

メンター・グラフィックス Nimbic製品 LPB Formatサポート

メンター・グラフィックス・ジャパン
2015年3月19日

プレス発表

■ Nimbicを買収

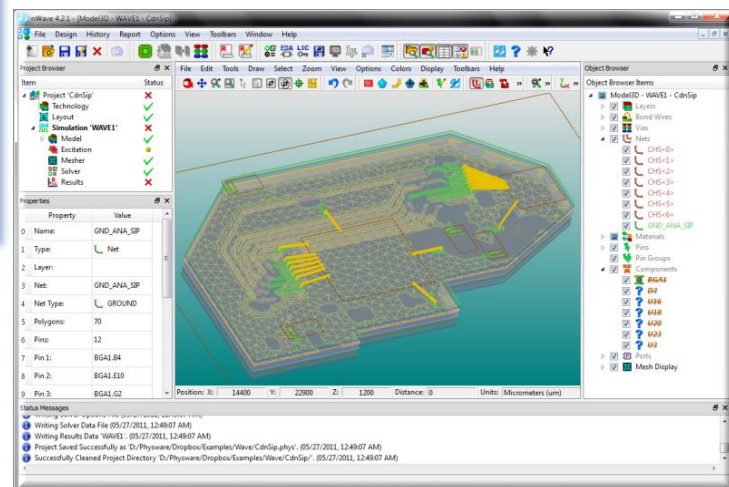
製品情報 PCB設計ソフトウェア&ツール ニュース メンター・グラフィックス、Nimbicを買収

メンター・グラフィックス、Nimbicを買収

2014年05月22日

メンター・グラフィックス・コーポレーション(本社: 米国オレゴン州、以下メンター・グラフィックス)は、マックスウェル精度の3Dフルウェーブ電磁界(EM)シミュレーションソリューションで業界をリードする、Nimbic, Inc.(本社: 米国カリフォルニア州、以下Nimbic)を買収したことを発表しました。Nimbicの高性能なハイエンド向けシミュレーションと複雑な電磁界を正確に解析する技術は、メンター・グラフィックスが提供するチップ・パッケージ・ボードのシミュレーション製品ラインを拡大強化します。

