UCLA COMPUTER SCIENCE DEPARTMENT

A Multilevel Analytical Placement for 3D ICs

Jason Cong[1,2] & **Guojie Luo**[1]

[1]UCLA Computer Science Department

[2]California NanoSystems Institute

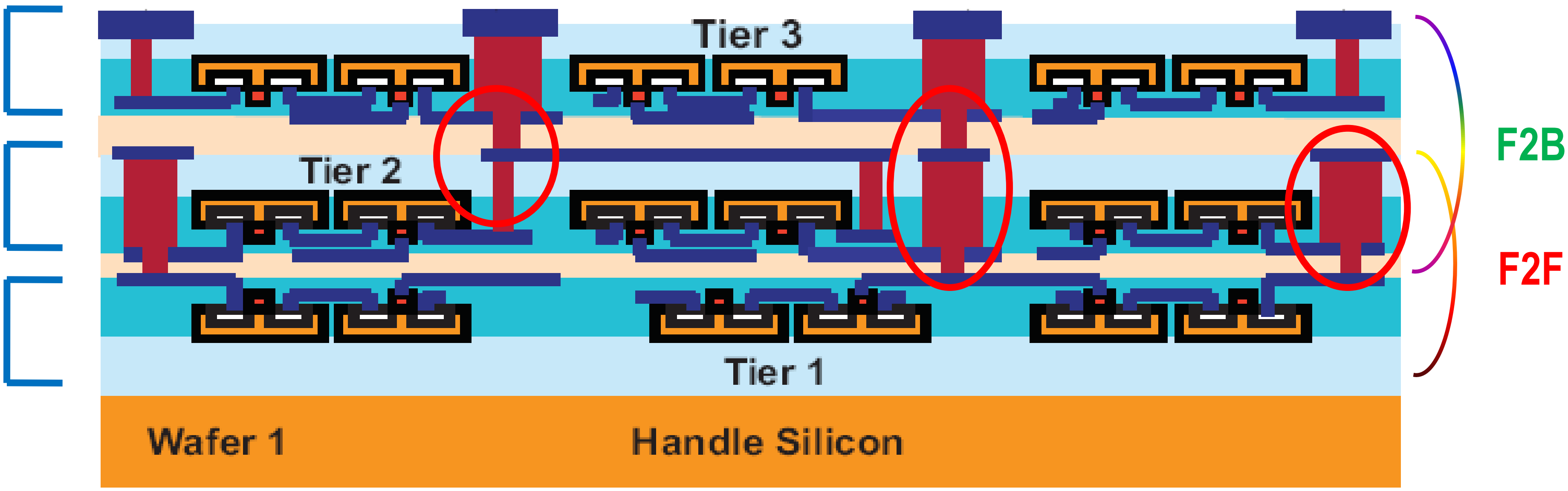
{ cong, gluo } @ cs.ucla.edu

This work is partially supported by

IBM (under a DARPA subcontract) and NSF

3D Integration

- Example (MIT Lincoln Lab 180nm SOI technology)
  - A collection of *tiers*
  - Through-silicon via (*TSV*)



Basic 3D Placement Problem

- **Variables**
  - (x_i, y_i, z_i), $i = 1, 2, \ldots, n$
  - cell i is placed at (x_i, y_i) on the tier z_i
- **Objective**
  - $\sum_e WL_e(x,y,z) = HPWL_{(x,y)} + \alpha_{TSV} HPWL_z$
  - To minimize weighted wirelength
- **Constraint**
  - no overlap between cells

Previous Works on 3D Placement

- **Force-directed method**
  - [Goplen & Sapatnekar, ICCAD'03]
- **Partitioning-based method**
  - [Goplen & Sapatnekar, DAC'07]
- **Quadratic modeling of density cost through DCT**
  - [Yan et al., Integration'09]
- **2D to 3D transformation method**
  - [Cong et al., ASPDAC'07]

Motivations

- 3D placement tool
  - Trade-offs between wirelength and TSV
  - Flexible to integrate other objective function and constraints
  - High-quality and scalable
- To study analytical placement

Our Contributions

- Analytical formulation with a novel density penalty function
  - Based on multiple-tier 2D density penalty functions
  - Introduce pseudo-layers, so that minimization of penalties on tiers and pseudo tiers guarantees a legal 3D placement
- Adaption of multilevel method
  - Provides extra TSV reduction in addition to increasing the TSV weight
- Improvements compared to 2D to 3D transformation
  - (best wirelength cases) 2% shorter wirelength and 29% fewer TSV
  - (best TS via cases) 20% shorter wirelength and 50% fewer TSV

