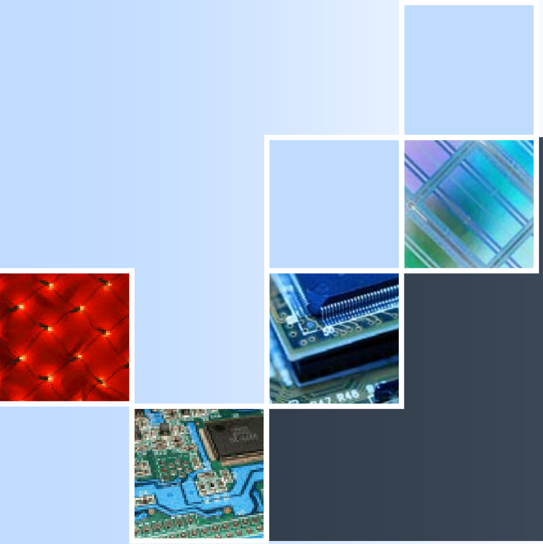


An introduction to model connection protocols

Brad Brim

bradb@sigirty.com

IBIS Summit

DesignCon

February 4, 2009

IBIS Model Connection Protocol
IBIS-MCP

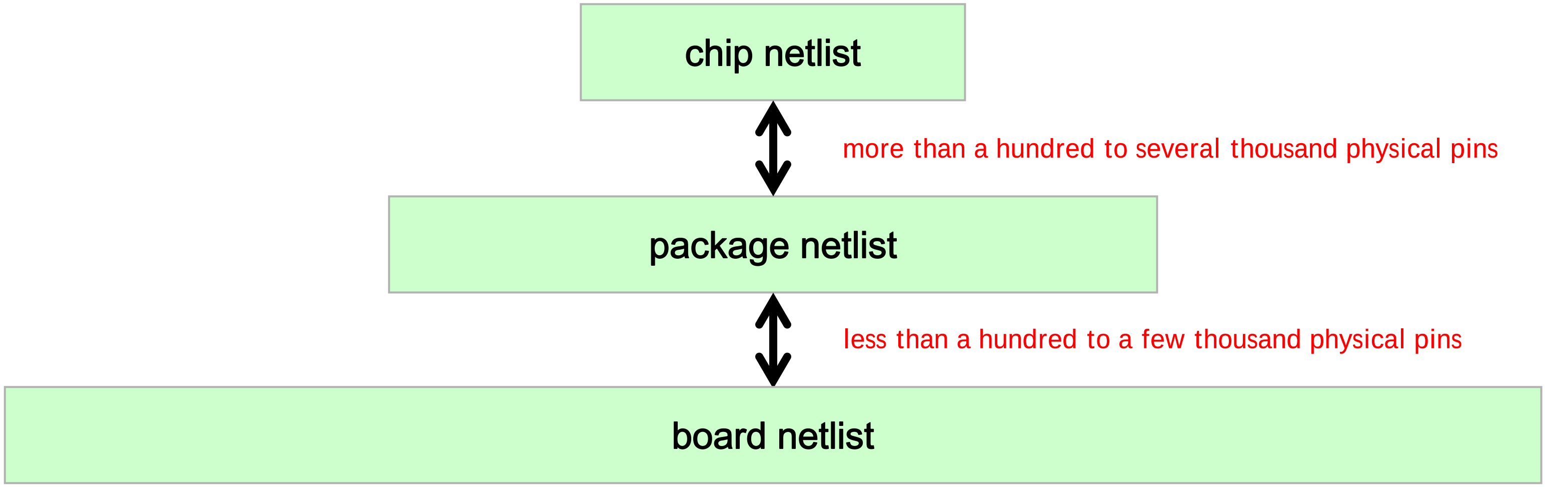
- Chip centric design requirements include
  - consideration of system loading effects
  - multi-chip design requires package models
- System design requirements for SI/PI/EMC include silicon-level drivers and power grid models

- Multi-vendor EDA flows are the norm, not the exception
  - EDA vendors may vary across chip/package/board
  - EDA vendors may vary for physical design versus extraction

- IBIS-MCP is a proposal to the IBIS Interconnect Task Group for a vendor-neutral method to specify electrical and physical connectivity information to enable automated connection of electrical models
  - MCP topic first discussed with IBIS community at DAC IBIS Summit in July, 2009
  - ref: “Model Connection Protocols for Chip-Package-Board System-level Analysis”, Brad Brim
    - http://www.vhdl.org/pub/ibis/summits/jul09/brim.pdf

Simple system

- How to quickly and reliably connect models for system-level analysis?
  - there may be literally thousands of physical connections
  - even with many physical pins “grouped” into single electrical nodes there may be hundreds of nodes to connect in a SPICE netlist
  - the order of nodes may not correspond from one netlist to the other