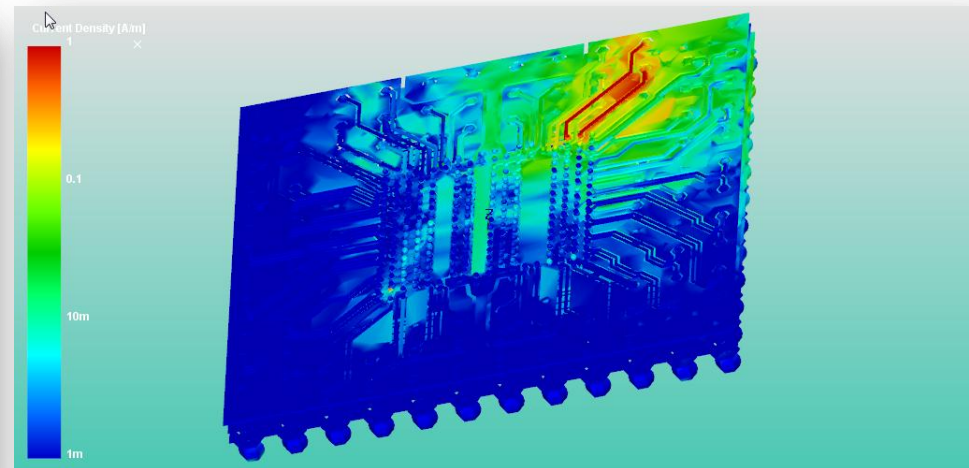
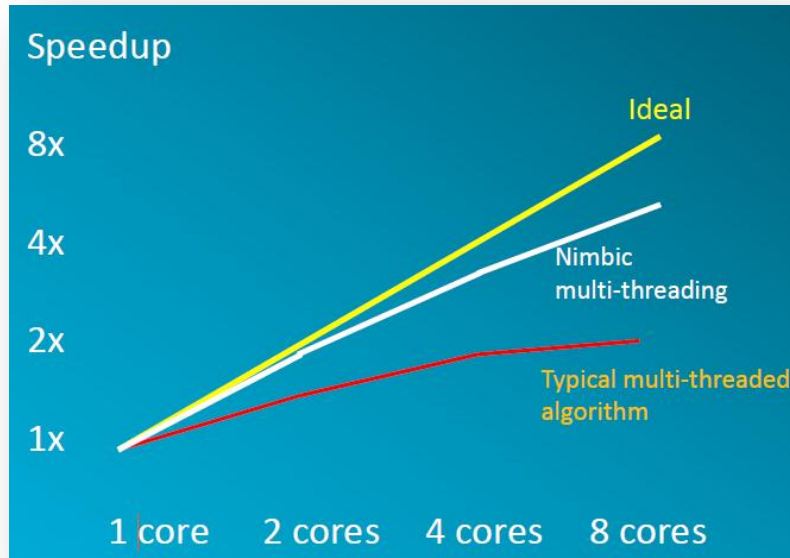
「世界トップレベルを誇る、Nimbicのシグナルインテグリティ(SI)、パワーインテグリティ(PI)、EMI(電磁干渉)解析向け3D電磁界シミュレーションソリューションは、エレクトロニクス産業を代表する多くの企業によって採用されており、チップ・パッケージ・ボードの全体設計におけるエンタープライズレベルの課題を解決し、とどまることのない設計の複雑化に直面している業界を支援します。メンター・グラフィックスは、プリント基板(PCB)およびパッケージのシステム設計におけるリーダーシップ、グローバル企業としての存在感、大規模な顧客ネットワークを持っており、理想的な買収先と言えるでしょう。Nimbicの顧客、従業員、そして業界全体にとっても、今回の買収は非常に喜ばしいことだと考えています。」Nimbic、CEO、Raul Camposano氏は、上記のように語っています。



Nimbic製品について

■ 製品の特長

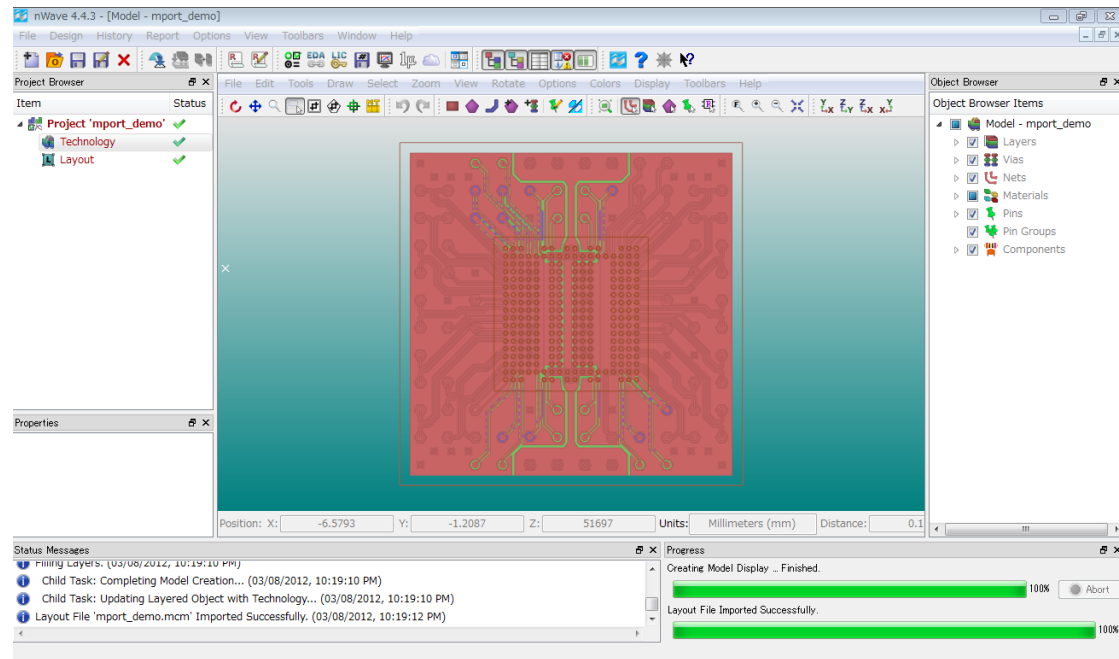
- Package/PCB特性解析に特化した電磁界解析ツール
 - Package/PCBのレイアウトからシミュレーションまでのユーザーの設定を少なくし、効率的な解析を実現
- 独自の高速境界要素法による高速な大規模モデルの解析
 - マトリクスの並列処理能力を持ち次世代大規模マトリクス計算が可能
- 3D電磁界ソルバであるので、BGA, ボンディングワイヤ, リードフレームの解析も可能
- ECADデータをダイレクトにImport可能なシームレスなECADリンク機能
- 解析結果から自動的にグラフ作成



