

Multi-Level Partitioning  
Methodologies and their  
applications in modern IC design

George Goudroumanis (2869)  
Sotiriou Christos, Plessas Fotios, Stamoulis Georgios

1



Μεθοδολογίες πολυεπίπεδης κατάτμησης κυκλωμάτων και  
οι εφαρμογές τους στη σχεδίαση σύγχρονων  
ολοκληρωμένων κυκλωμάτων

Γιώργος Γκουντρούμανης (2869)  
Σωτηρίου Χρήστος, Πλέσσας Φώτιος, Σταμούλης Γεώργιος

2

## Thesis Outline

1. Definitions
2. Applications
3. Existing tools
4. Our contribution
5. Evaluation
6. Conclusions
7. Future work





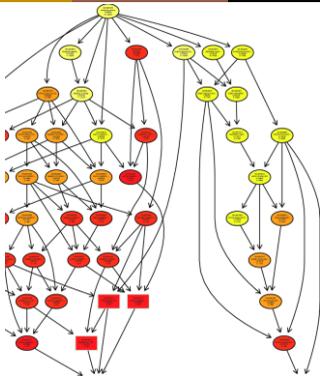




▶ 3
UTH Thesis
31/3/2024

3

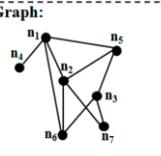
## Definitions (1): Graph Representations



▶ **Directed Graph**

- ▶ G: Set whose elements are called vertices
- ▶ E: Set of pairs of vertices, called directed edges

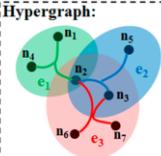
**Graph:**



	n <sub>1</sub>	n <sub>2</sub>	n <sub>3</sub>	n <sub>4</sub>	n <sub>5</sub>	n <sub>6</sub>	n <sub>7</sub>
n <sub>1</sub>	0	1	0	1	1	1	0
n <sub>2</sub>	1	0	0	0	1	1	1
n <sub>3</sub>	0	0	0	0	1	1	1
n <sub>4</sub>	1	0	0	0	0	0	0
n <sub>5</sub>	1	1	1	0	0	0	0
n <sub>6</sub>	1	1	1	0	0	0	0
n <sub>7</sub>	0	1	1	0	0	0	0

adjacency matrix

**Hypergraph:**



	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>
n <sub>1</sub>	1	0	0
n <sub>2</sub>	1	1	1
n <sub>3</sub>	0	1	1
n <sub>4</sub>	1	0	0
n <sub>5</sub>	0	1	0
n <sub>6</sub>	0	0	1
n <sub>7</sub>	0	0	1

incidence matrix

▶ **Hypergraph**

- ▶ G: Set of elements called nodes, vertices
- ▶ E: Set of pairs of subsets of G called an edge or hyperedge

▶ 4
UTH Thesis
31/3/2024

4

## Definitions (2): Physical Design oriented notions

- ▶ **Fan-out & Fan-in**
  - ▶ The number of gate inputs driven by the output of another single logic gate
  - ▶ The number of inputs a logic gate can handle
- ▶ **Path**
  - ▶ The set of continuously connected gates
- ▶ **Half Perimeter Wire Length (HPWL)**
  - ▶ The height and width sum of the net Bounding Box
- ▶ **Design Rules Constraints (DRCs)**
  - ▶ A set of geometric constraints imposed on the IC to ensure that the designs function properly, reliably, and can be produced with acceptable yield

▶ 5
UTH Thesis
31/3/2024

5

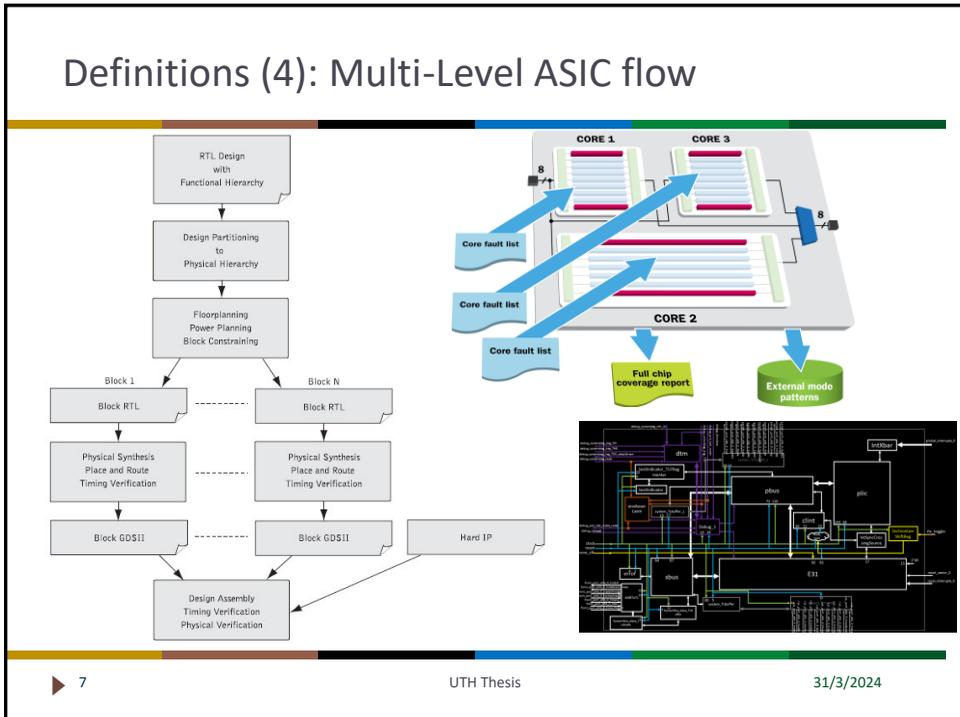
## Definitions (3): Static Timing Analysis oriented notions

- ▶ **Total Negative Slack (TNS)**
  - ▶ The sum of the (real) negative slack in your design.
- ▶ **Worst Negative Slack (WNS)**
  - ▶ Identifies the most critical timing violation within a circuit
- ▶ **Gate Delay**
  - ▶ the time taken by a logic gate to process an input signal and produce the corresponding output
- ▶ **Longest Path Delay**
  - ▶ Is the delay along the most extended path in a digital circuit

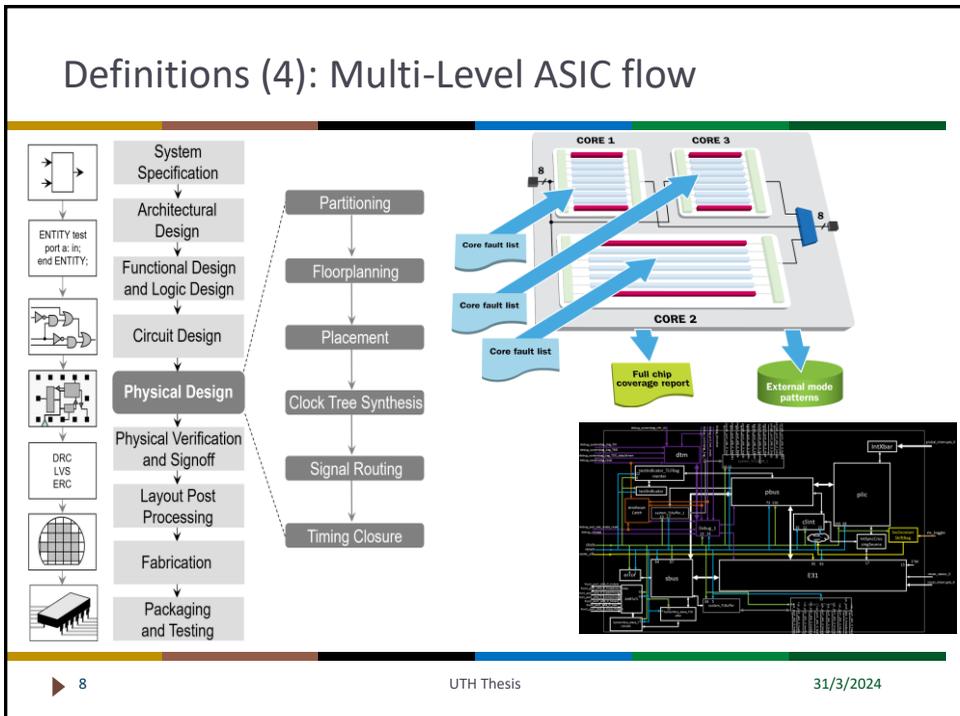
**Critical Path and Slack Time**

▶ 6
UTH Thesis
31/3/2024

6



7



8

## Definitions (5): Clustering

```

SIMPLEHAC( $d_1, \dots, d_N$ )
1 for  $i \leftarrow 1$  to  $N$ 
2 do for  $j \leftarrow 1$  to  $N$ 
3   do  $C[i][j] \leftarrow \text{SIM}$ 
4    $I[i] \leftarrow 1$  (keeps track)
5  $A \leftarrow \{\}$  (assembles clusters)
6 for  $k \leftarrow 1$  to  $N-1$ 
7 do  $(i, m) \leftarrow \text{arg max}_x$ 
8    $A.\text{APPEND}(i, m)$ 
9   for  $j \leftarrow 1$  to  $N$ 
10    do  $C[j][i] \leftarrow \text{SIM}(C[j][m])$ 
11    do  $C[j][m] \leftarrow \text{SIM}(C[j][i])$ 
12     $I[m] \leftarrow 0$  (deactivates)
13 return  $A$ 
        
```

