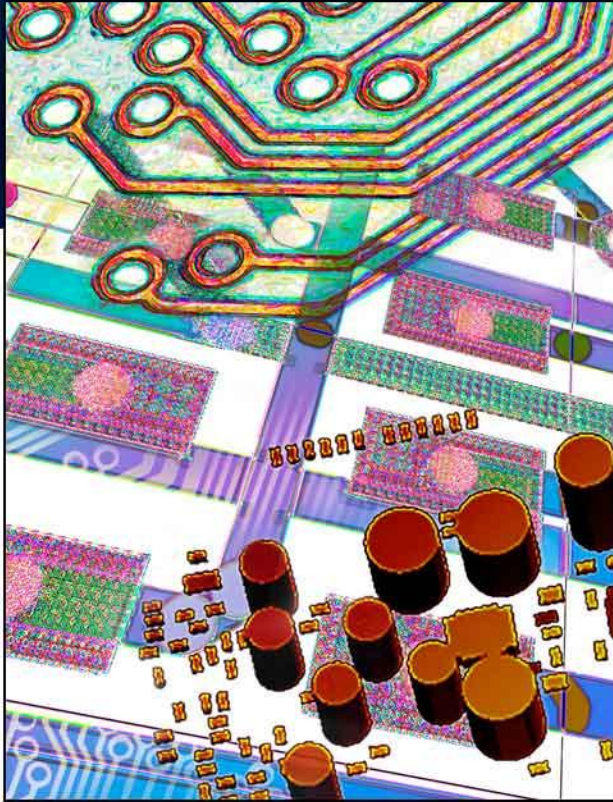




LPB Formatを用いた Package+PCB EMI解析

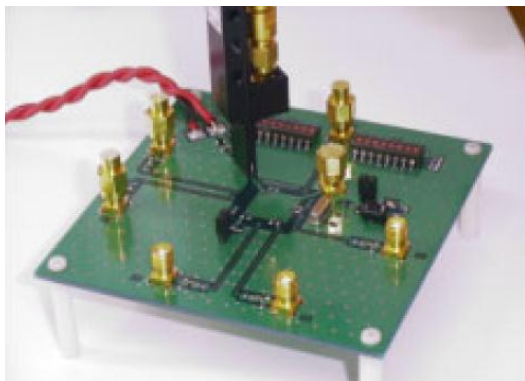
2015.03.19

Mentor Graphics Japan Co.,Ltd

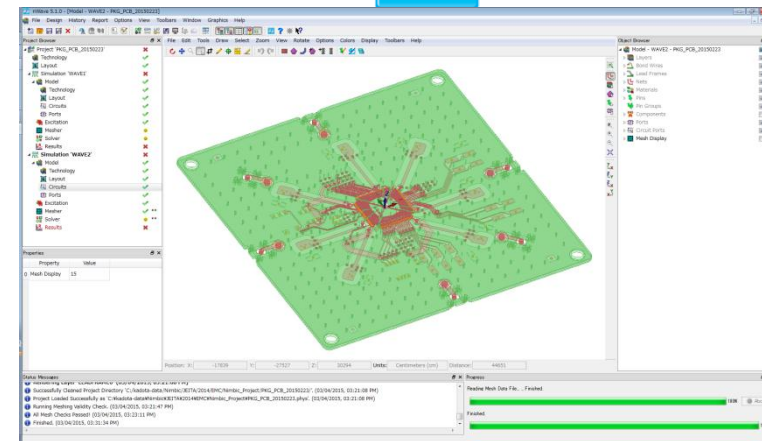
Kazuhiro Kadota

解析内容と目的

- LPB-Formatを用いて、JEITA 半導体EMC サブコミティーで実施したシミュレーションvs 実測比較Modelの解析を実施
- Nimbic製品のLPB-Formatサポート(2015年3月現在)
 - G-Format
 - M-Format(ver.2.0)
- 本解析においては、G-Format以外にPython Scriptを用いて C および N-Formatを活用
- Scriptによる Package Model と PCB Model のMergeを実施
- 煩雑なPackage/PCB ModelのMerge Model作成作業を効率化
- 作成したPackage+PCB ModelでEMI解析を実施



半導体EMCサブコミティーより



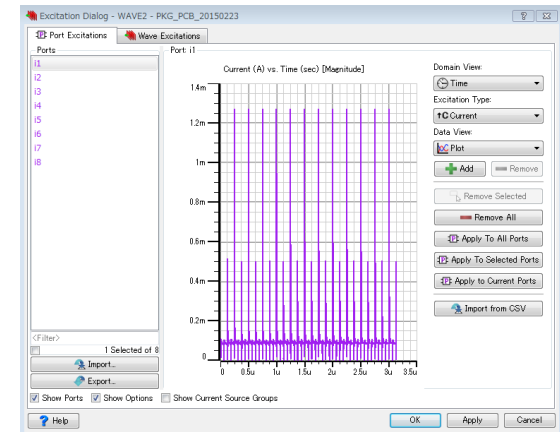
解析に用いるデータ

■ LPB Format

- G-Format (Geometry Format)
 - Package : step15_forum_IC1_G.xfl
 - PCB : MICON1_new.xfl
- C-Format (Component Format)
 - PCB : step14_drule_IC1_C.xml
- N-Format (Netlist Format)
 - PCB : step20_Forum_N.v