Basic 3D Placement Problem

- **Variables**
  - (x_i,y_i,z_i), $i=1,2,\ldots,n$
  - cell i is placed at (x_i,y_i) on the tier z_i
- **Objective**
  - $\sum_e WL_e(x,y,z) = HPWL_{(x,y)} + \alpha_{TSV} HPWL_z$
  - To minimize weighted wirelength
- **Constraint**
  - no overlap between cells

Weighted Wirelength

- $WL_e(x,y,z) = \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j|$

Weighted Wirelength

- $$WL_e(x,y,z) = \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right)$$

2D HPWL

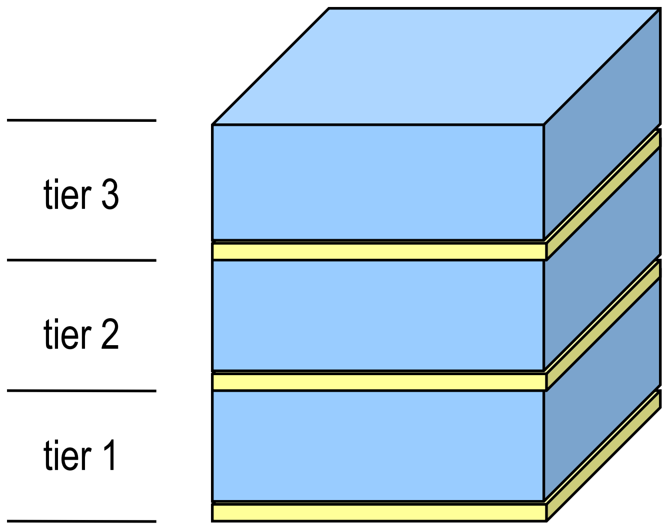
Weighted Wirelength

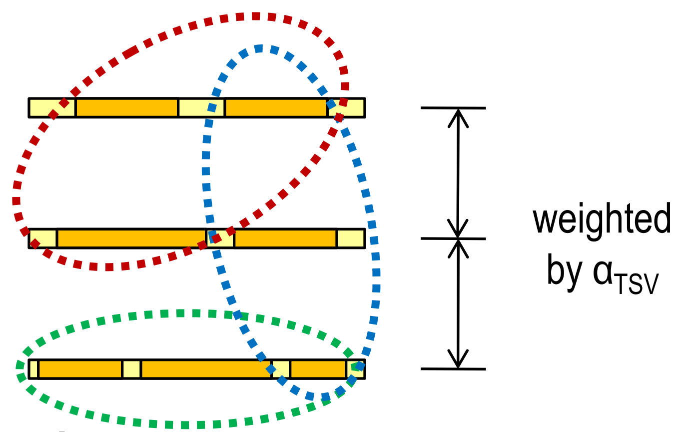
- $$WL_e(x,y,z) = \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j|$$

3D
Weighted
HPWL

Weighted Wirelength

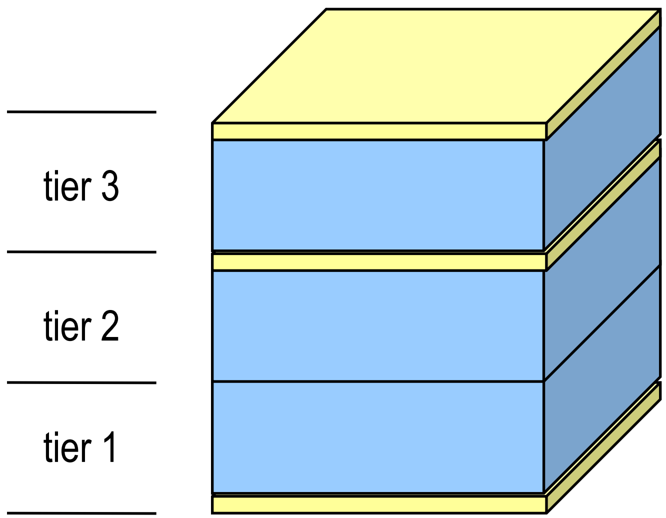
- $$WL_e(x,y,z) = \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j|$$
- **Model TSV by a length of wire**
  - For example [Davis et al., DTC'05]
    - MIT Lincoln Lab 180 nm 3D SOI technology
    - 3 μm thick TSV ≈ 8 to 20 μm metal 2 wire, in terms of capacitance
    - 3 μm thick TSV ≈ 0.2 μm metal 2 wire, in terms of resistance