ve  
1/3/2024

9

## Definitions (6): Multi-Level Partitioning

Recursive Bipartitioning

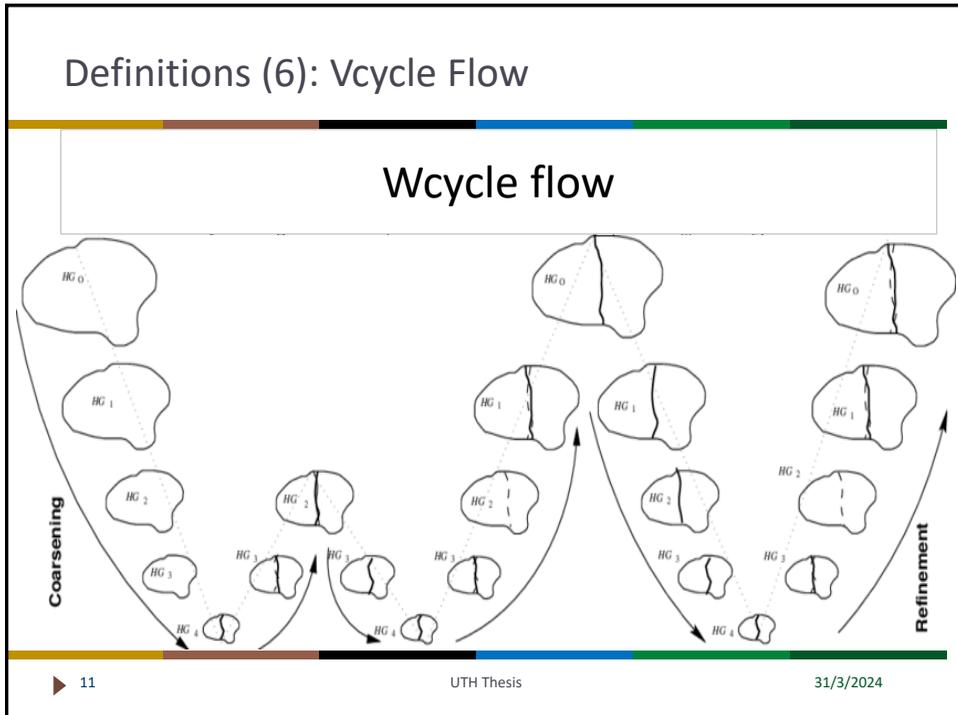
Kway Partitioning

**GAP Model**  
**Machine Learning Methods**

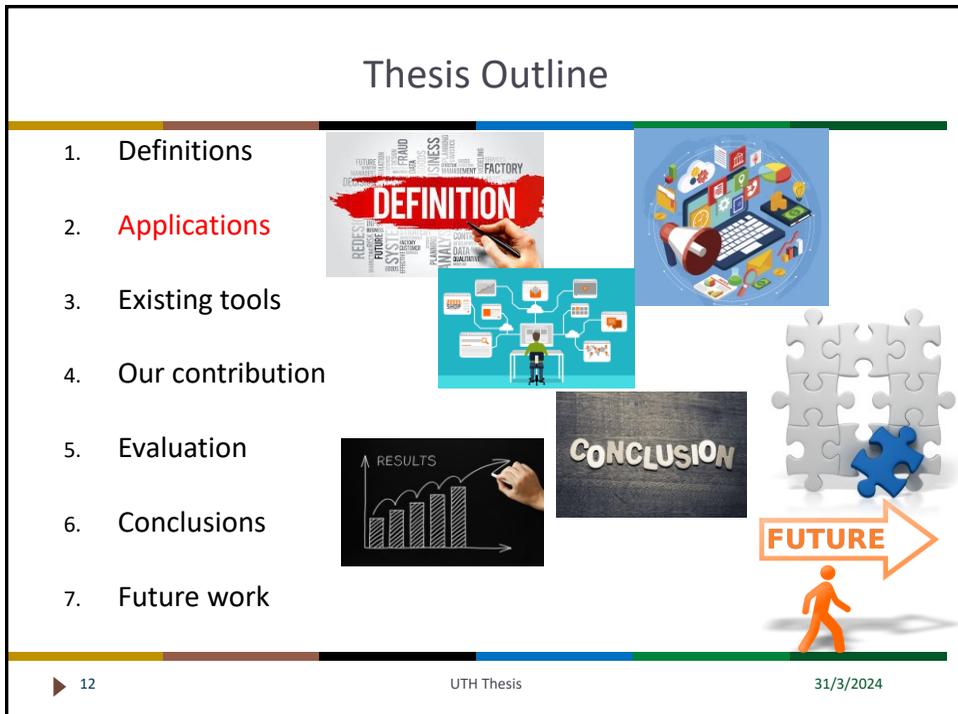
**Spectral Partitioning**

10  
UTH Thesis  
31/3/2024

10



11



12

## Applications: Multi-Level Placement

Flat Placement



Partitioned Placement



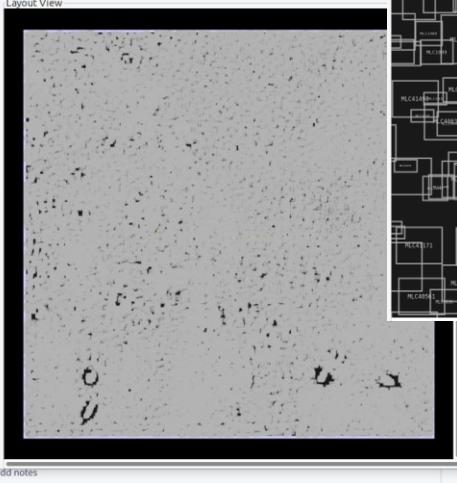
▶ 13UTH Thesis31/3/2024

13

## Applications: Multi-Level Placement

Hieri  
ca53d

Layout View





▶ 14UTH Thesis31/3/2024

14

### Applications: Circuit Floorplan

The diagram illustrates a circuit floorplan with several key components labeled: Memory Controller at the top; four Core blocks in the middle; a Queue block between the second and third cores; Shared L3 Cache at the bottom; Misc IO on the left; and PCIe on the right. A navigation bar at the bottom shows a play button, the number 15, 'UTH Thesis', and '31/3/2024'.

15

### Applications: Floorplan

The diagram shows a floorplan consisting of a grid of Memory Local Cache (MLC) blocks. Each block is labeled with a unique ID, such as MLC202, MLC201, MLC200, MLC199, MLC198, MLC197, MLC196, MLC195, MLC194, MLC193, MLC192, MLC191, MLC190, MLC189, MLC188, MLC187, MLC186, MLC185, MLC184, MLC183, MLC182, MLC181, MLC180, MLC179, MLC178, MLC177, MLC176, MLC175, MLC174, MLC173, MLC172, MLC171, MLC170, MLC169, MLC168, MLC167, MLC166, MLC165, MLC164, MLC163, MLC162, MLC161, MLC160, MLC159, MLC158, MLC157, MLC156, MLC155, MLC154, MLC153, MLC152, MLC151, MLC150, MLC149, MLC148, MLC147, MLC146, MLC145, MLC144, MLC143, MLC142, MLC141, MLC140, MLC139, MLC138, MLC137, MLC136, MLC135, MLC134, MLC133, MLC132, MLC131, MLC130, MLC129, MLC128, MLC127, MLC126, MLC125, MLC124, MLC123, MLC122, MLC121, MLC120, MLC119, MLC118, MLC117, MLC116, MLC115, MLC114, MLC113, MLC112, MLC111, MLC110, MLC109, MLC108, MLC107, MLC106, MLC105, MLC104, MLC103, MLC102, MLC101, MLC100, MLC99, MLC98, MLC97, MLC96, MLC95, MLC94, MLC93, MLC92, MLC91, MLC90, MLC89, MLC88, MLC87, MLC86, MLC85, MLC84, MLC83, MLC82, MLC81, MLC80, MLC79, MLC78, MLC77, MLC76, MLC75, MLC74, MLC73, MLC72, MLC71, MLC70, MLC69, MLC68, MLC67, MLC66, MLC65, MLC64, MLC63, MLC62, MLC61, MLC60, MLC59, MLC58, MLC57, MLC56, MLC55, MLC54, MLC53, MLC52, MLC51, MLC50, MLC49, MLC48, MLC47, MLC46, MLC45, MLC44, MLC43, MLC42, MLC41, MLC40, MLC39, MLC38, MLC37, MLC36, MLC35, MLC34, MLC33, MLC32, MLC31, MLC30, MLC29, MLC28, MLC27, MLC26, MLC25, MLC24, MLC23, MLC22, MLC21, MLC20, MLC19, MLC18, MLC17, MLC16, MLC15, MLC14, MLC13, MLC12, MLC11, MLC10, MLC9, MLC8, MLC7, MLC6, MLC5, MLC4, MLC3, MLC2, MLC1, MLC0. A navigation bar at the bottom shows a play button, the number 16, 'UTH Thesis', and '31/3/2024'.