■ Excitation

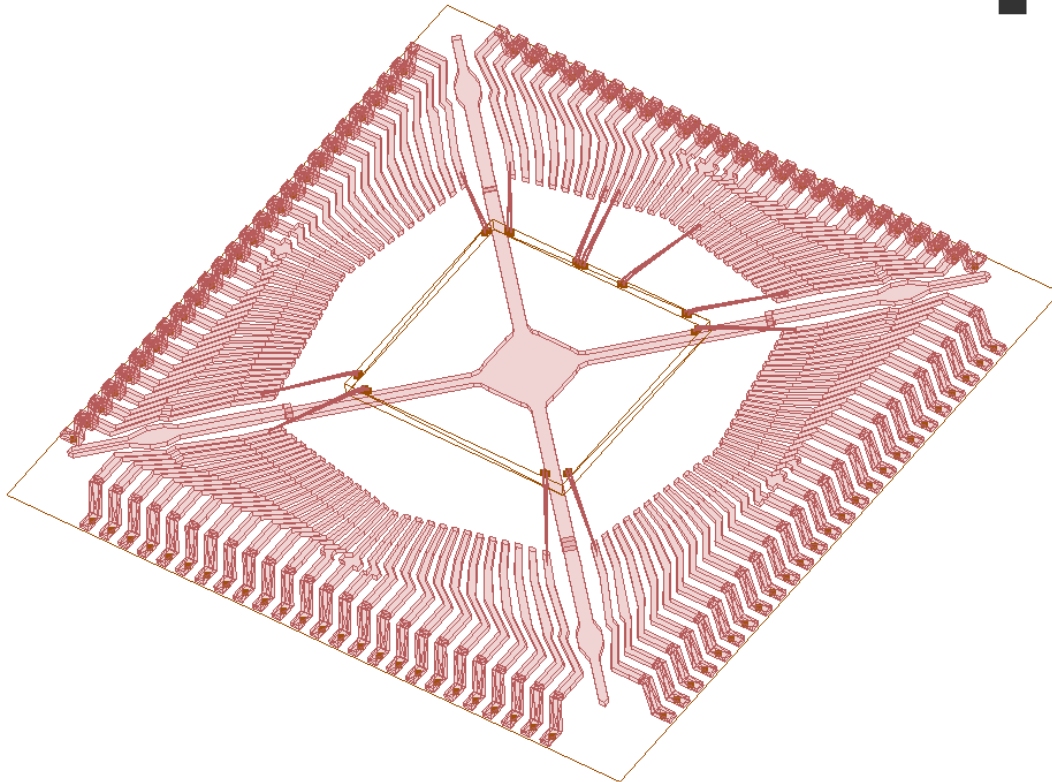
- CPM File : PowerModel.sp/PowerModel.sp.inc



G-Formatを用いてModel作成(Package)

■ Package Model

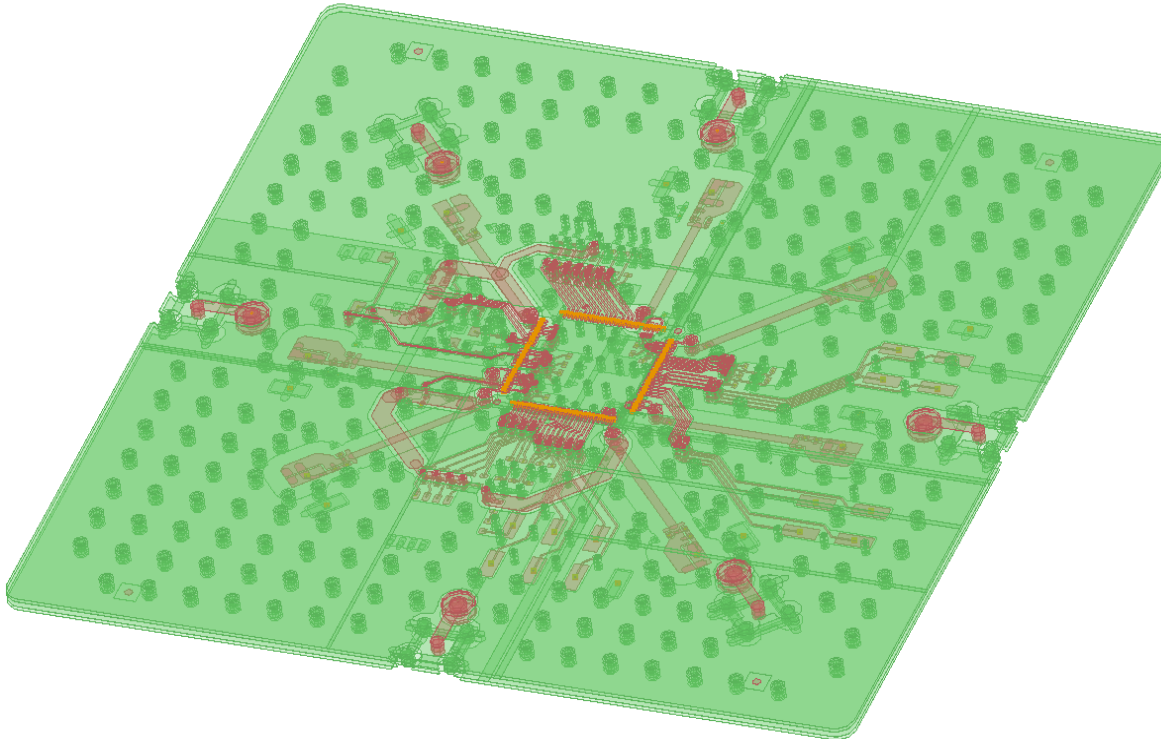
- G-FormatをImport
- Bond Wire はG-Formatに含まれている
- Lead FrameをnWaveの機能で自動生成