chip netlist

more than a hundred to several thousand physical pins

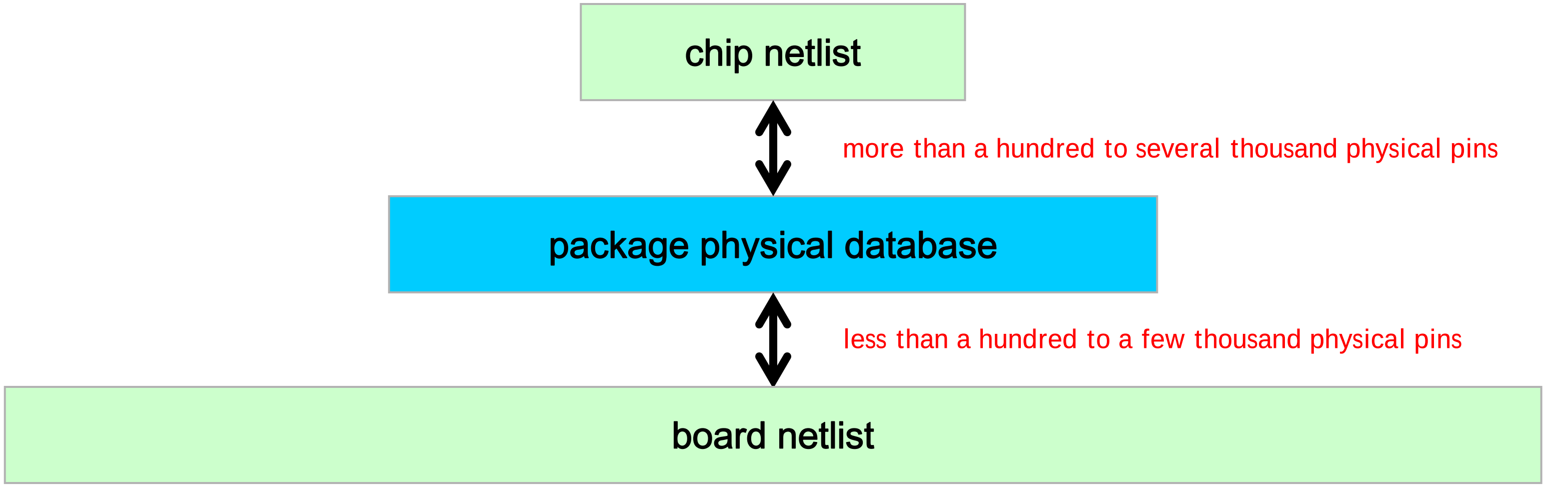
package netlist

less than a hundred to a few thousand physical pins

board netlist

Simple system

- This will be applied in EDA tools for both physical and electrical connectivity purposes
  - the process must be automated
  - the enabling mechanism must be EDA vendor neutral and model neutral
    - an industry standard connectivity protocol is needed



chip netlist

more than a hundred to several thousand physical pins

package physical database

less than a hundred to a few thousand physical pins

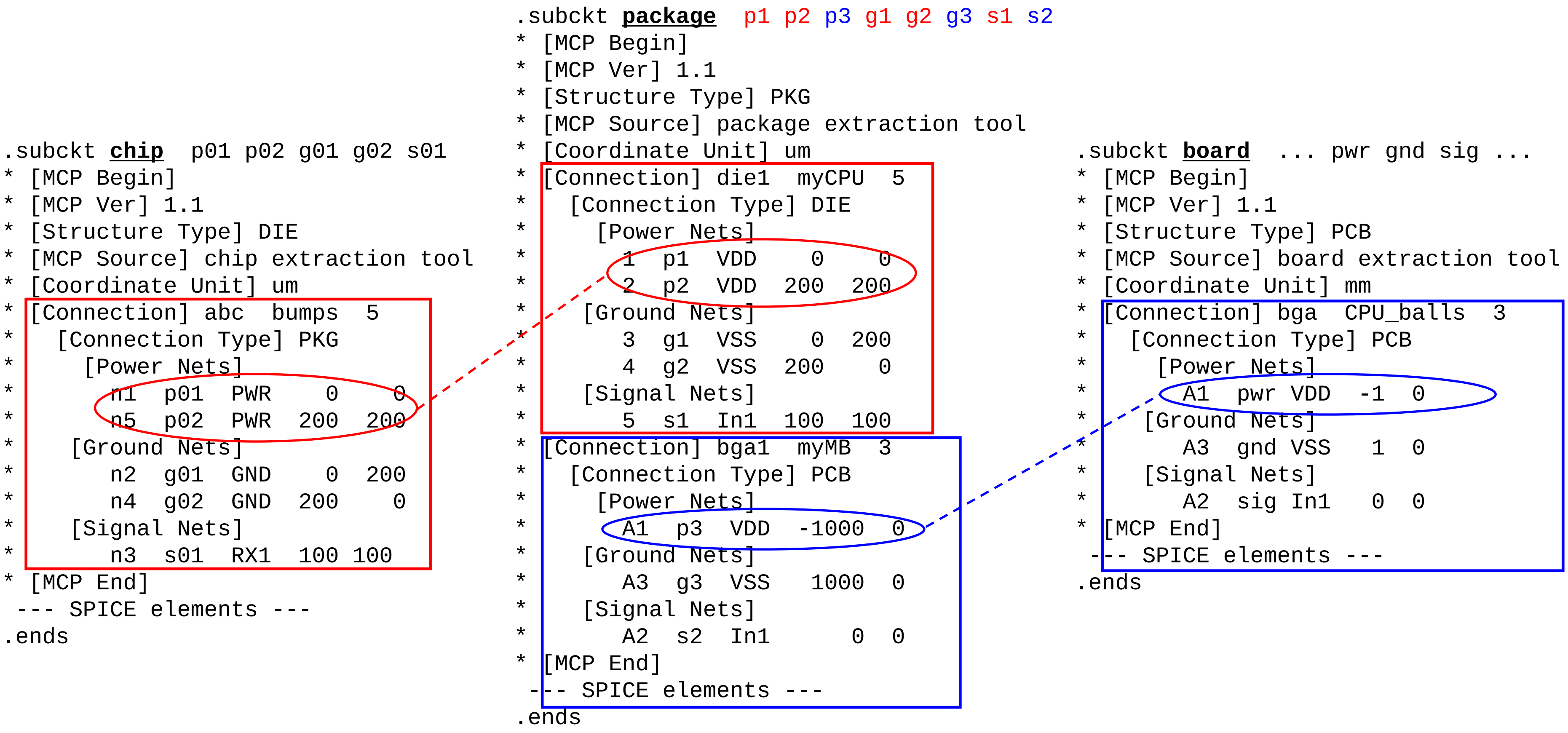
board netlist

An existing MCP

comments in a SPICE netlist

```
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] {DIE|PKG|PCB}
* [MCP Source] source text
* [Coordinate Unit] unit
* [Connection] Name  Description  numberPhysicalPins
*   [Connection Type] {DIE|PKG|PCB}
*     [Power Nets]
*       pinName  modelNodeName  netName  x  y
*       …
*    [Ground Nets]
*       pinName  modelNodeName  netName  x  y
*       …
*     [Signal Nets]
*       pinName  modelNodeName  netName  x  y
*       …
* [MCP End]
```