16

### Applications: 3D design

**MEMS** **Memory**

Micro-bump  
TSV  
metal layers

Wirebond Flip Chip Stacked Die Package-on-Package 3D-IC

1990 1995 2000 2005 2010

17 UTH Thesis 31/3/2024

17

### Applications: 3D design

**MEMS** **Memory**

Micro-bump  
TSV  
Rerun  
metal layers

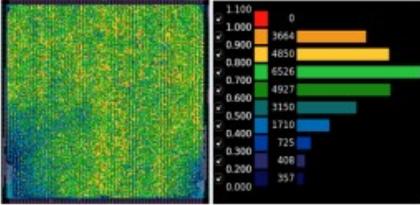
**Monolithic Device** **First 3D FPGA: Virtex-7 2000T** **First Heterogeneous 3D FPGA: Virtex-7 H580T**

Based on Stacked Silicon Interconnect Based on Stacked Silicon Interconnect

18 UTH Thesis 31/3/2024

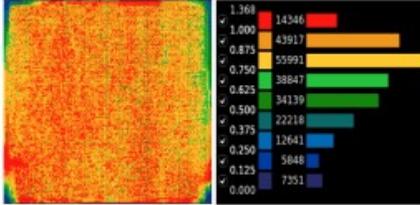
18

## Applications: 3D design (2)



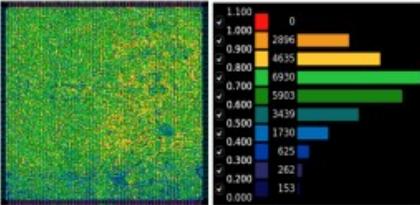
(a) Cell Density map for 1<sup>st</sup> tier of FFT 3D Design

1.100	0
1.000	3664
0.900	4830
0.800	6526
0.700	4927
0.600	3150
0.500	1710
0.400	725
0.300	408
0.200	357
0.000	



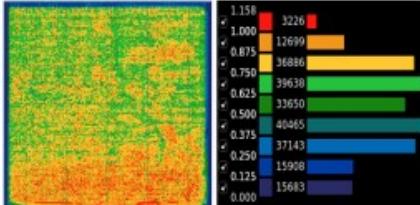
(b) Wire Utilization map for 1<sup>st</sup> tier of FFT 3D Design

1.368	14346
1.000	43917
0.875	55993
0.750	38847
0.625	34139
0.500	22218
0.375	12641
0.250	5848
0.125	7351
0.000	



(c) Cell Density map for 2<sup>nd</sup> tier of FFT 3D Design

1.100	0
1.000	2896
0.900	4635
0.800	6930
0.700	5903
0.600	3439
0.500	1730
0.400	625
0.300	262
0.200	153
0.000	



(d) Wire Utilization map for 2<sup>nd</sup> tier of FFT 3D Design

1.150	3226
1.000	12699
0.875	36886
0.750	39630
0.625	33650
0.500	40465
0.375	37143
0.250	15908
0.125	15683
0.000	

▶ 19
UTH Thesis
31/3/2024

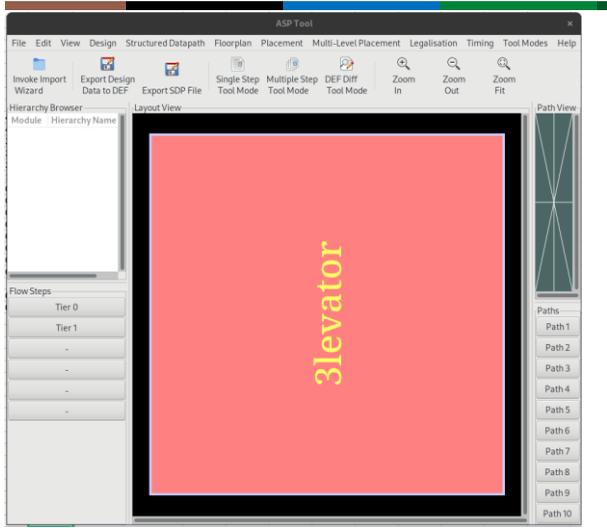
19



2023 INTERNATIONAL  
CONFERENCE ON  
COMPUTER-AIDED  
DESIGN  
42nd Edition

# ICCAD 2023





The screenshot shows the ASP Tool interface with a central 3D model of an elevator. The word "3levator" is written vertically on the model. The interface includes a menu bar (File, Edit, View, Design, etc.), a Hierarchy Browser on the left, a Flow Steps panel with "Tier 0" and "Tier 1" buttons, and a Path View panel on the right listing "Path 1" through "Path 10".

▶ 20
UTH Thesis
31/3/2024

20

IEEE/ACM ICCAD 2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 42nd Edition

IC/CAD ontest

### ICCAD 2023

21 UTH Thesis 31/3/2024

21

IEEE/ACM ICCAD 2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 42nd Edition

IC/CAD ontest

### ICCAD 2023

1st floor 2nd floor

22 UTH Thesis 31/3/2024

22

## Thesis Outline

1. Definitions
2. Applications
3. Existing tools
4. Our contribution
5. Evaluation
6. Conclusions
7. Future work



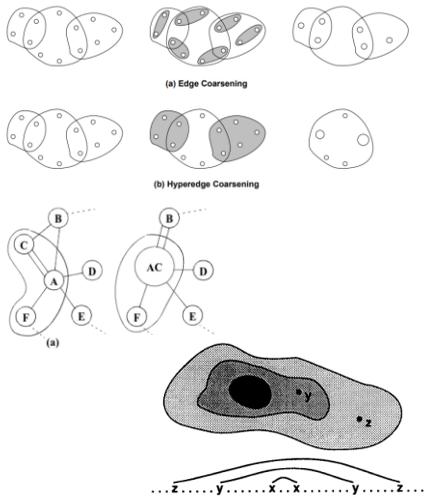
The visual elements for the thesis outline include: a word cloud with 'DEFINITION' in large red letters; a person at a computer with various icons; a megaphone with a laptop and various icons; a person at a computer with various icons; a bar chart with 'RESULTS' on the y-axis; a chalkboard with 'CONCLUSION' written on it; a puzzle with one blue piece missing; and an orange arrow pointing right with 'FUTURE' written inside it, with a person walking towards it.

▶ 23 UTH Thesis 31/3/2024

23

## Existing Clustering Algorithms: Novel

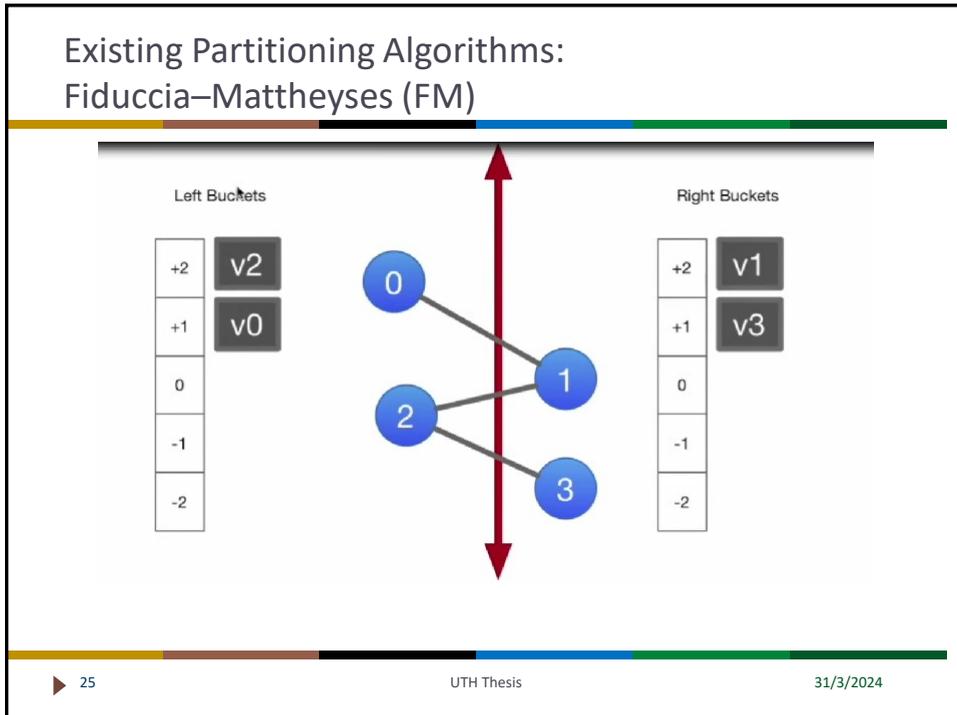
- ▶ **Edge Coarsening**
  - ▶ Find the maximal set of vertices pairs that belong in many hyperedges
- ▶ **Hyperedge Coarsening**
  - ▶ Find the maximal independent set of hyperedges, and group their nodes
- ▶ **First Choice**
  - ▶ Groups together vertices, such that each vertex in the group is highly connected with at least one other vertex in the same group
- ▶ **Best Choice**
  - ▶ Identifies the globally best pair of objects to cluster by managing a priority-queue with the clustering score as a key.
- ▶ **Random Walk**
  - ▶ Through random walks detects cycles/reconvergent points in the circuit creating natural clusters



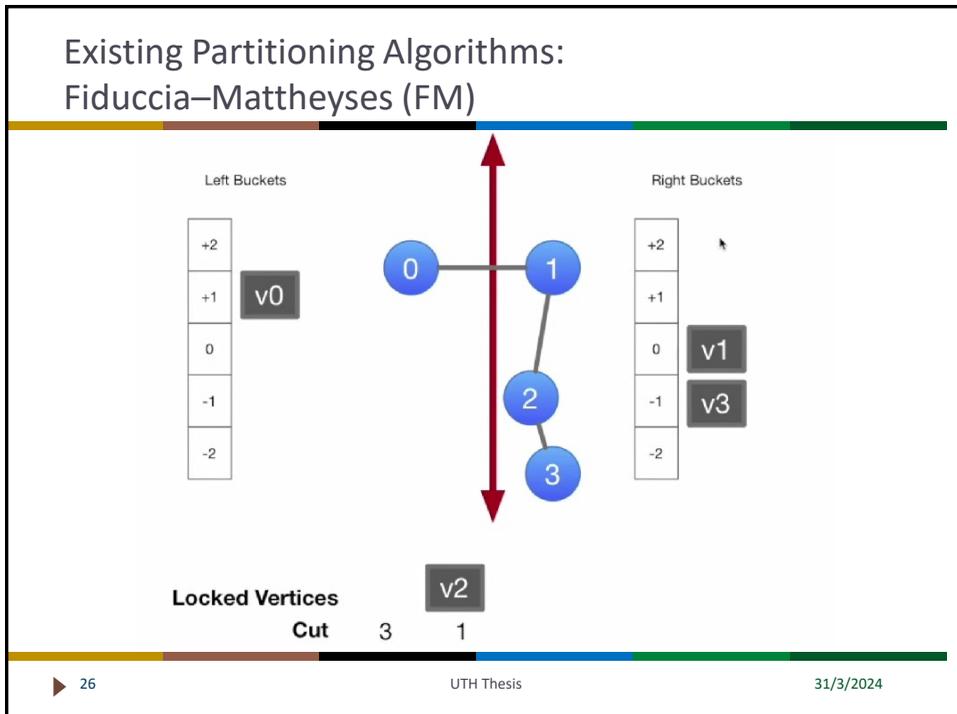
The diagrams illustrate the algorithms: (a) Edge Coarsening shows two overlapping sets of nodes being merged; (b) Hyperedge Coarsening shows a hyperedge being split into two; (a) shows a graph with nodes A-F and edges AC, AD, AE, AF, BC, CD, DE, EF; (b) shows a graph with nodes A-F and edges AC, AD, AE, AF, BC, CD, DE, EF, with a shaded region around nodes A and C; and a diagram showing a path z...y...x...y...z with a shaded region around nodes y and z.