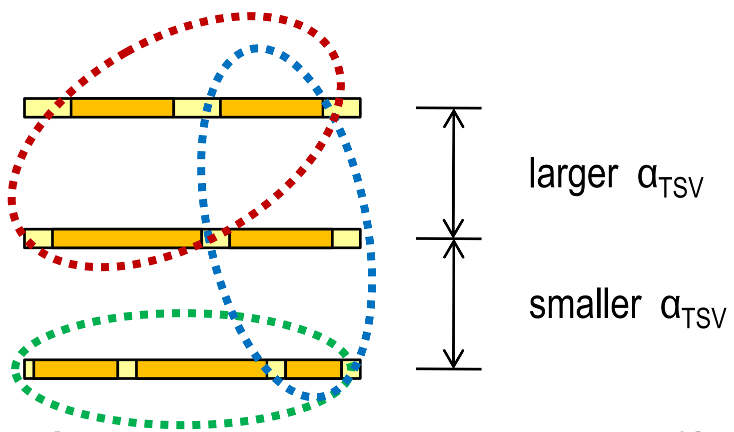




Weighted Wirelength

- Another case
  - Tier 1 and tier 2: face-to-face
  - Tier 2 and tier 3: back-to-back
- Different weights between tiers





Weighted Wirelength

- **Practical weighed wirelength**

$$WL_e(x,y,z) = (1+p_e)\left(\max_{v_i,v_j \in e} \left| x_i - x_j \right| + \max_{v_i,v_j \in e} \left| y_i - y_j \right| \right)$$

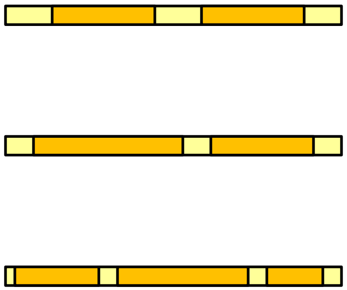
$$+(1+q_e)\cdot \alpha_{TSV} \cdot \max_{v_i,v_j \in e} \left| z_i - z_j \right|$$

  - Additional net weights p_e and q_e to model and optimize performance or temperature [Goplen & Sapatnekar, DAC'07]
  - It is a convex function w.r.t. (x,y,z)

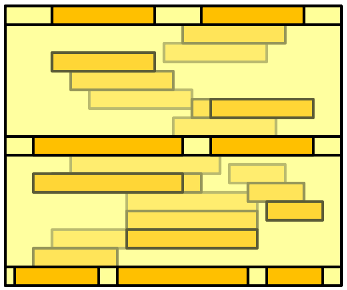
- **Such weighted wirelength is the form of objective function in the 3D placement problem formulation**

Analytical Engine

- Discrete tier assignment
- Relaxed tier assignment



discrete
(legalized solution)

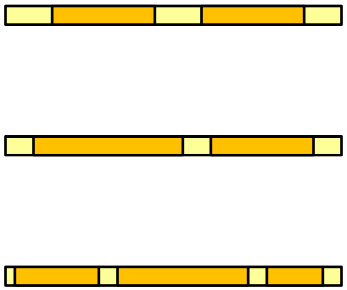


relaxed
(intermediate solution)

- Variables
  - (x_i, y_i, z_i), i=1,2,…,n
  - cell i is placed at (x_i, y_i) on the tier z_i

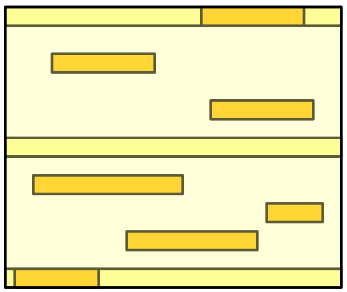
Analytical Engine

- Discrete tier assignment



discrete
(legalized solution)

- Relaxed tier assignment



relaxed
(intermediate solution)

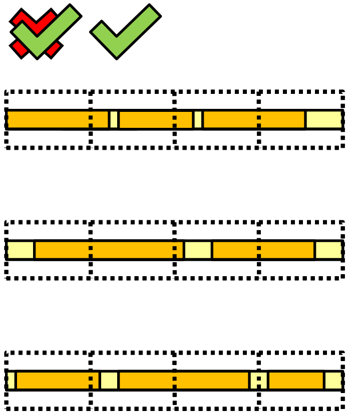
- Formulate 3D placement problem as continuous optimization

$$\text{minimize} \quad \sum_e WL_e(x, y, z)$$

$$\text{subject to} \quad \text{(no overlap between cells)}$$

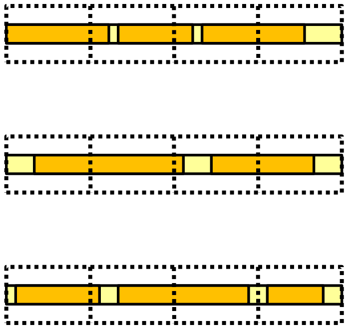
Non-overlap Constraints

- Relaxed by area density constraints
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
    - Cell overlaps in overflow bins violate density constraints
    - Cell overlaps not in overflow bins do not violate density constraints