A Simple Example



```
.subckt chip  p01 p02 g01 g02 s01
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] DIE
* [MCP Source] chip extraction tool
* [Coordinate Unit] um
* [Connection] abc  bumps  5
*   [Connection Type] PKG
*     [Power Nets]
*        n1  p01  PWR    0    0
*        n5  p02  PWR  200  200
*    [Ground Nets]
*        n2  g01  GND    0  200
*        n4  g02  GND  200    0
*    [Signal Nets]
*        n3  s01  RX1  100 100
* [MCP End]
 --- SPICE elements ---
.ends
```

```
.subckt package  p1 p2 p3 g1 g2 g3 s1 s2
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] PKG
* [MCP Source] package extraction tool
* [Coordinate Unit] um
* [Connection] die1  myCPU  5
*   [Connection Type] DIE
*     [Power Nets]
*        1  p1  VDD    0    0
*        2  p2  VDD  200  200
*    [Ground Nets]
*        3  g1  VSS    0  200
*        4  g2  VSS  200    0
*    [Signal Nets]
*        5  s1  In1  100  100
* [Connection] bga1  myMB  3
*   [Connection Type] PCB
*     [Power Nets]
*        A1  p3  VDD  -1000  0
*    [Ground Nets]
*        A3  g3  VSS   1000  0
*    [Signal Nets]
*        A2  s2  In1      0  0
* [MCP End]
 --- SPICE elements ---
.ends
```

```
.subckt board  ... pwr gnd sig ...
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] PCB
* [MCP Source] board extraction tool
* [Coordinate Unit] mm
* [Connection] bga  CPU_balls  3
*   [Connection Type] PCB
*     [Power Nets]
*        A1  pwr VDD  -1  0
*    [Ground Nets]
*        A3  gnd VSS   1  0
*    [Signal Nets]
*        A2  sig In1   0  0
* [MCP End]
 --- SPICE elements ---
.ends
```

what if SPICE netlist supported ‘vector nodes’

```
.subckt chip  abc[]
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] DIE
* [MCP Source] chip extraction tool
* [Coordinate Unit] um
* [Connection] abc  bumps  5
*   [Connection Type] PKG
*     [Power Nets]
*       n1  p01  PWR    0    0
*       n5  p02  PWR  200  200
*    [Ground Nets]
*       n2  g01  GND    0  200
*       n4  g02  GND  200    0
*    [Signal Nets]
*       n3  s01  RX1  100 100
* [MCP End]
 --- SPICE elements ---
.ends
```

```
.subckt package  die1[] bga1[]
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] PKG
* [MCP Source] package extraction tool
* [Coordinate Unit] um
* [Connection] die1  myCPU  5
*   [Connection Type] DIE
*     [Power Nets]
*       1  p1   VDD    0    0
*       2  p2   VDD  200  200
*    [Ground Nets]
*       3  g1   VSS    0  200
*       4  g2   VSS  200    0
*    [Signal Nets]
*       5  s1   In1  100  100
* [Connection] bga1  myMB  3
*   [Connection Type] PCB
*     [Power Nets]
*       A1  p3  VDD  -1000  0
*    [Ground Nets]
*       A3  g3  VSS   1000  0
*    [Signal Nets]
*       A2  s2  In1      0  0
* [MCP End]
 --- SPICE elements ---
.ends
```

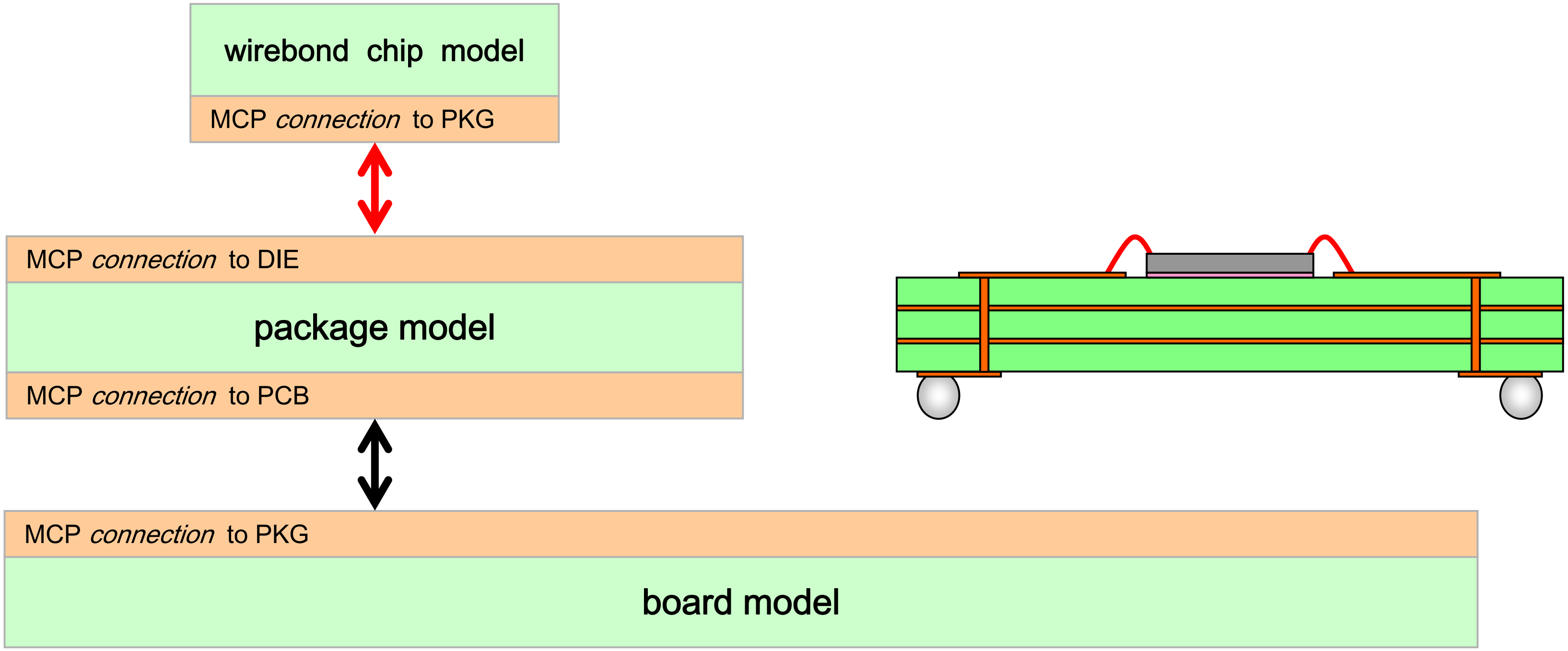
```
.subckt board  ... bga[] ...
* [MCP Begin]
* [MCP Ver] 1.1
* [Structure Type] PCB
* [MCP Source] board extraction tool
* [Coordinate Unit] mm
* [Connection] bga  processor  3
*   [Connection Type] PCB
*     [Power Nets]
*       A1  pwr VDD  -1  0
*    [Ground Nets]
*       A3  gnd VSS   1  0
*    [Signal Nets]
*       A2  sig In1   0  0
* [MCP End]
 --- SPICE elements ---
.ends
```