▶ 24 UTH Thesis 31/3/2024

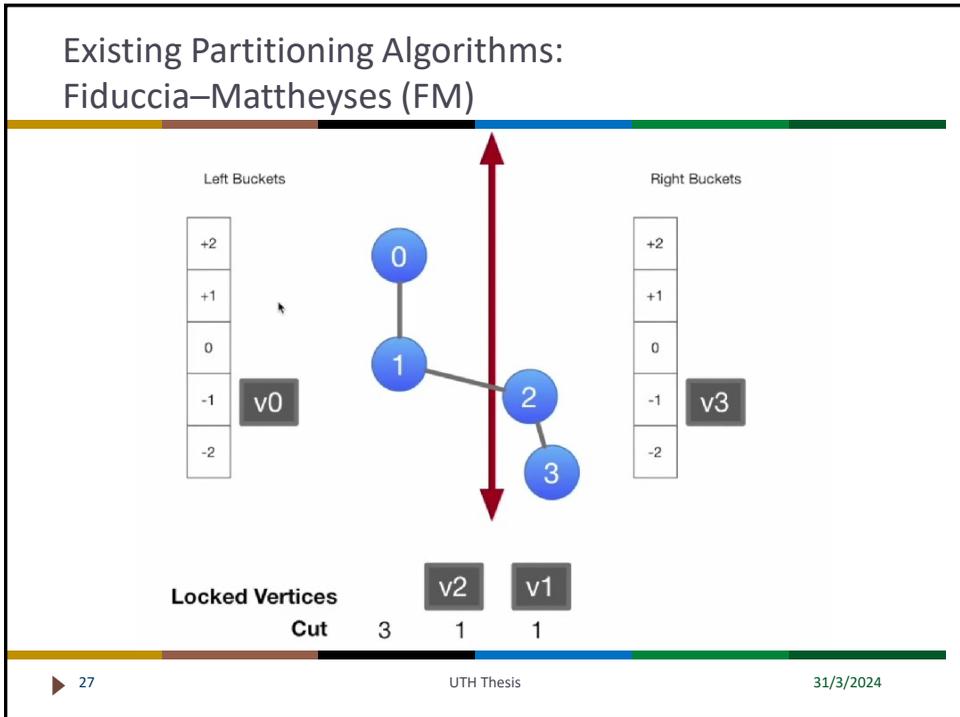
24



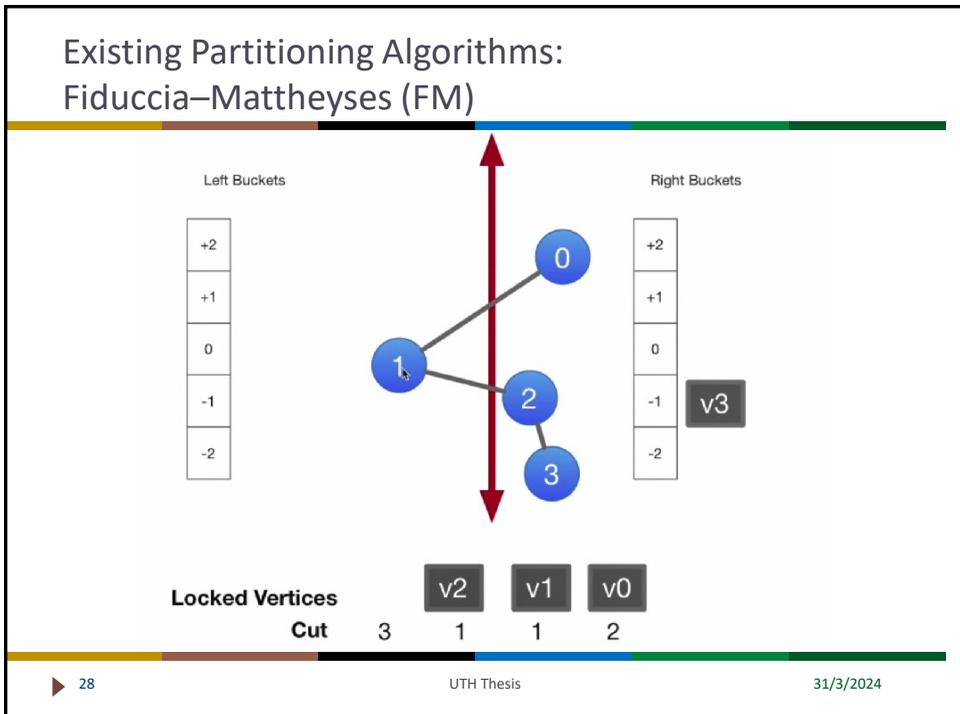
25



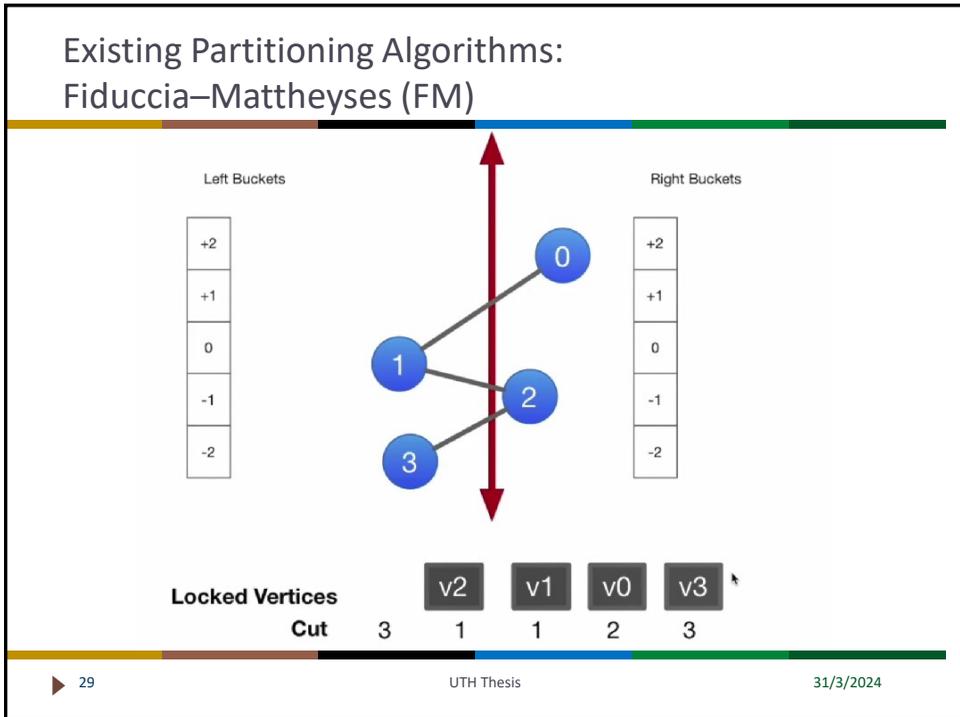
26



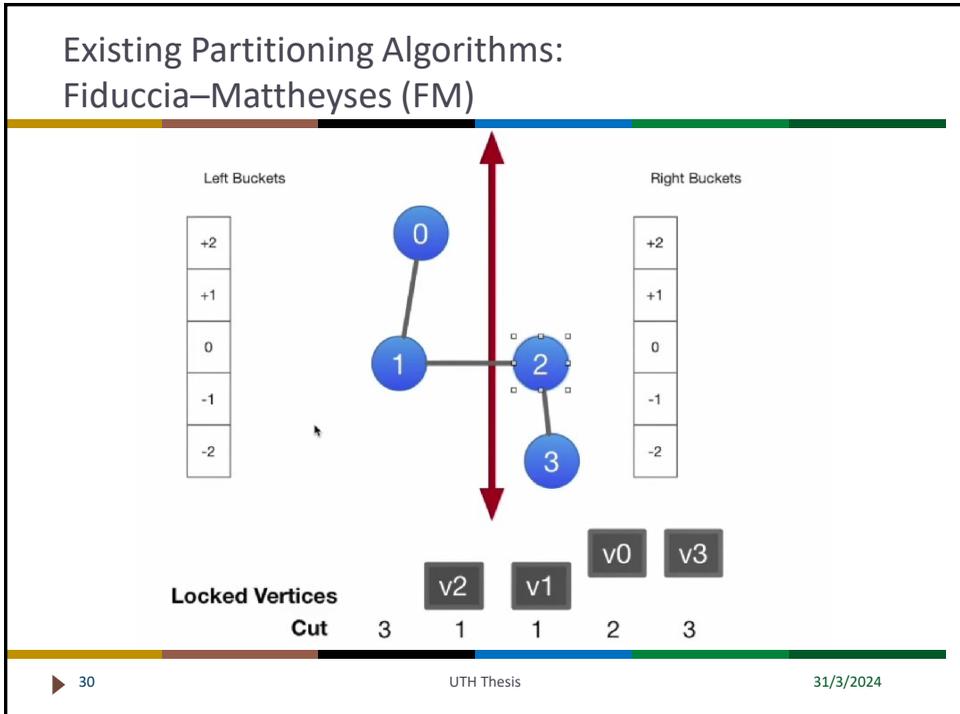
27



28



29



30

## Existing Partitioning Algorithms: Kernighan–Lin (KL)

**Cut cost: 9**  
**Not fixed:**  
1,2,3,4,5,6,7,8

$D_1 = 1$     $D_5 = 1$   
 $D_2 = 1$     $D_6 = 2$   
 $D_3 = 2$     $D_7 = 1$   
 $D_4 = 1$     $D_8 = 1$

$g_1 = 2+1-0-3$   
for Swap (3,5)