G-Formatを用いてModel作成(PCB)

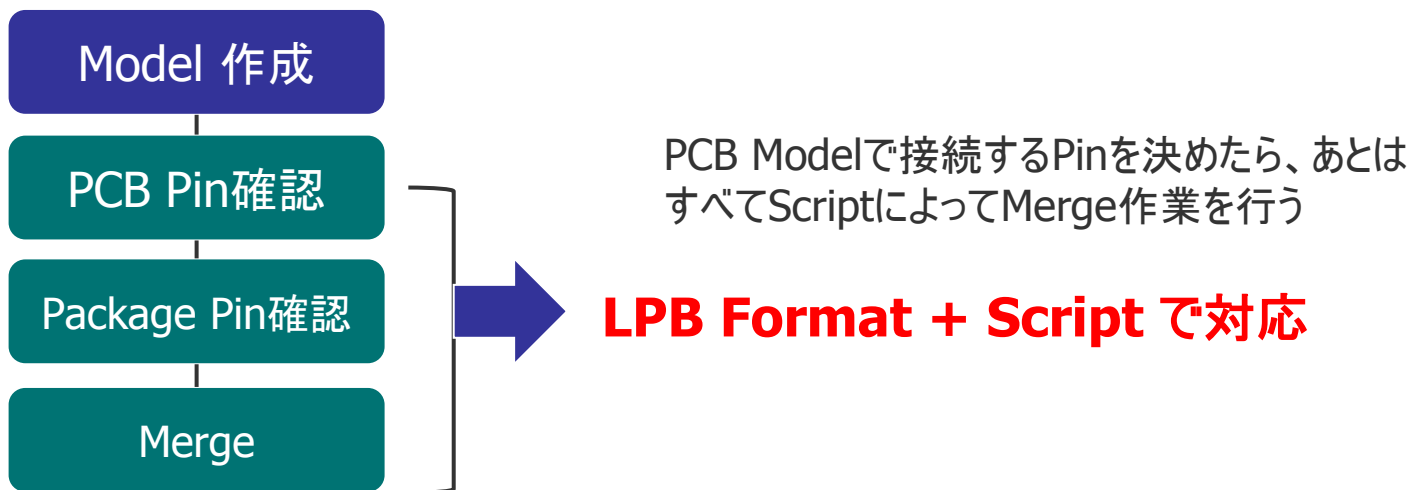
■ PCB Model

■ G-FormatをImport



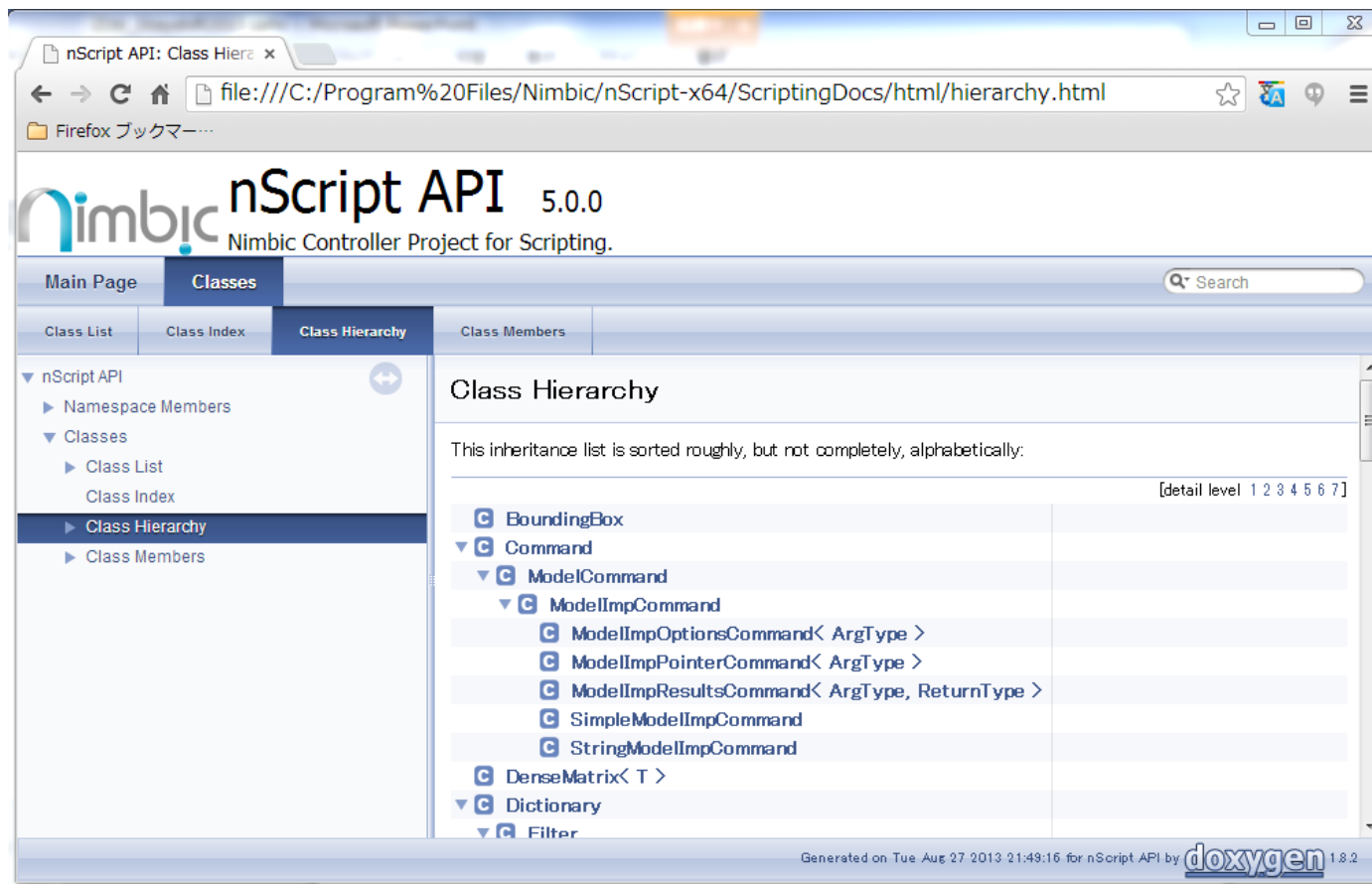
PackageとPCBのMerge

- 通常の手順 (Nimbicの場合)
 - それぞれのModelを作成したのち、接続する2か所のPinを目視で確認してnWaveのMerge機能を使ってMerge
- 通常の手順での問題点
 - それぞれのModelでPinを確認しなければMergeすることができない
 - 目視によって接続Pinを確認するため、人的ミスが発生しやすい
- Merge Script



Nimbic nScrip

- nScript は、Python script によるNimbicツールコントロール機能
 - Nimbic 製品のほとんどのUI をScriptで制御可能
 - 各種制御コマンドについてはHTMLドキュメントを用意



Merge Script

■ Merge Script Flow

PCB
Layout



Package
N-Format



Package
C-Format



Merge

1. PCBの接続Pin名とNet名を確認

Pin :IC1.50 > Net :SIGN143

Pin :IC1.76 > Net :SIGN116

2. **PCB N-Format**で、PCBのNet名から接続するPackageのNet名を検索

SIGN116 > net27

SIGN143 > net26

3. **Package C-Format**で、Package Net名からPackageの接続するPort名を検索し、Module名と組み合わせPackage Pin名を指定