then we could easily use a netlist for system-level analysis setup

```
...
x1 con1[] chip
x2 con1[] con2[] package
x3 con2[] board
R board.con2[A1] board.con2[A3] 5m
...
```

Simple system
with MCP based connectivity

- MCP defines both electrical and physical connectivity
  - pin or net names may or not be the same across domain boundaries
  - pin locations may be translated, rotated or flipped between physical databases



wirebond chip model

MCP *connection* to PKG

MCP *connection* to DIE

package model

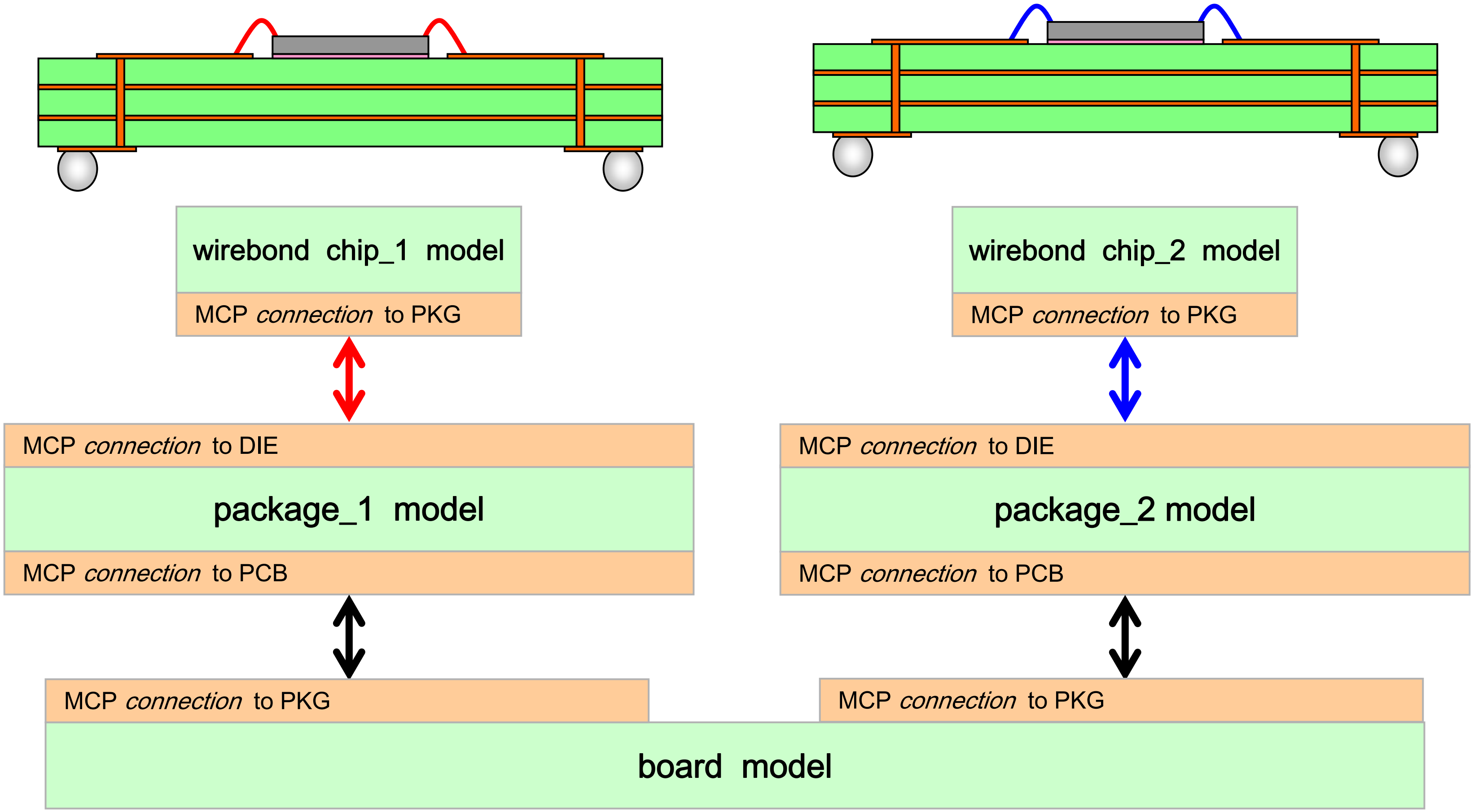
MCP *connection* to PCB

MCP *connection* to PKG

board model

MCP based connectivity for a 2 packaged chips

- 2 packaged chips on a board



wirebond chip_1 model

MCP *connection* to PKG

MCP *connection* to DIE

package_1 model

MCP *connection* to PCB

wirebond chip_2 model

MCP *connection* to PKG

MCP *connection* to DIE

package_2 model

MCP *connection* to PCB

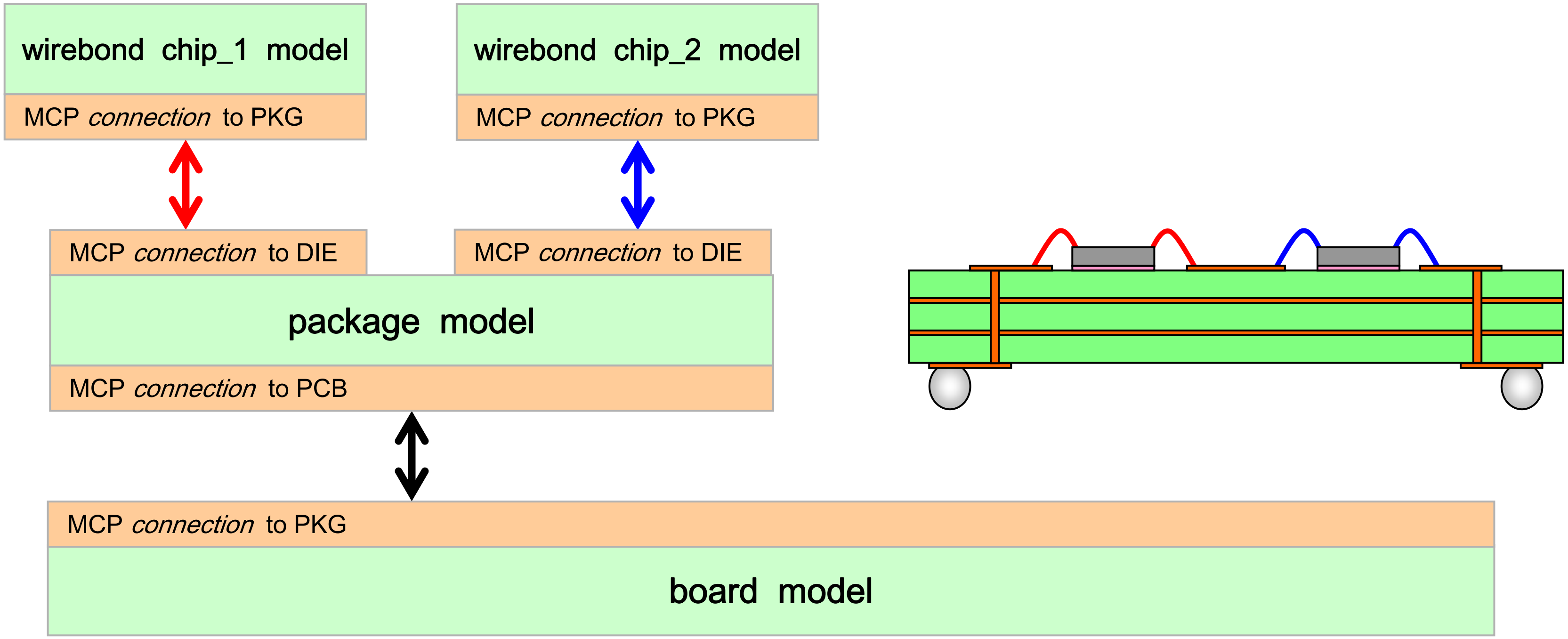
MCP *connection* to PKG

MCP *connection* to PKG

board model

MCP based connectivity for 2-die SIP

- A 2-die SIP
  - each chip has connectivity to the package but not directly to the other chip