**Cut cost: 6**  
**Not fixed:**  
1,2,4,6,7,8

$D_1 = -1$     $D_6 = 2$   
 $D_2 = -1$     $D_7 = -1$   
 $D_4 = 3$     $D_8 = -1$

$g_2 = 3+2-0-5$   
for Swap (4,6)

**Cut cost: 1**  
**Not fixed:**  
1,2,7,8

$D_1 = -3$     $D_7 = -3$   
 $D_2 = -3$     $D_8 = -3$

$g_3 = -3-3-0-6$   
for Swap (1,7)

**Cut cost: 7**  
**Not fixed:**  
2,8

$D_2 = -1$   
 $D_8 = -1$

$g_4 = -1-1-0-2$   
for Swap (2,8)

**Cut cost: 9**  
**Not fixed:**  
-

▶ 31
UTH Thesis
31/3/2024

31

## Existing Partitioning Tools:

- ▶ **MLPart**
- ▶ **hMetis**
- ▶ **PaToH**
- ▶ **KaHyPar**
- ▶ **SpecPart**
- ▶ **GAP**

*k*-way Hypergraph Partitioning via *n*-Level Recursive Bisection

Sebastian Schlag\*   Vitali Heuer\*   Tobias Heuer\*   Henning Meyerhenke\*  
Peter Sanders\*   Christian Schulte\*

**Multilevel Circuit Partitioning<sup>1</sup>**

Charles J. Alpert<sup>†</sup>, Jen-Hsun Huang<sup>‡</sup> and Andrew B. Kahng<sup>‡</sup>

UCLA Computer Science Department, Los Angeles, CA 90095-1596  
<sup>†</sup> IBM Austin Research Laboratory, Austin, TX 78758  
<sup>‡</sup> Synopsys, Inc., Mountain View, CA 94043  
\* Cadence Design Systems, Inc., San Jose, CA 95134

**Multilevel Hypergraph Partitioning: Application in VLSI Domain**

George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar  
(karypis, rajat, kumar, shekhar)@cs.umn.edu  
University of Minnesota, Computer Science Department, Minneapolis, MN 55455

**Advanced Flow-Based Multilevel Hypergraph Partitioning**

Lars Gottschleben  
Karlsruhe Institute of Technology, Germany  
lars.gottschleben@kit.edu

Michael Humann  
Karlsruhe Institute of Technology, Germany  
michael.humann@kit.edu

Sebastian Schlag  
Karlsruhe Institute of Technology, Germany  
schlag@kit.edu

Dorothea Wagner  
Karlsruhe Institute of Technology, Germany  
dorothea.wagner@kit.edu

**Asynchronous n-Level Hypergraph Partitioning**

Master Thesis of  
Moritz Laupichler

**SpecPart: A Supervised Spectral Framework for Hypergraph Partitioning Solution Improvement**

Imanol Bustany   Andrew B. Kahng   Ioannis Kourtis  
Advanced Micro Devices   University of California San Diego   New Jersey Institute of Technology  
San Jose, CA, USA   La Jolla, CA, USA   Newark, NJ, USA  
imanol.bustany@gmail.com   abk@ucsd.edu   ikourtis@njit.edu

Rodhiastita Pramank   Zhang Wang  
Louis State University   University of California  
Ames, IA, USA   La Jolla, CA, US  
rodhi9@louis.edu   zhw002@ucsd.edu

**Deterministic Parallel Hypergraph Partitioning**

Lars Gottschleben   Michael Humann  
lars.gottschleben@kit.edu   michael.humann@kit.edu  
Karlsruhe Institute of Technology   Karlsruhe Institute of Technology  
Karlsruhe, Germany

**Parallel Flow-Based Hypergraph Partitioning**

Lars Gottschleben  
Karlsruhe Institute of Technology, Karlsruhe, Germany

Tobias Heuer  
Karlsruhe Institute of Technology, Karlsruhe, Germany

Peter Sanders  
Karlsruhe Institute of Technology, Karlsruhe, Germany

**High-Quality Hypergraph Partitioning**

SEBASTIAN SCHLAG, TOBIAS HEUER, LARS GOTTESBÜREN, and YAROSLAV AKHREMITSEV, Karlsruhe Institute of Technology  
CHRISTIAN SCHULZ, Heidelberg University  
PETER SANDERS, Karlsruhe Institute of Technology

**Parallel Flow-Based Hypergraph Partitioning**

Lars Gottschleben  
Karlsruhe Institute of Technology, Karlsruhe, Germany

Tobias Heuer  
Karlsruhe Institute of Technology, Karlsruhe, Germany

Peter Sanders  
Karlsruhe Institute of Technology, Karlsruhe, Germany

**Deterministic Parallel Hypergraph Partitioning**

Lars Gottschleben   Michael Humann  
lars.gottschleben@kit.edu   michael.humann@kit.edu  
Karlsruhe Institute of Technology   Karlsruhe Institute of Technology  
Karlsruhe, Germany

**GAP: Generalizable Approximate Graph Partitioning Framework**

Azade Nazi<sup>†</sup>, Will Hang<sup>†</sup>, Anna Goldie<sup>†</sup>, Sujith Ravi<sup>†</sup>, Azalia Mirhoseini<sup>†</sup>

<sup>†</sup>Google Research; <sup>‡</sup>Stanford University  
<sup>†</sup>{nazide, azade, will, anna}@google.com, <sup>‡</sup>{ellhang}@stanford.edu

▶ 32
UTH Thesis
31/3/2024

32

## Thesis Outline

1. Definitions 
2. Applications 
3. Existing tools 
4. **Our contribution** 
5. Results 
6. Conclusions 
7. Future work 

▶ 41
UTH Thesis
31/3/2024

41

## Test Cases

	Design	Components	Nets	Macros	Pins	Library		Design	Components	Nets	Macros	Pins	Library		
ISPD06	adaptec1	5.72E+05	5.83E+05	542	1	ASAP7	DAC12	superblue11	9.26E+05	9.59E+05	1458	6872	ASAP7		
	adaptec2	4.57E+05	4.69E+05	543	1			superblue12	1.27E+06	1.29E+06	89	1580			
	adaptec3	9.69E+05	9.93E+05	723	1			superblue14	6.05E+05	6.30E+05	340	5473			
	adaptec4	1.09E+06	1.13E+06	1329	1			superblue16	6.71E+05	6.98E+05	419	4448			
	adaptec5	2.15E+06	2.18E+06	646	1			superblue19	4.95E+05	5.12E+05	286	3735			
	bigblue1	5.98E+05	6.06E+05	559	1			superblue2	9.51E+05	9.91E+05	654	8047			
	bigblue2	8.30E+05	8.83E+05	3313	1			superblue3	9.08E+05	9.33E+05	575	6482			
	bigblue3	1.65E+06	1.69E+06	675	1			superblue5	7.09E+05	7.87E+05	784	4082			
	bigblue4	5.14E+06	5.22E+06	667	1			superblue6	9.52E+05	1.01E+06	565	5380			
	newblue1	4.73E+05	4.86E+05	390	1			superblue7	1.32E+06	1.34E+06	419	6499			
	newblue2	6.61E+05	7.11E+05	1171	1			superblue9	8.11E+05	8.34E+05	272	4014			
	newblue3	8.32E+05	9.23E+05	690	1			Industrial 1	5.00E+04	6.09E+04	0	2176		IND	
	newblue4	1.47E+06	1.51E+06	569	1			Industrial 2	1.40E+05	1.48E+05	0	1159			
	ISPD11	newblue5	1.84E+06	1.93E+06	1052			1	OpenSource	b19	2.20E+05	2.26E+05		0	47
newblue6		2.71E+06	2.77E+06	1376	1	jpeg	6.70E+05	6.74E+05		0	67				
newblue7		4.39E+06	4.62E+06	6151	1	leon3	6.50E+05	7.58E+05		0	333				
superblue1		7.98E+05	8.23E+05	432	6521	netcard	9.60E+05	1.06E+06		0	1846				
superblue10		1.05E+06	1.09E+06	1619	15141	ASAP7	ICCAD15	superblue1		1.21E+06	1.22E+06	3787	3767496	Contest	
superblue12		1.27E+06	1.29E+06	89	1580			superblue10		1.88E+06	1.90E+06	1696	5560508		
superblue15		1.07E+06	1.08E+06	153	10556			superblue16		9.82E+05	1.00E+06	101	3013270		
superblue18	4.59E+05	4.69E+05	207	3978	superblue18			7.68E+05	7.71E+05	653	2559145				
superblue2	9.51E+05	9.91E+05	654	8047	superblue3			1.21E+06	1.22E+06	2074	3905323				
superblue4	5.59E+05	5.81E+05	306	6623	superblue4			7.96E+05	8.02E+05	3471	2497942				
superblue5	7.09E+05	7.87E+05	784	4082	superblue5			1.09E+06	1.10E+06	1872	3246880				
						superblue7	1.93E+06	1.93E+06	4910	6372096					