Non-overlap Constraint

- **Replaced by area density constraint**
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps



minimize $\sum_e WL_e(x,y,z)$

subject to (no overlap between cells)

$\Downarrow$

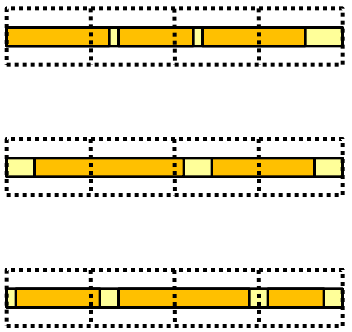
minimize $\sum_e WL_e(x,y,z)$

subject to $A_{i,j,k}(x,y,z) \le C_{i,j,k}$

for all i,j,k

Non-overlap Constraint

- **Replaced by area density constraint**
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps



$$\text{minimize} \quad \sum_e WL_e(x,y,z)$$

$$\text{subject to} \quad \text{(no overlap between cells)}$$

$$\Downarrow$$

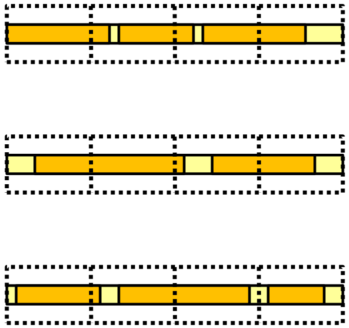
$$\text{minimize} \quad \sum_e WL_e(x,y,z)$$

$$\text{subject to} \quad A_{i,j,k}(x,y,z) = C_{i,j,k} \quad \text{for all } i,j,k$$

add filler cells
[Chan et al., ISPD'06]

Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps



$$\text{minimize} \quad \sum_e WL_e(x,y,z)$$

$$\text{subject to} \quad A_{i,j,k}(x,y,z) = C_{i,j,k} \quad \text{for all } i, j, k$$

$$\Downarrow$$

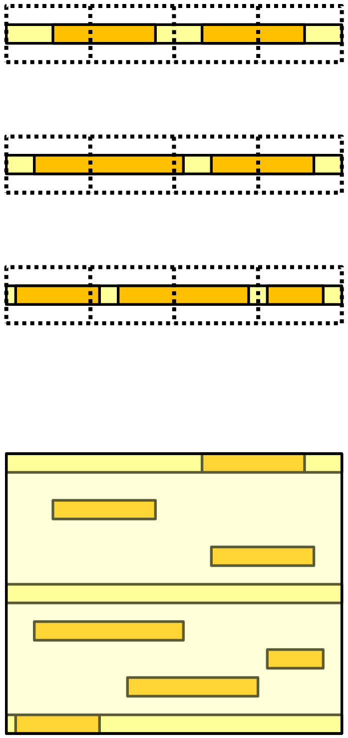
$$\text{minimize} \quad \sum_e WL_e(x,y,z) + \frac{\mu}{2}\sum_k \sum_{i,j} (A_{i,j,k}(x,y,z) - C_{i,j,k})^2$$

increase μ until overlaps are removed

[Nam & Cong, Springer'07]
[Cong & Luo, ISPD'08]

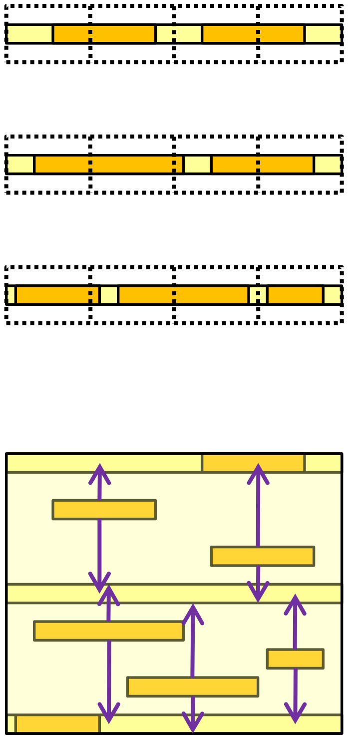
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
- Area projection to obtain bin densities from intermediate solution



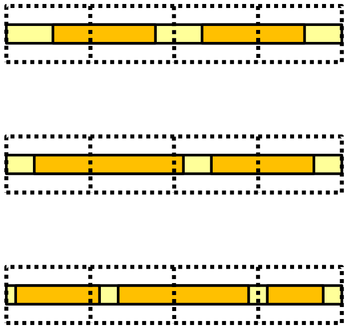
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
- Area projection to obtain bin densities from intermediate solution