ECAD Import機能

■ 各社ECAD データを直接Import

Multi-Chip Module (*.mcm)
Board Layout (*.brd)
System in Package (*.sip)
GDSII (*.gds *.gdsii *.gds2 *.sf *.strm)
DXF (*.dxf)
Layout Format 1 (*.xfl)
Caltech Intermediate Form (*.cif)
Mentor Layout (*.hkp)
HyperLynx Layout (*.hyp)
Zuken Layouts (*.pcb *.rif *.pcf *.ftf)
JEITA LPB Project (*.xml)
Nimbic ASCII Model - Text (*.pam)
Nimbic Saved Geometry - Binary (*.geom)
Boolean Scripting Language (*.bool)
All Files (*)



■ Import ダイアログから各ECADファイルを直接指定することで 容易に電磁界モデルの作成が可能

LPB Format Support

- Nimbic ツールが現状サポートしているフォーマット
 - G-Format
 - M-Format (ただしLPB ver2.0)
- その他フォーマットへの対応
 - ScriptでC-FormatおよびN-Formatを用いたPackage-PCB マージ

Example: LPBファイルを用いた自動化

■ ScriptによるLPBフォーマットのImport とPackage/Board Merge モデルの作成

```
#!/python
from nimbic import ncore
```

```
c = ncore.Controller("nWave") # Controller
```

基本設定

```
def importDesign(strDesignFile):
    "Import a Design File; return the Project and Model."
    global c
    dio = c.scanDesignFile(strDesignFile)
    strProjectFile = strDesignFile + ".phys"
    project = c.importDesignFile(strDesignFile, strProjectFile,
    dio)
    # project.writeProjectFile(strProjectFile)
    model = project.getModel()
    return (project, model)
```

```
def addSolderBall(model, layer, component, bump):
    t = model.getTechnologyDetails()
    ball = t.addNewSolderBall()
    ball.setBump(bump)
    ball.setName("SB.%"s" % component)
    ball.setComponent(component)
    ball.setSupportLayer(layer)
    ball.setTerminationLayer(ball.getName())
    ball.setDiameter("300")
    ball.setPadDiameter("300")
    ball.setPortDiameter("300")
    ball.setHeight("100")
    print "Solder Ball Added:", ball
    model.setTechnologyDetails(t)
```