▶ 42
UTH Thesis
31/3/2024

42

## ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example:**

- Specify the number of the clusters

cl1    cl2    cl3    cl4

4 clusters must created/grown in this level

▶ 43
UTH Thesis
31/3/2024

43

## ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example: Round 1**

- Add the 1<sup>st</sup> pair of objects of highest fanout net (or affinity)
- Lock used objects

cl1    cl2    cl3    cl4

▶ 44
UTH Thesis
31/3/2024

44

## ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example: Round 1**

- Add the 1<sup>st</sup> two objects of the 2<sup>nd</sup> highest fanout net
- Lock them

cl1    cl2    cl3    cl4

▶ 45
UTH Thesis
31/3/2024

45

## ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example: Round 1**

- Add the 1<sup>st</sup> two objects of the 3<sup>rd</sup> highest fanout net
- Lock them

cl1    cl2    cl3    cl4

▶ 46
UTH Thesis
31/3/2024

46

## ML Cluster Growth

? Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

? except for the first move

**Example: Round 1**

- Add the 1<sup>st</sup> two objects of the 4<sup>th</sup> highest fanout net
- Lock these objects

cl1



cl2

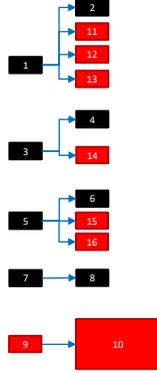


cl3



cl4





▶ 47
UTH Thesis
31/3/2024

47

## ML Cluster Growth

? Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

? except for the first move

**Example: Round 2**

- Add the next object of the 1<sup>st</sup> highest fanout net
- Lock it

cl1



cl2

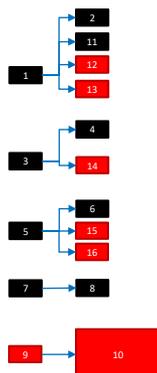


cl3



cl4





▶ 48
UTH Thesis
31/3/2024

48

### ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example: Round 2**

- Add the next object of the 2<sup>nd</sup> highest fanout net
- Lock it

cl1    cl2    cl3    cl4

▶ 49
UTH Thesis
31/3/2024

49

### ML Cluster Growth

---

**?** Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

**?** except for the first move

**Example: Round 2**

- Add the next object of the 3<sup>rd</sup> highest fanout net
- Lock it

cl1    cl2    cl3    cl4

▶ 50
UTH Thesis
31/3/2024

50

## ML Cluster Growth

? Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

? except for the first move

**Example: Round 2**  
 - Add the next object of the 3<sup>rd</sup> highest fanout net  
 - Lock it

cl1

cl2

cl3

cl4

If we want clusters to be area balanced, growth must stop at round 2; otherwise, e.g. if area balance is 2, we continue until area balance constraint is satisfied.

▶ 51
UTH Thesis
31/3/2024

51

## ML Cluster Growth

? Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time

? except for the first move

**Example: Round 3**  
 - Add the next object of the 1<sup>st</sup> highest fanout net  
 - Lock it

cl1

cl2

cl3

cl4

Area balance is 2

▶ 52
UTH Thesis
31/3/2024

52

### ML Cluster Growth

? Idea of Multiple Cluster Growing it to grow all current level clusters by adding one object at the time  
? except for the first move

**Example: Round 3**  
 - Add the next object of the 2<sup>nd</sup> highest fanout net  
 - Lock it

cl1    cl2    cl3    cl4

Area balance is 2

▶ 53
UTH Thesis
31/3/2024

53

### ML Cluster Growth

? The idea of Parallel Clusters Growing it to grow all clusters of the current level, by adding one object at the time (except the first grow).

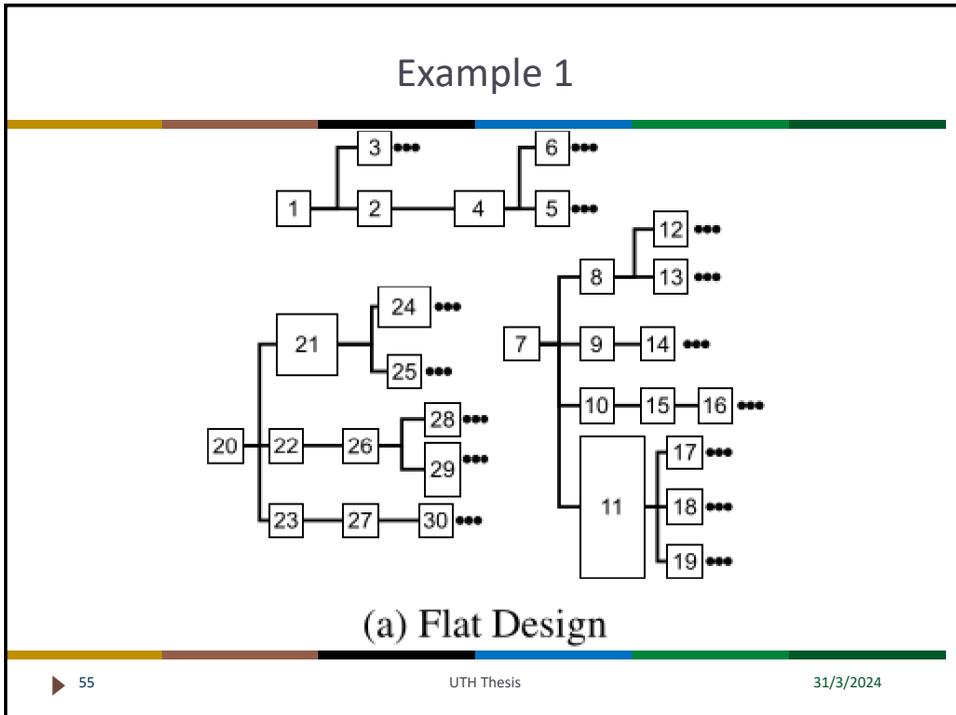
**Example: Round 3**  
 - Add the next object of the 2<sup>nd</sup> highest fanout net  
 - Lock this it

cl1    cl2    cl3    cl4

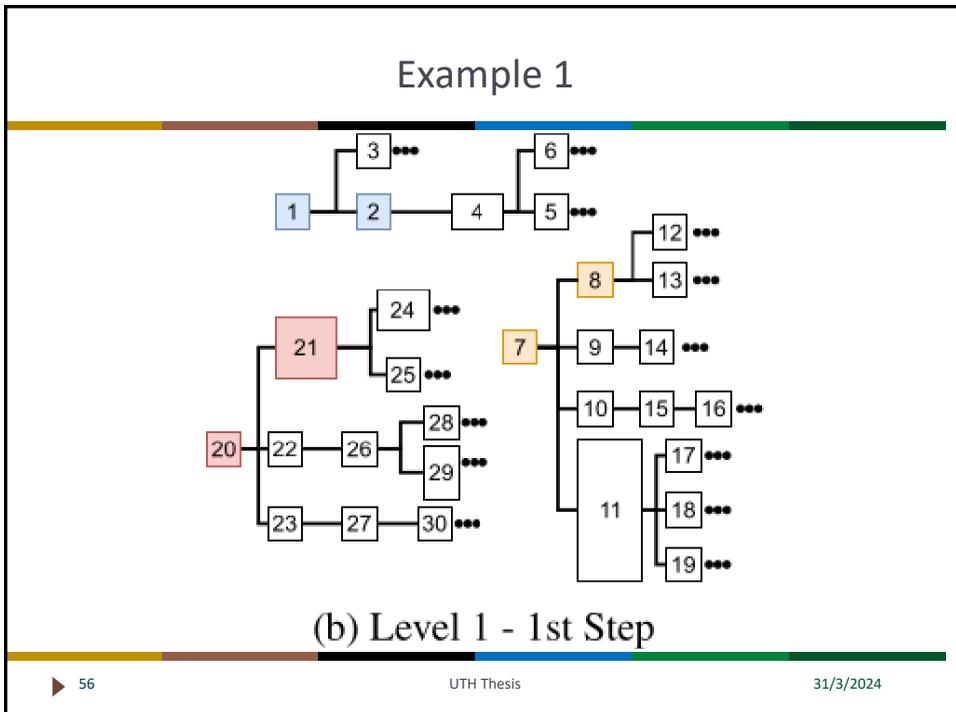
Object "13" will NOT added to ML Cluster "cl1" due to the area balance criterion, as the area of cl1 is twice that of smaller ML Cluster (cl4)