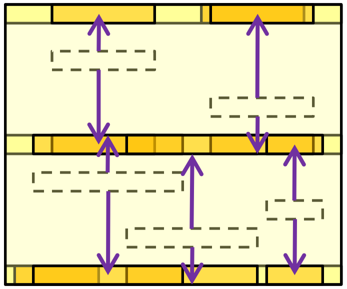


Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

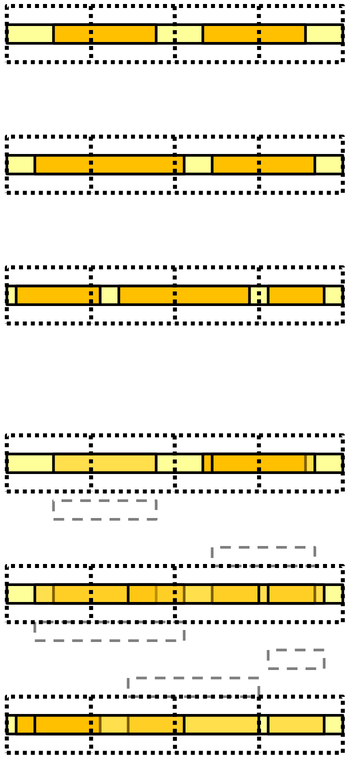


- Area projection to obtain bin densities from intermediate solution



Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
- Area projection to obtain bin densities from intermediate solution



Area Projection

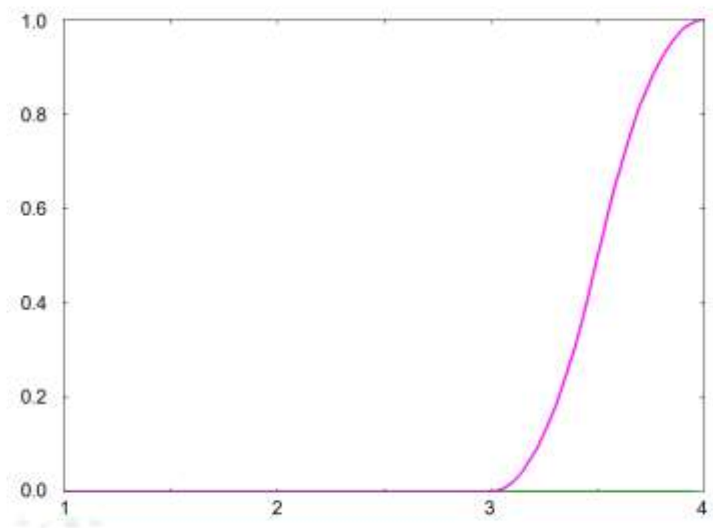
- **Bell-shaped function to project area**

$$\eta(k,z)=\begin{cases}1-2(z-k)^2 & |z-k|\le 1/2\\ 2(|z-k|-1)^2 & 1/2<|z-k|\le 1\\ 0 & \text{otherwise}\end{cases}$$

$\eta(k,z)$ - The projection ratio from "tier z" to tier k

$$Ai,j,k(x,y,z)$$
$$=\sum_{v\in V}A_{i,j}(x_v,y_v)\cdot\eta(k,z_v)$$

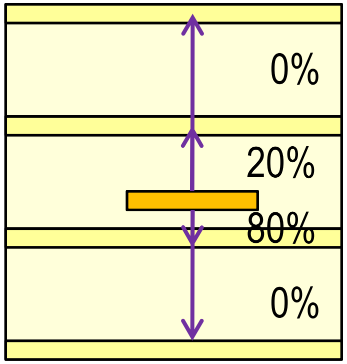
$\eta(1,z)$ $\eta(2,z)$ $\eta(3,z)$ $\eta(4,z)$