wirebond chip_1 model

MCP *connection* to PKG

wirebond chip_2 model

MCP *connection* to PKG

MCP *connection* to DIE

MCP *connection* to DIE

package model

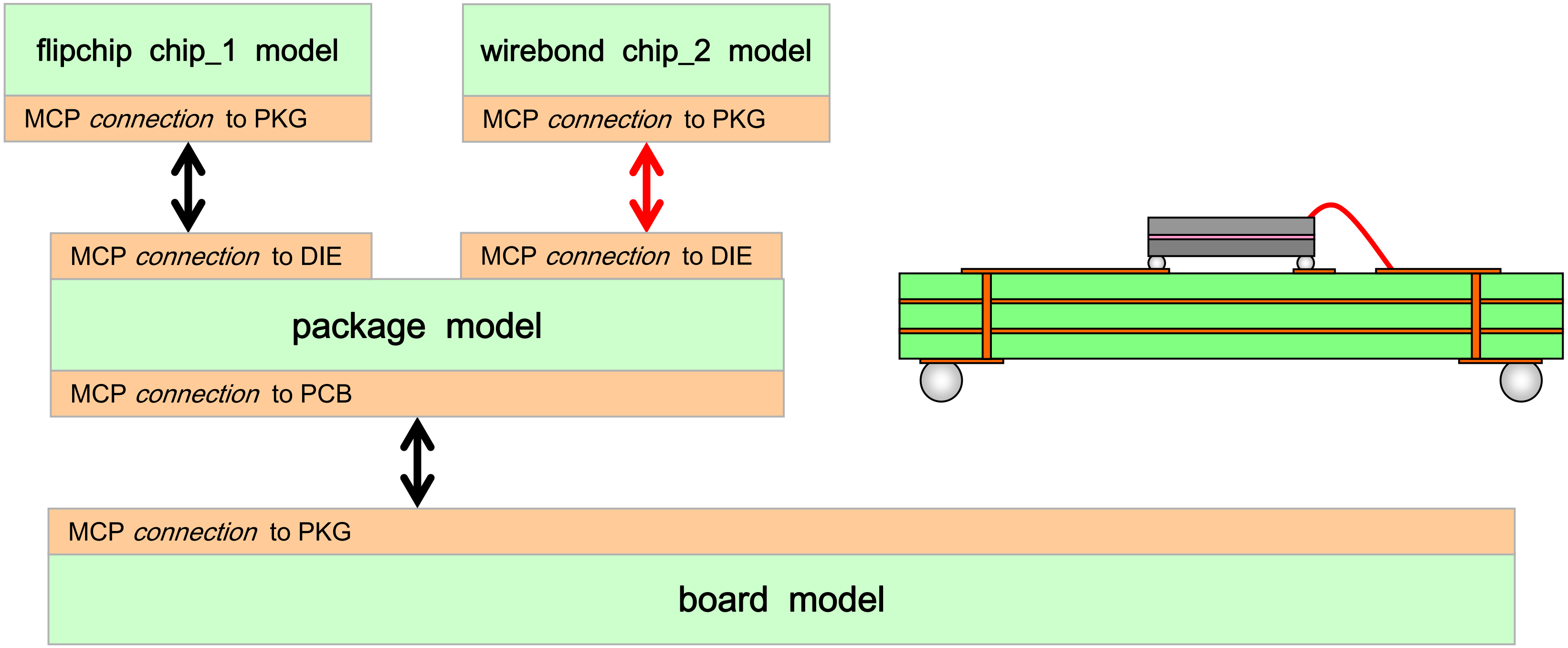
MCP *connection* to PCB

MCP *connection* to PKG

board model

MCP based connectivity for 2 stacked die

- 2 stacked die in a package (one flipchip, one wirebond)
  - again, no direct die-to-die physical connections