▶ 54
UTH Thesis
31/3/2024

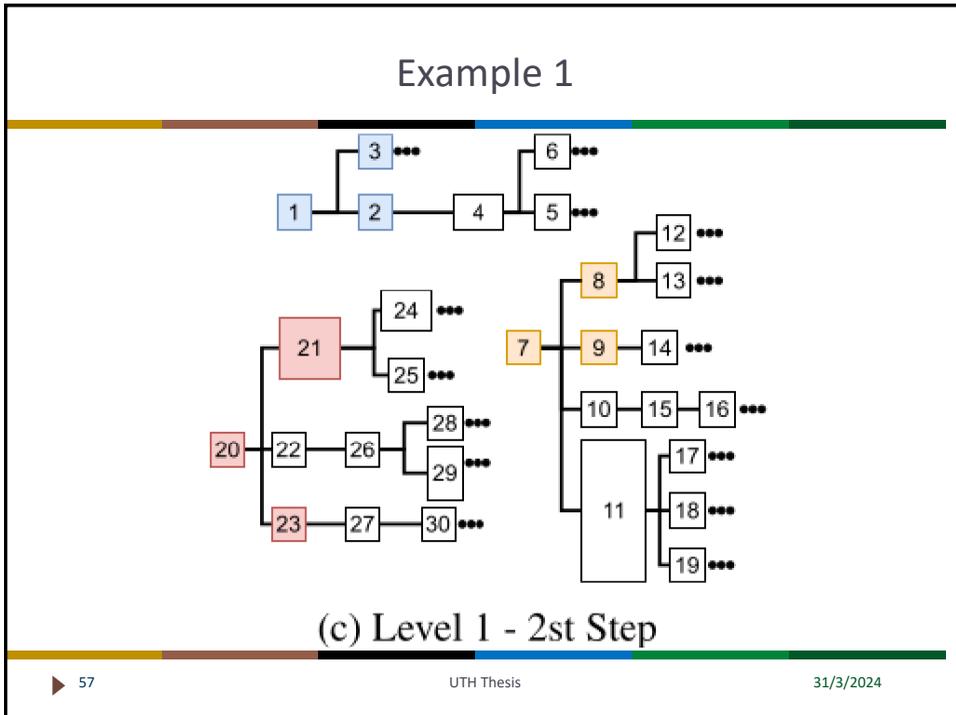
54



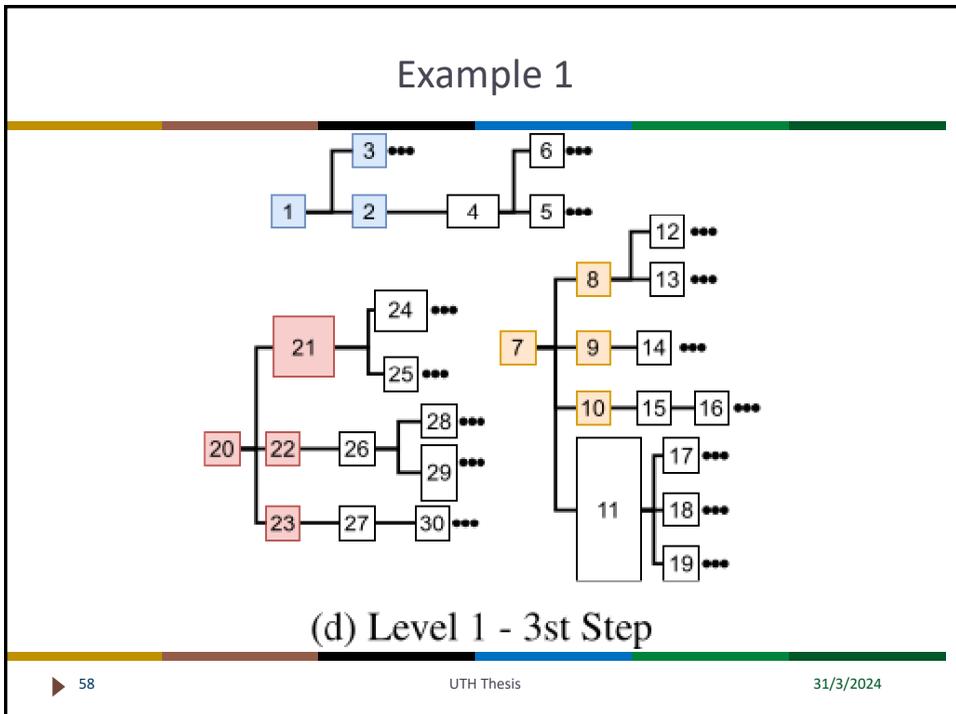
55



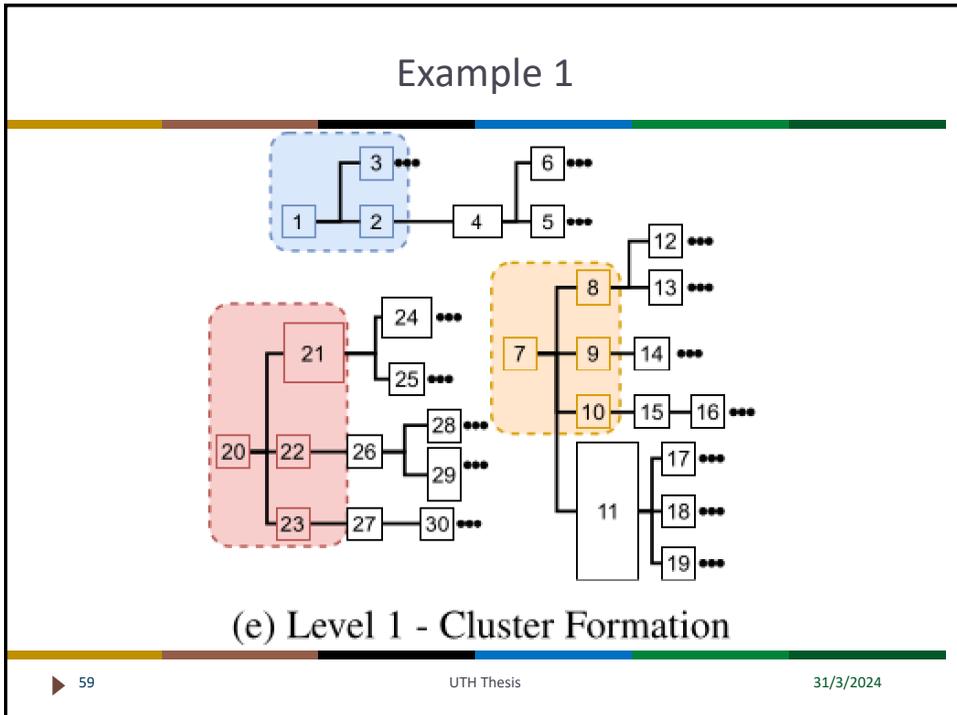
56



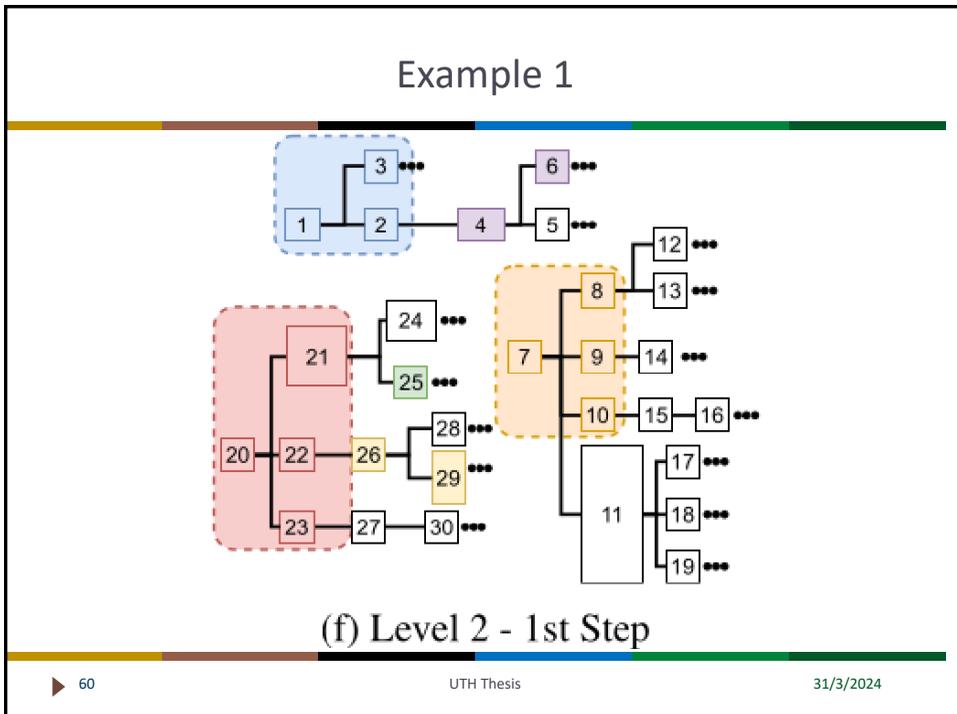
57