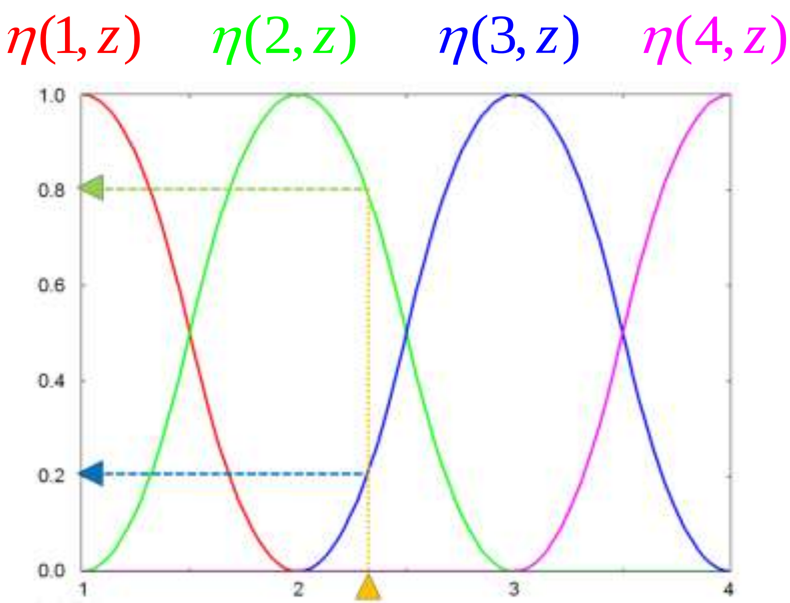
Area Projection

- **Bell-shaped function to project area**

$$\eta(k,z)=\begin{cases}1-2(z-k)^2 & |z-k|\le 1/2\\ 2(|z-k|-1)^2 & 1/2<|z-k|\le 1\\ 0 & \text{otherwise}\end{cases}$$



- **An Example**
  - Intermediate placement of a cell at “tier 2.316”
  - Projects 0% area to tier 1
  - Projects 80% area to tier 2
  - Projects 20% area to tier 3
  - Projects 0% area to tier 4



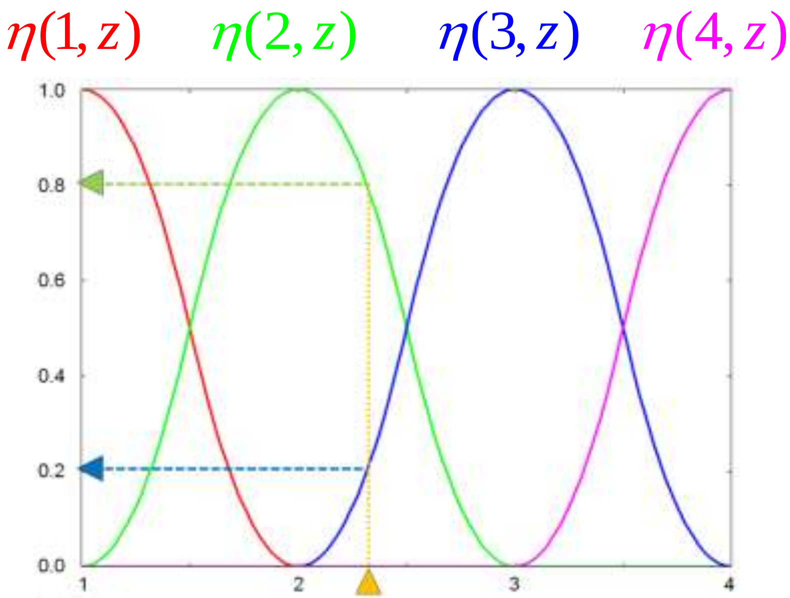
Area Projection

- **Bell-shaped function to project area**

$$\eta(k,z)=\begin{cases}1-2(z-k)^2 & |z-k|\le 1/2\\ 2(|z-k|-1)^2 & 1/2<|z-k|\le 1\\ 0 & \text{otherwise}\end{cases}$$

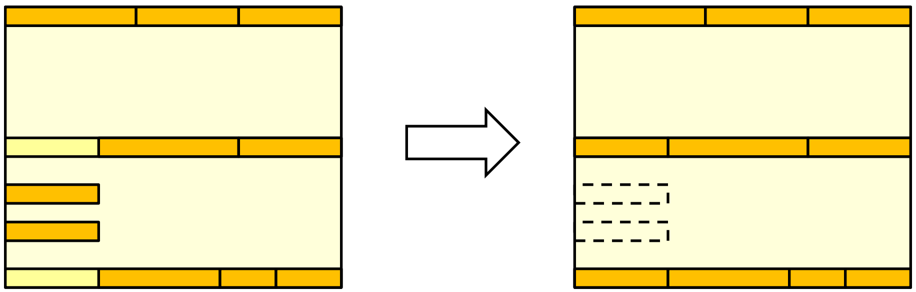
η(k,z) - The projection ratio from "tier z" to tier k

- **An Example**
  - Intermediate placement of a cell at "tier 2.316"
  - Projects 0% area to tier 1
  - Projects 80% area to tier 2
  - Projects 20% area to tier 3
  - Projects 0% area to tier 4