flipchip chip_1 model

MCP *connection* to PKG

wirebond chip_2 model

MCP *connection* to PKG

MCP *connection* to DIE

MCP *connection* to DIE

package model

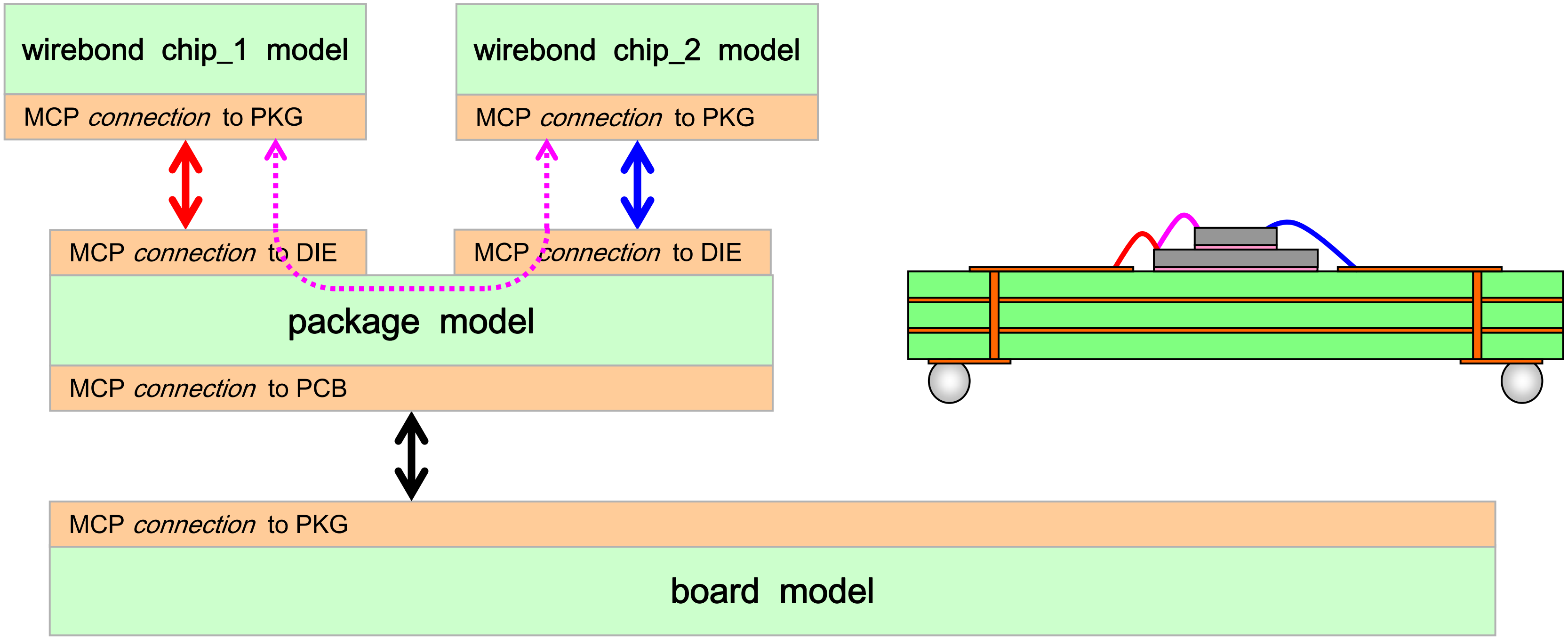
MCP *connection* to PCB

MCP *connection* to PKG

board model

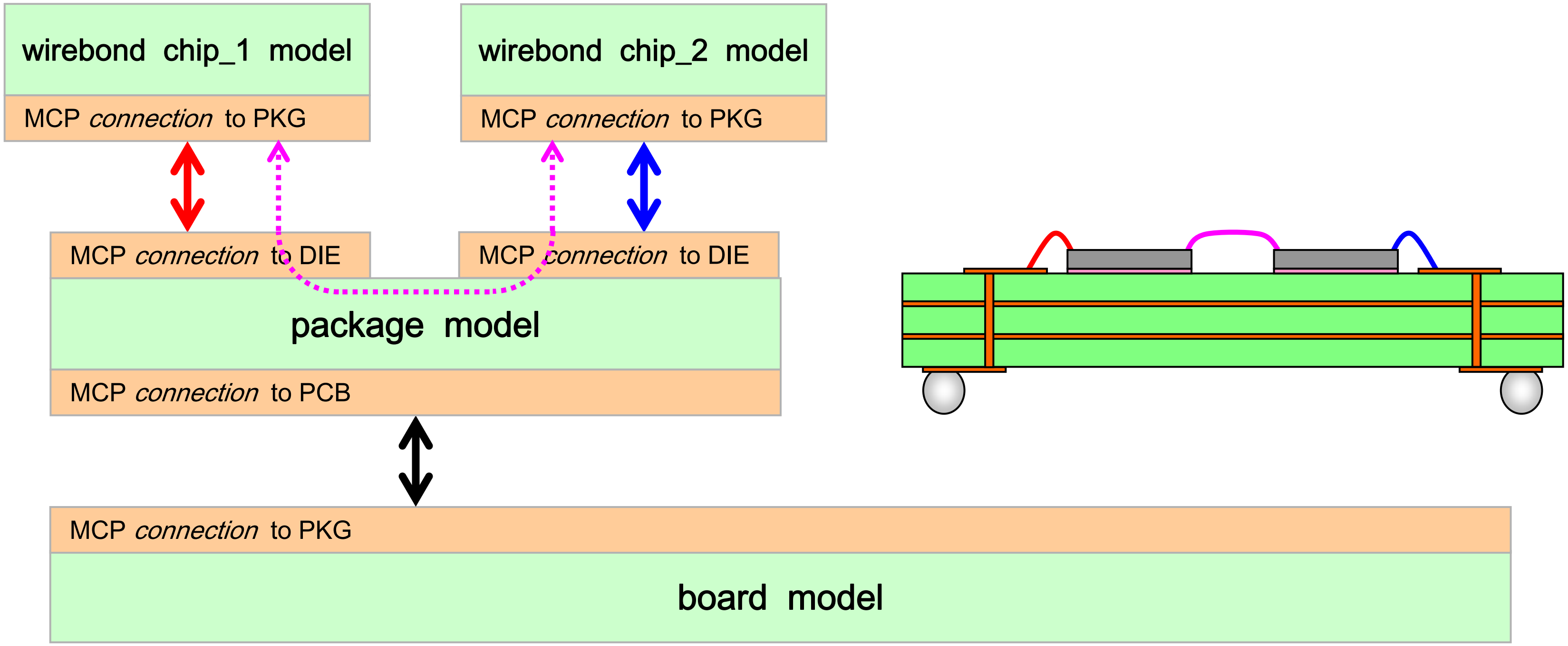
MCP based connectivity for 2 stacked die with die-to-die wirebonds

- 2 stacked die in a package (both wirebond)