net27 > 25 > PKG_1008_A.25

net26 > 51 > PKG_1008_A.51

Module名 : PKG_1008_A

4. 確認できた2か所の接続するPin名を用いてMerge機能でPackage+PCBモデルを作成

Net接続からPinを指定するのでミスが無い

Merge Script

■ Python Script

— 今回のScriptは、100行足らずの記述で構成

```
1 # coding=utf-8↓
2 # !python↓
3 import os↓
4 import re↓
5 ↓
6 from nimble import ncore↓
7 ↓
8 c = ncore.Controller("Wave") # Controller↓
9 ↓
10 ↓
11 def thisDir(fileName):↓
12     dir = os.path.dirname(os.path.abspath(__file__))↓
13     fn = os.path.join(dir, fileName)↓
14     return fn↓
15 ↓
16 # ***** Package Net Search*****↓
17 f_N = open(thisDir("step20_forum_N.v")) # PCB N-Format Import↓
18 linesPCB = f_N.readlines()↓
19 f_N.close()↓
20 ↓
21 pcbNet1 = "SIGN116" # PCB Net name1 Input↓
22 pcbNet2 = "SIGN143" # PCB Net name2 Input↓
23 pkgNet01 = None↓
24 pkgNet02 = None↓
25 ↓
26 for line in linesPCB:↓
27     if ".net" not in line:↓
28         continue↓
29     # print line↓
30     if pcbNet1 in line:↓
31         netName = re.findall(r"net%d+" % pcbNet1, line)↓
32         pkgNet01 = netName[0]↓
33         print("pcbNet1 found: %s" % pkgNet01)↓
34     if pcbNet2 in line:↓
35         netName = re.findall(r"net%d+" % pcbNet2, line)↓
36         pkgNet02 = netName[0]↓
37         print("pcbNet2 found: %s" % pkgNet02)↓
38     pass↓
39 ↓
40 if not pkgNet01:↓
41     raise RuntimeError("Unable to find 'pkgNet01' from 'pcbNet1': " + pcbNet1)↓
42 if not pkgNet02:↓
43     raise RuntimeError("Unable to find 'pkgNet02' from 'pcbNet2': " + pcbNet2)↓
44 print("pkgNet01: %s" % pkgNet01)↓
45 print("pkgNet02: %s" % pkgNet02)↓
46 ↓
47 ↓
48 # ***** Package Pin Search*****↓
49 f_C = open(thisDir("step14_druid_IC1_C.xml")) # Package C-Format Import↓
50 lines = f_C.readlines()↓
```

■ その他Scriptで対応可能な機能

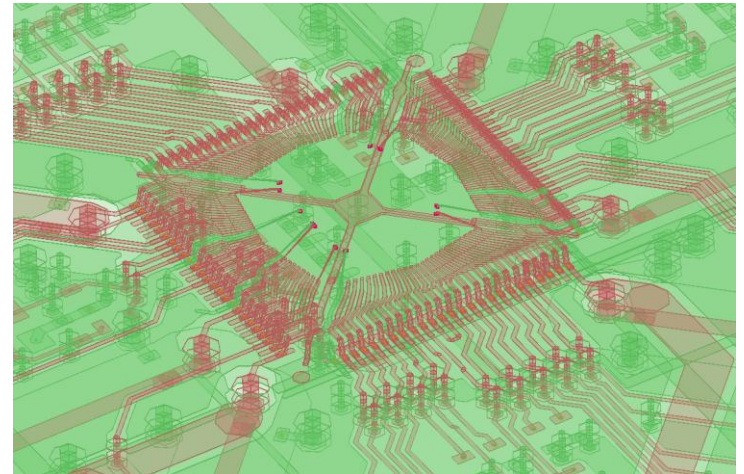
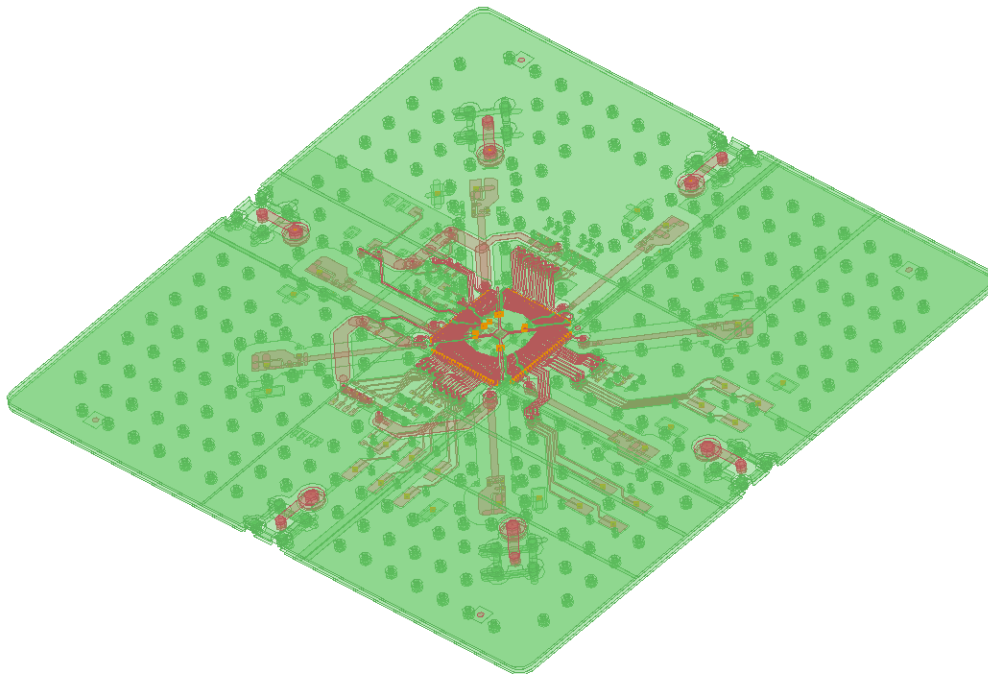
- G-FormatのImport
- Lead Frameの自動生成
- CPMのPinへのアサイン
- Mesh設定
- 解析設定
- GUIの作成

— etc...