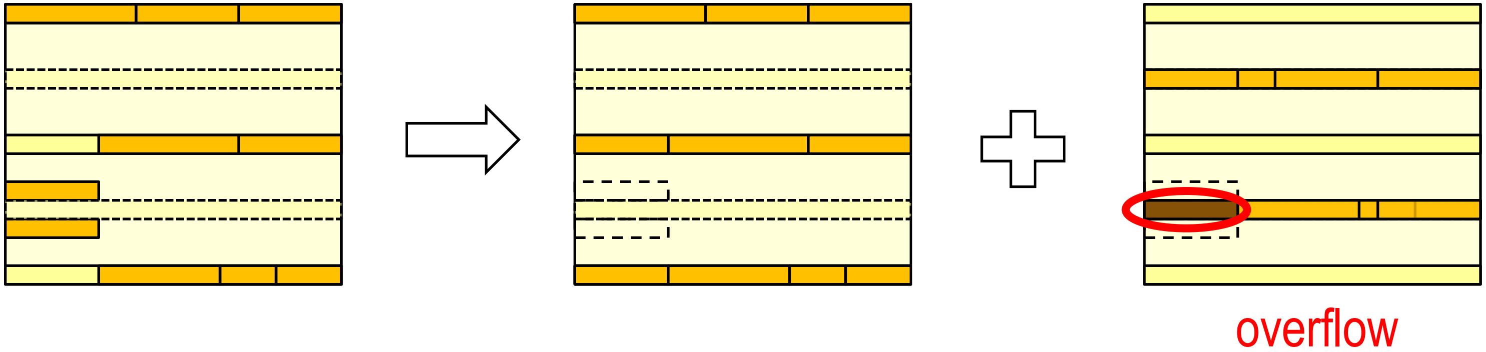


Equivalence to Non-overlap Constraint

- Area projection to tiers is not enough
  - Counter example: projected area failed to capture illegality



- Solution: area projection on pseudo-tiers



Equivalence to Non-overlap Constraint

- **Theorem: (x,y,z) satisfy the constraints**

$$\begin{cases} A_{i,j,k}(x,y,z) = C_{i,j,k} \\ A'_{i,j,k}(x,y,z) = C'_{i,j,k} \end{cases} \quad \text{for all } i,j,k$$

if.f. (x,y,z) is a legal placement (no overlaps)

** after adding filler cells

Equivalence to Non-overlap Constraint

- **Theorem: (x,y,z) is a minimizer of the function:**

$$\frac{\mu}{2}\sum_k \sum_{i,j}(A_{i,j,k}(x,y,z)-C_{i,j,k})^2$$

$$+\quad \frac{\mu}{2}\sum_k \sum_{i,j}(A'_{i,j,k}(x,y,z)-C'_{i,j,k})^2$$

if.f. (x,y,z) is a legal placement (no overlaps)

** after adding filler cells

Analytical Engine

- minimize $\sum_e WL_e(x,y,z)$

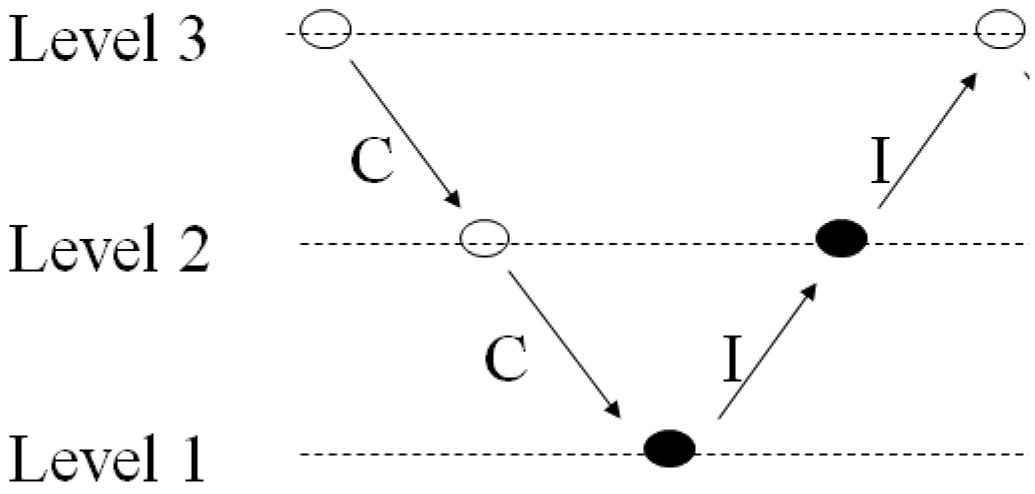
$$+\frac{\mu}{2}\sum_k \sum_{i,j} (A_{i,j,k}(x,y,z) - C_{i,j,k})^2$$

$$+\frac{\mu}{2}\sum_k \sum_{i,j} (A'_{i,j,k}(x,y,z) - C'_{i,j,k})^2$$

increase μ until overlaps are removed

- $A_{i,j,k}(x,y,z)$: **area projected in bin *(i,j)* of tier *k***
- $C_{i,j,k}$: **area capacitance on tier *k***
- $A'_{i,j,k}(x,y,z)$: **area projected in bin *(i,j)* of pseudo-tier *k***
- $C'_{i,j,k}$: **area capacitance on pseudo-tier *k***