  - direct die-to-die physical connections
    - whose parasitics are included as part of the “package” model
    - potentially confusing at first but logical when considering all die connections as a SiP package with a single model
    - avoids application-dependent die MCP definitions



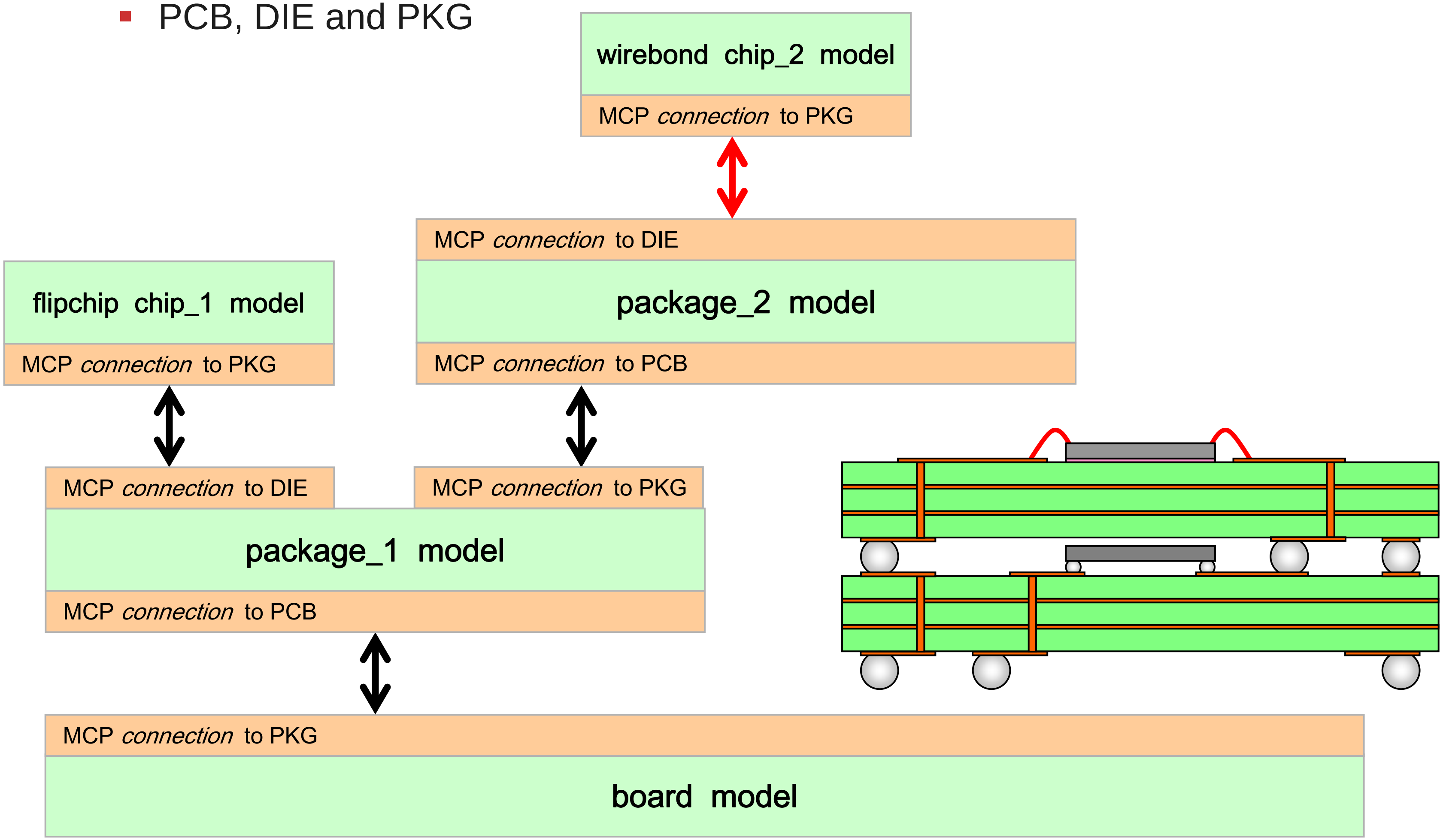
MCP based connectivity for 2-die SIP with die-to-die wirebonds