Scriptで作成されたPackage+PCB Model

■ Package+PCB Model

- 実装されるRC チップ部品を設定
- CPMで励振源を設定
 - CPMデータが理想GNDをReferenceとしたデータであったため、今回は各端子のPWLデータに分割してそれぞれ設定
 - 励振源を設定した時点で解析Portは自動生成される



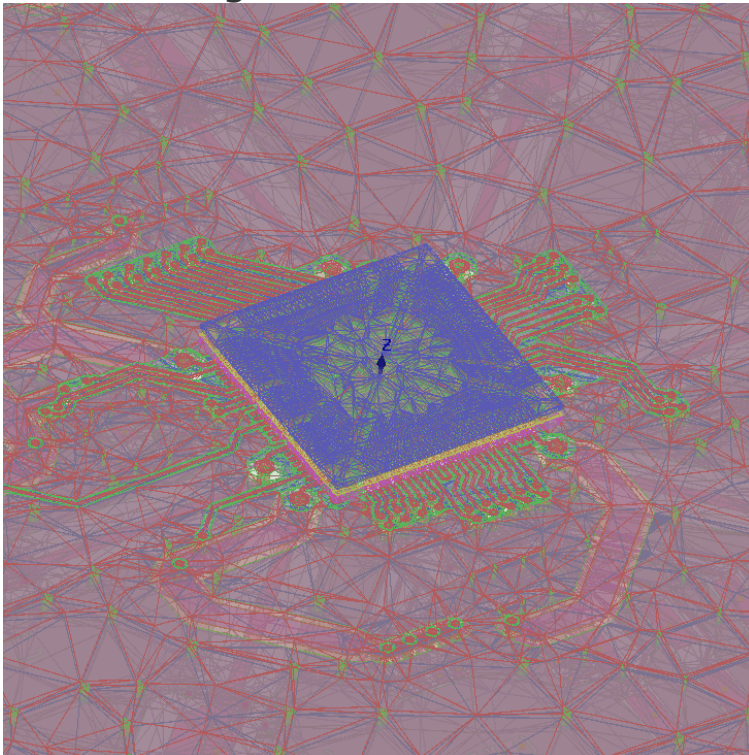
The background features a dark blue gradient with two distinct radial light patterns. The top pattern consists of numerous thin, greenish-blue lines radiating from the top center towards the edges. The bottom pattern consists of similar lines, but in a slightly darker blue shade, radiating from the bottom center. These patterns create a sense of depth and focus on the central text.

Merge Script Demo

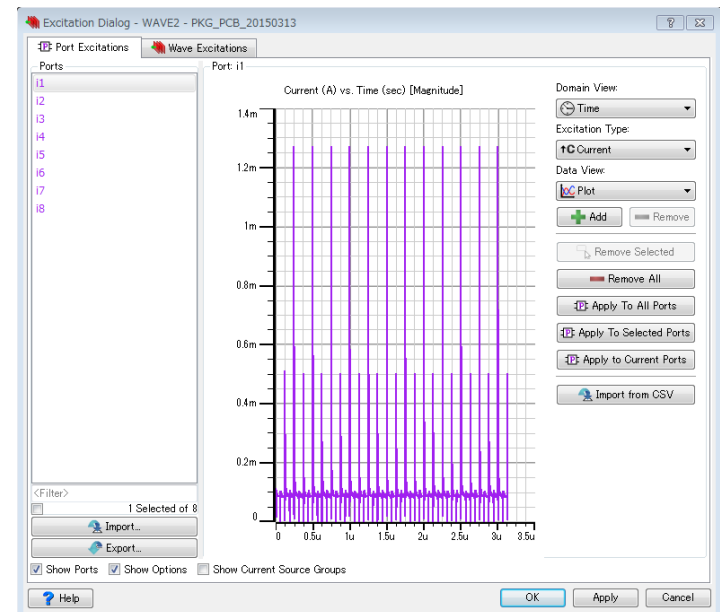
EMI解析

- Mesh Frequency=2GHz でMeshを作成
- 解析周波数 : 1MHz~1GHz 30point
- 電磁場表示のために、解析設定でPackage上面から5mmの高さのPlaneを設定

Package+PCB ModelのMesh



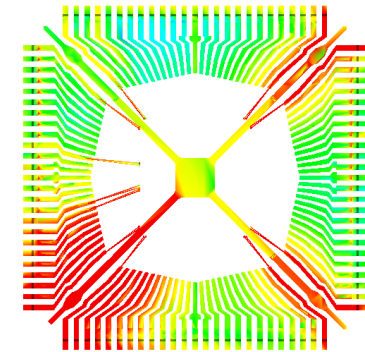
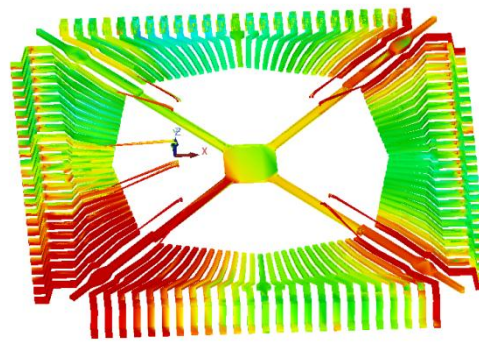
励振条件の設定