Ball 設定

```
# Import Designs:
(p1, m1) = importDesign("Board_2_xflv1.xfl")
(p2, m2) = importDesign("memory_pkg.xfl")
# Add Solder Bumps to Board:
addSolderBall(m1, "TOP", "CHIP_B", True)
# Add Solder Balls to Memory Pkg:
#addSolderBall(m2, "Wir_2", "I_MYPKG", False)
p1.writeProjectFile("Board_2_xflv1.phys")
p2.writeProjectFile("memory_pkg.phys")
# Set the Pin Pairs:
mo = ncore.MergeOptions()
mo.addPinPair("CHIP_B.A9", "I_MYPKG.A9")
mo.addPinPair("CHIP_B.J9", "I_MYPKG.J9")
# Avoid Name Collisions:
mo.setNameSuffix("BRD")
#mo.setNameSuffixOther("PKG")
# Keep Pins at Merge:
#mo.setKeepInterfacePins(True)
# Keep Net Names of the Model being Merged.
Default False.
#mo.setKeepOtherNetNames(True)
```

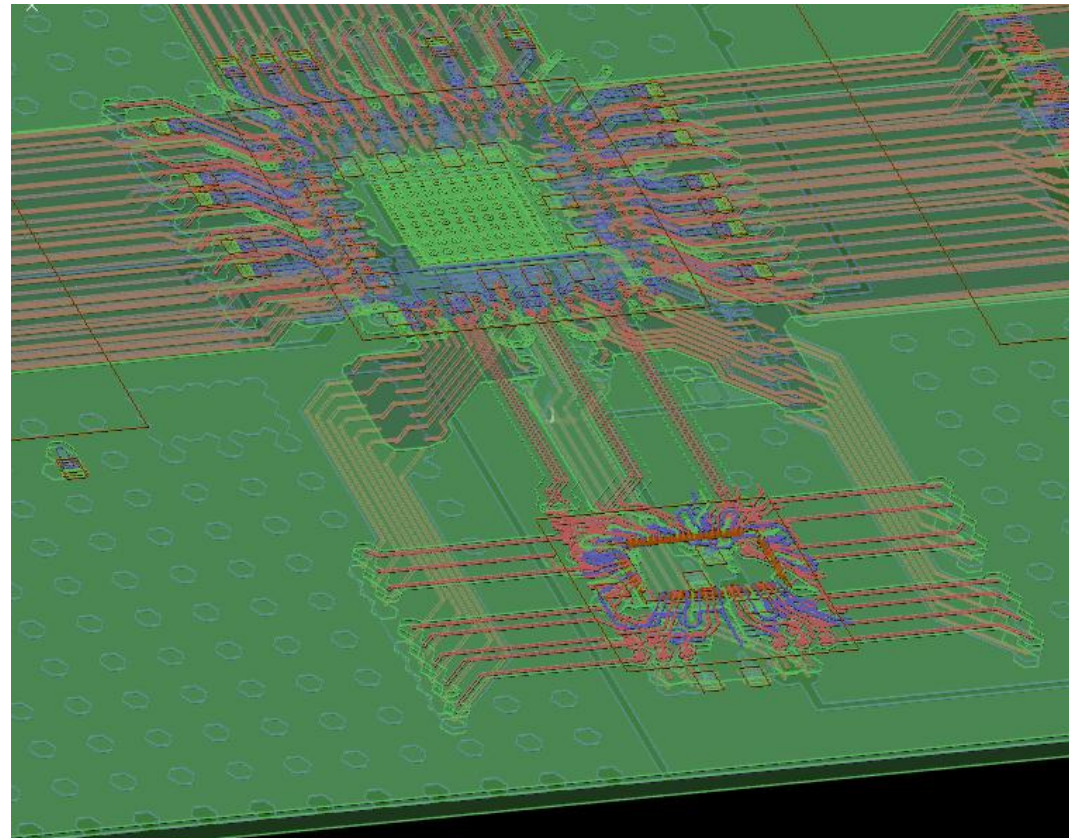
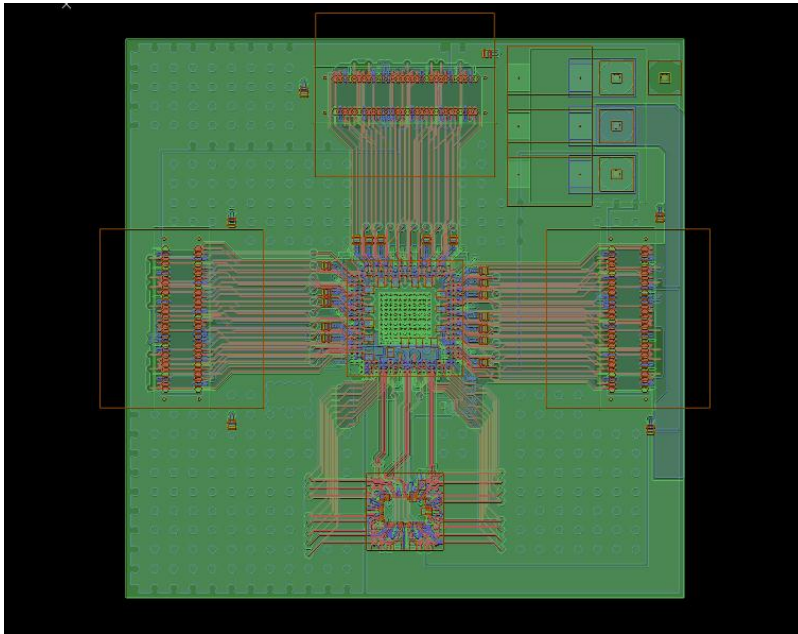
Layout Import

```
# Merge Model m2 into Model m1:
m1.merge(m2, mo)
# Save the Design:
p1.writeProjectFile("MemPkgOnBoard.phys")
```