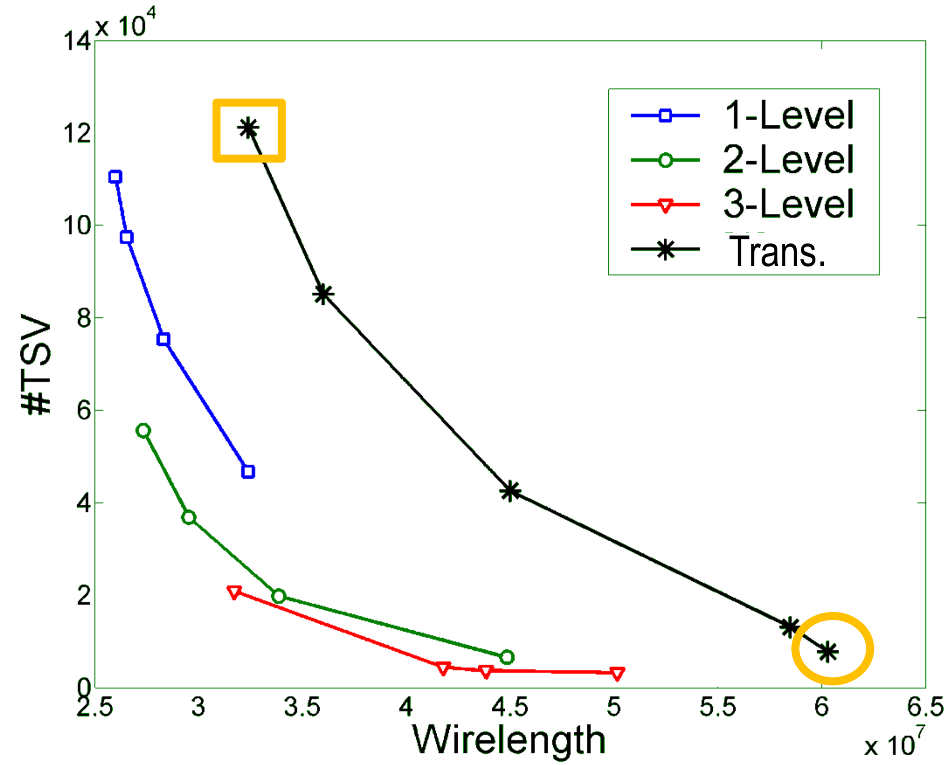
Multilevel Framework



- Level at which analytical engine is applied
- C Coarsening
- I Interpolation

Experimental Results (1/2)

- **Comparison of trade-off curves (ibm13)**
  - 19% shorter WL
    9% fewer TSV
    than □
  - 15% shorter WL
    43% fewer TSV
    than ○



(consistent behavior on other circuits)

Experimental Results (2/2)

- The ability to reduce the TSV number

Circuit	3-Level Placement			4-way Mincut	
	GP WL ($\times 10^7$)	DP WL ($\times 10^7$)	#TSV ($\times 10^3$)	cutsize ($\times 10^3$)	#TSV ($\times 10^3$)
ibm01	0.39	0.39	0.92	0.35	0.42
ibm03	0.92	0.91	2.10	1.28	2.02
ibm04	1.36	1.31	2.01	1.41	1.89
ibm06	1.67	1.62	2.60	1.63	2.63
ibm07	2.79	2.70	2.72	2.13	2.97
ibm08	2.99	2.89	2.83	2.02	2.60
ibm09	2.36	2.29	2.47	1.35	1.90
ibm13	5.02	4.89	3.20	1.62	2.21
ibm15	12.05	11.40	8.27	4.20	6.19
ibm18	18.36	17.37	9.82	2.95	4.68
geo-mean	2.66	2.58	2.95	1.61	2.29

Summary

- Non-overlap constraints
  - Handled by a novel area projection method
  - Pseudo-tiers added for equivalence to non-overlap constraints
- Multilevel framework
  - Effective to reduce TS via number
- Trade-offs between WL and #TSV
  - 12% shorter WL and 29% fewer TSV
    - Compared to the 2D to 3D transformation method with best WL
  - 20% shorter WL and 50% fewer TSV
    - Compared to the 2D to 3D transformation method with best TSV

Thank you!