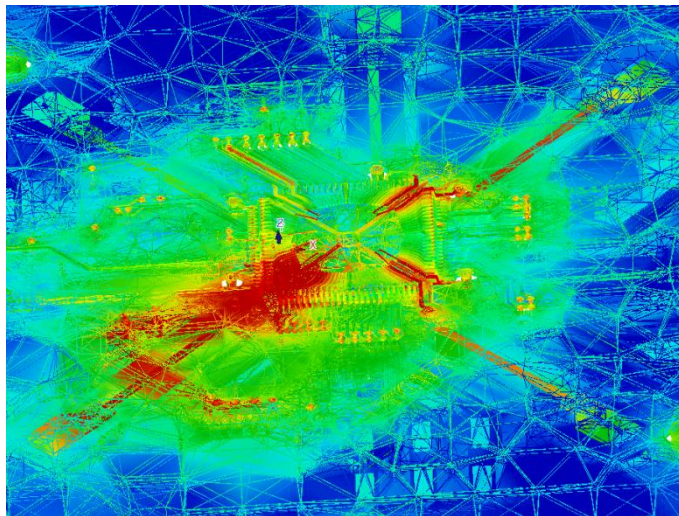
解析結果(1)

- 電流分布
 - Frequency = 1GHz

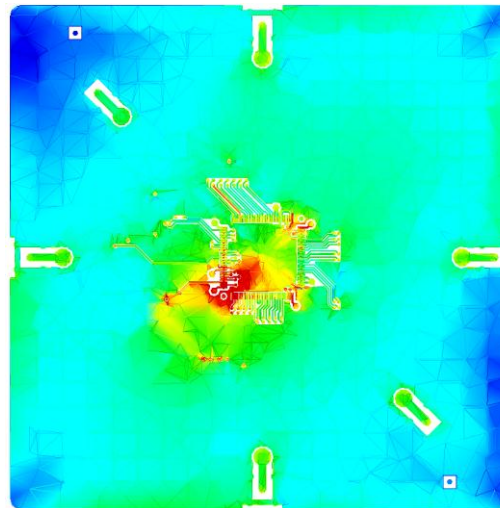
Package



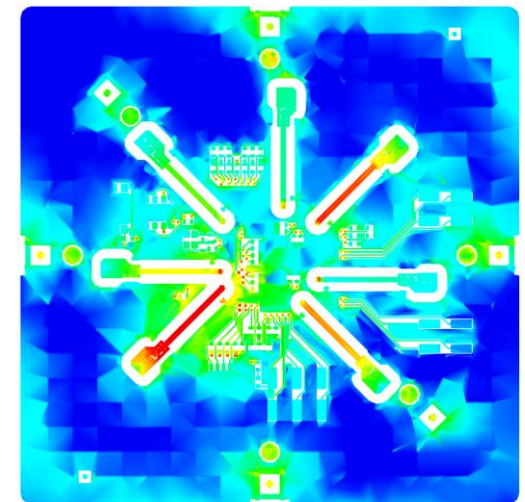
Package + PCB



PCB Top



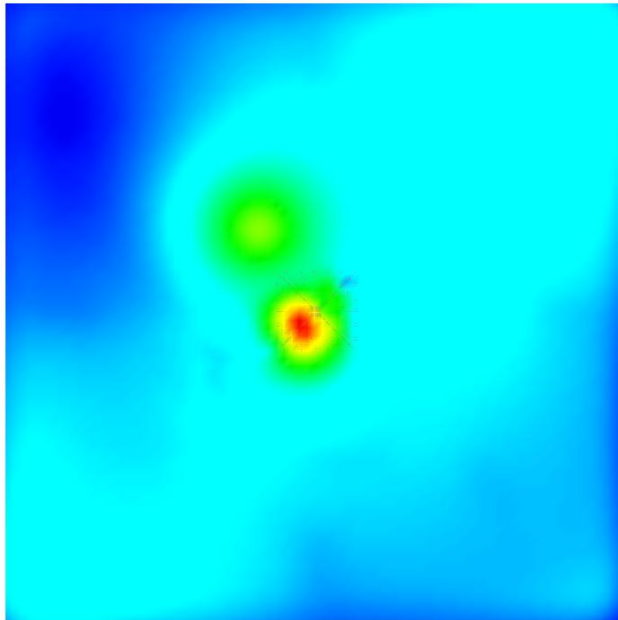
PCB Bottom



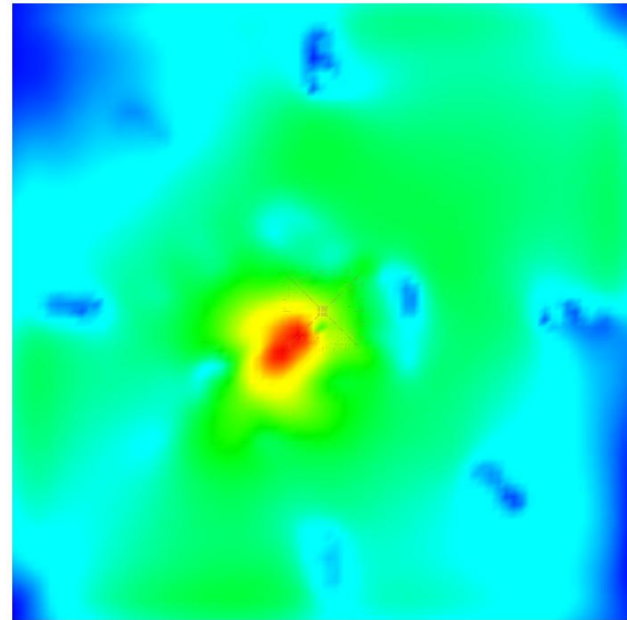
解析結果(2)

- 電界および磁界
 - Package上面から5mmの位置
 - Frequency = 1GHz

E-Field

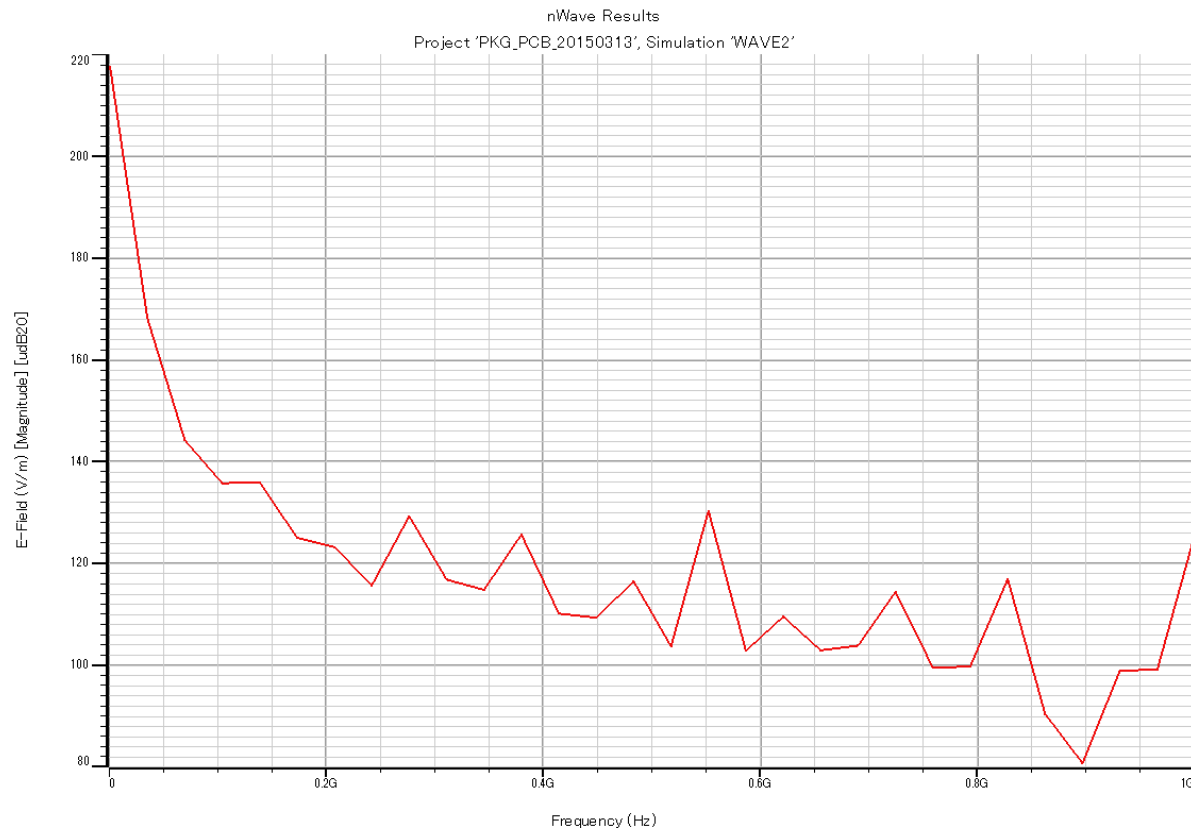


H-Field



解析結果(3)

■ 電界強度 — 1MHz~1GHz 30point



Conclusion

- ScriptでLPB-Format を用い、PackageとPCBモデルのMergeを実施
- C-FormatおよびN-Formatを用いることでPackage/PCBのNet名から接続するPinを参照してMerge
- LPB-Format を活用することで、Package + PCBモデル作成作業は2/3程度に削減
- 通常の手順で発生しやすい人的ミスによるやり直し作業を回避
- 今回はG-FormatのImport、CPMの設定については手作業で行ったが、これらの作業もScriptで対応が可能
- これらすべてのModel作成作業にLPB-Format と Scriptを活用すれば、Model作成時間は1/3程度には短縮出来ると考える
- 今回は時間の関係上、実測比較等の精度検証は行えなかったが、引き続き精度検証、およびLPBを活用した更なる効率化を進めていきたい



www.mentor.com