Model Merge

Scriptにより作成されたMerge Model

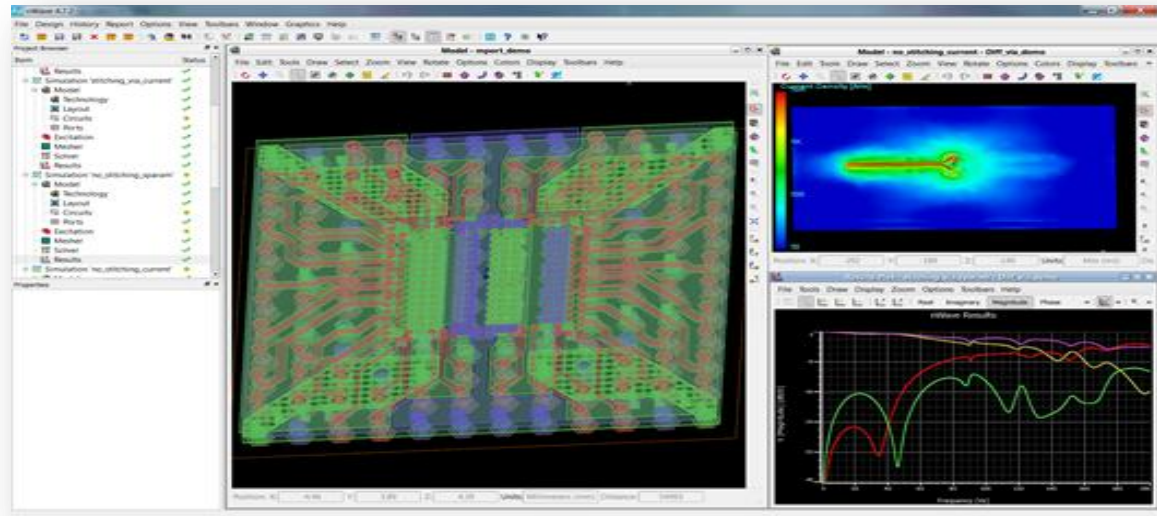
- ScriptによりMergeされたLPBフォーマットのPackage/Board Model
— 同様の手法で、シリコンインターポーザ等も含んだ解析が可能



まとめ

Nimbic製品とLPBフォーマット

- Nimbicはメンターの買収以前からLPBフォーマットをサポート
- 引き続きLPBフォーマットをサポート
- 今回のフォーラムで活用事例を発表





www.mentor.com