- A 2-die SIP
  - with direct die-to-die physical connections
  - again, whose parasitics are included in the package model



MCP based connectivity for Package-on-Package

- 2 stacked packages, each single die
  - stackable package has 3 MCP connections
    - PCB, DIE and PKG


wirebond chip_2 model

MCP *connection* to PKG

MCP *connection* to DIE

package_2 model

MCP *connection* to PCB

flipchip chip_1 model

MCP *connection* to PKG

MCP *connection* to DIE

MCP *connection* to PKG

package_1 model

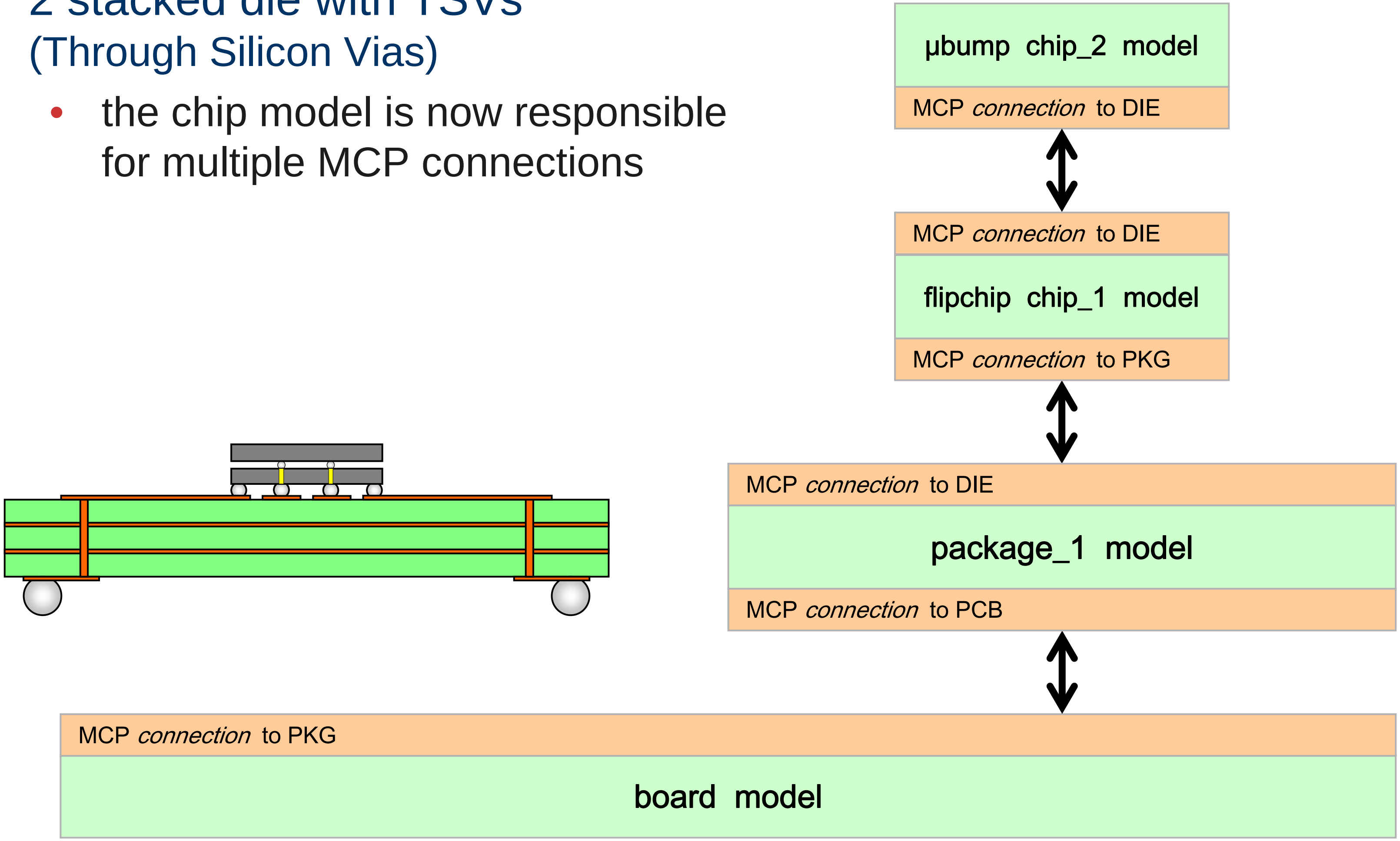
MCP *connection* to PCB

MCP *connection* to PKG

board model

MCP based connectivity for 2 stacked die with TSVs

- 2 stacked die with TSVs (Through Silicon Vias)
  - the chip model is now responsible for multiple MCP connections



μbump chip_2 model

MCP *connection* to DIE

MCP *connection* to DIE

flipchip chip_1 model

MCP *connection* to PKG

MCP *connection* to DIE

package_1 model

MCP *connection* to PCB

MCP *connection* to PKG

board model

IBIS-MCP Status

- MCP is quite general for chip-package-board systems
  - chip-centric design is supported with readily available and easily connected package/board models
  - system-centric design is supported with readily available and easily connected chip models
- Other applications may require additions or modifications to the initial proposal ???
- Discussions are schedule to begin on this topic in the IBIS Interconnect Task Group in the next few weeks
  - please join the Task Group discussions and help make this a more robust proposal
  - meetings are Wednesday mornings at 9:00am pacific time

Thank You!