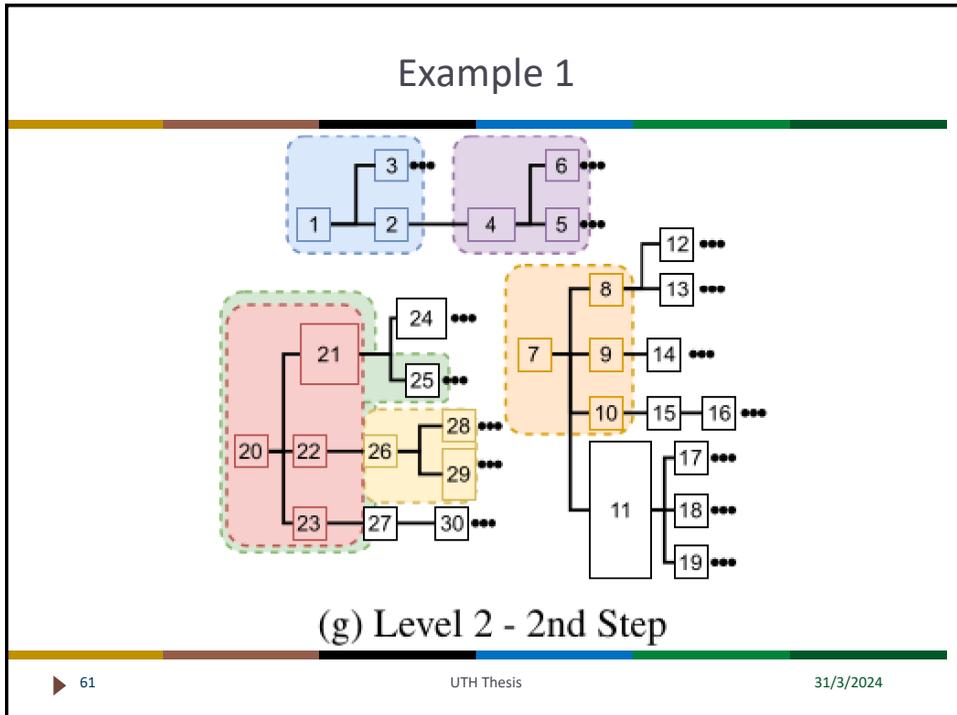
58



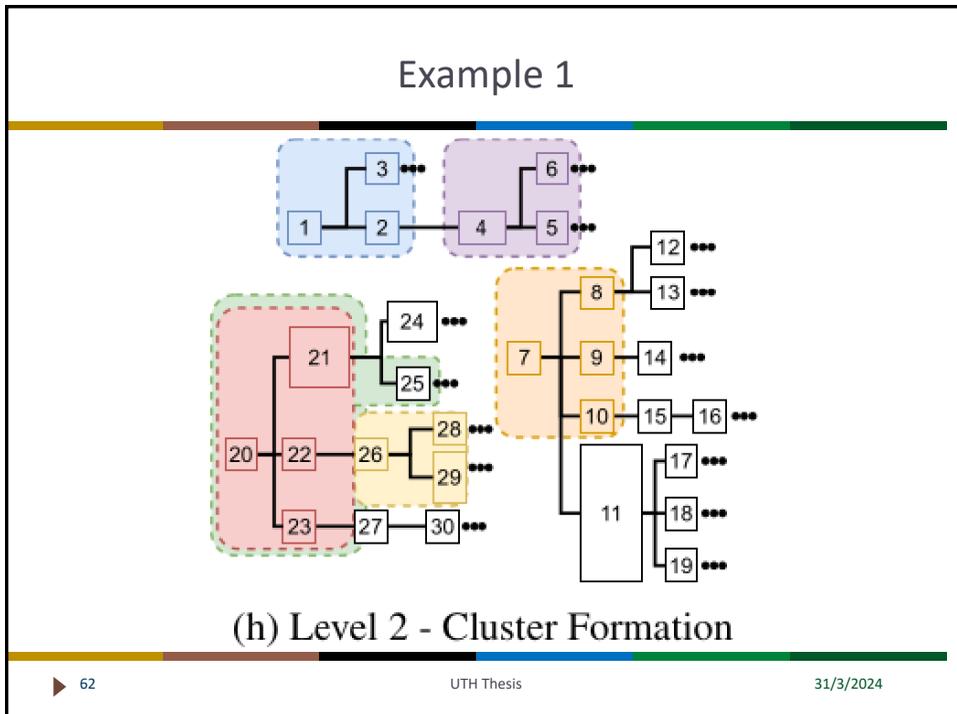
59



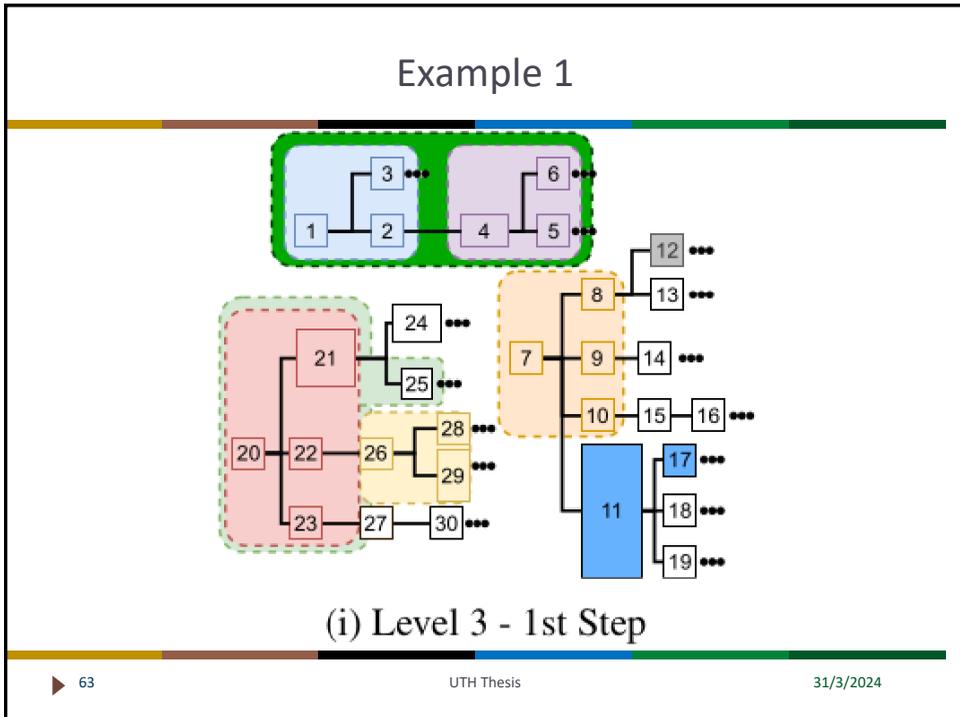
60



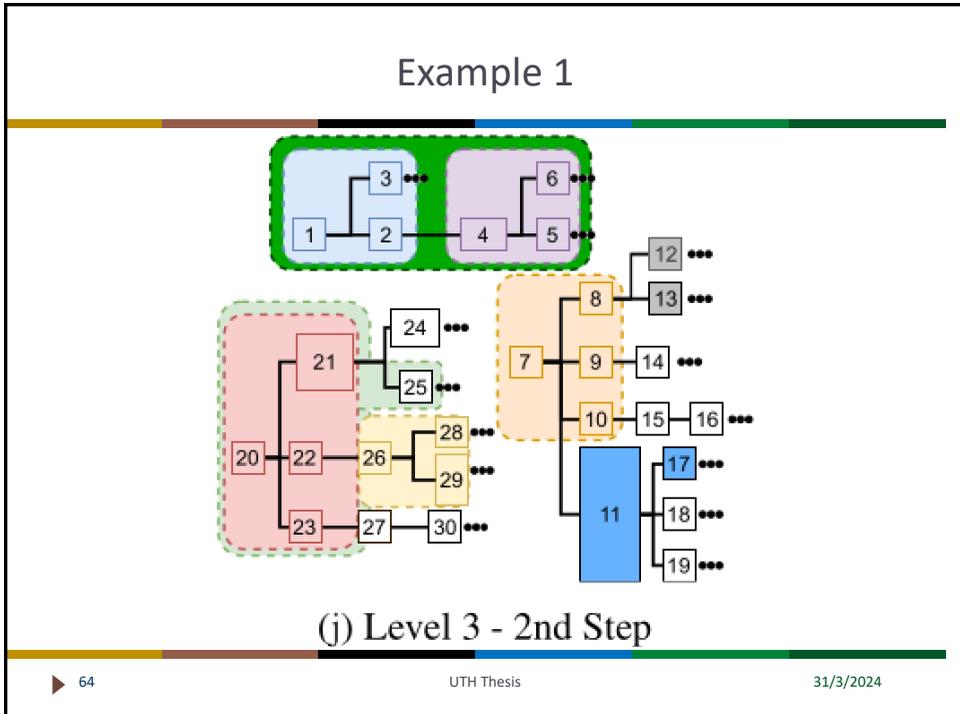
61



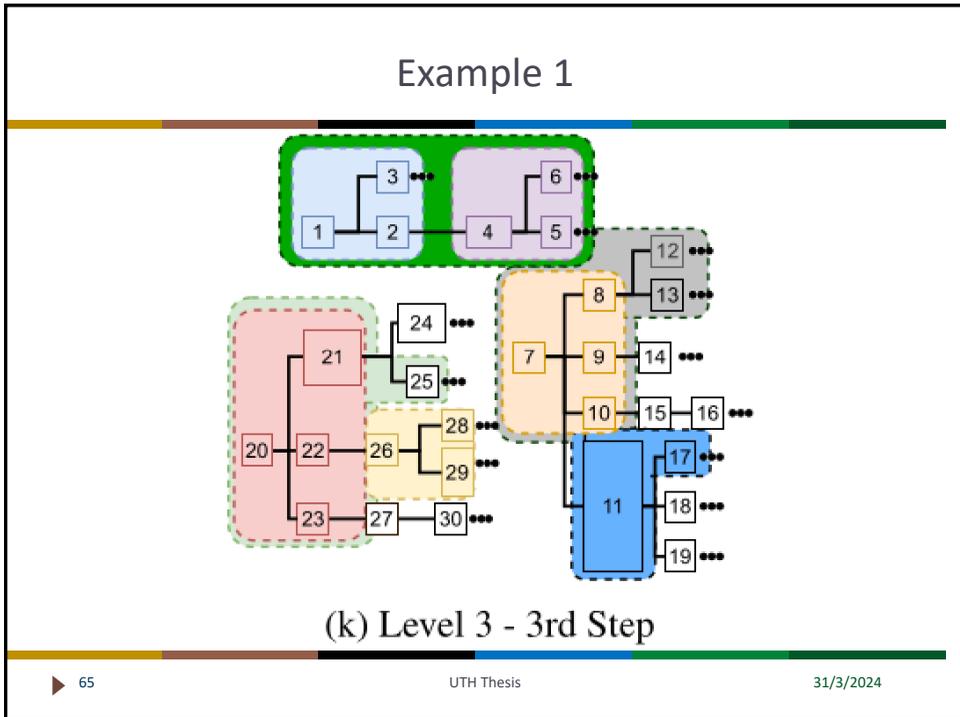
62



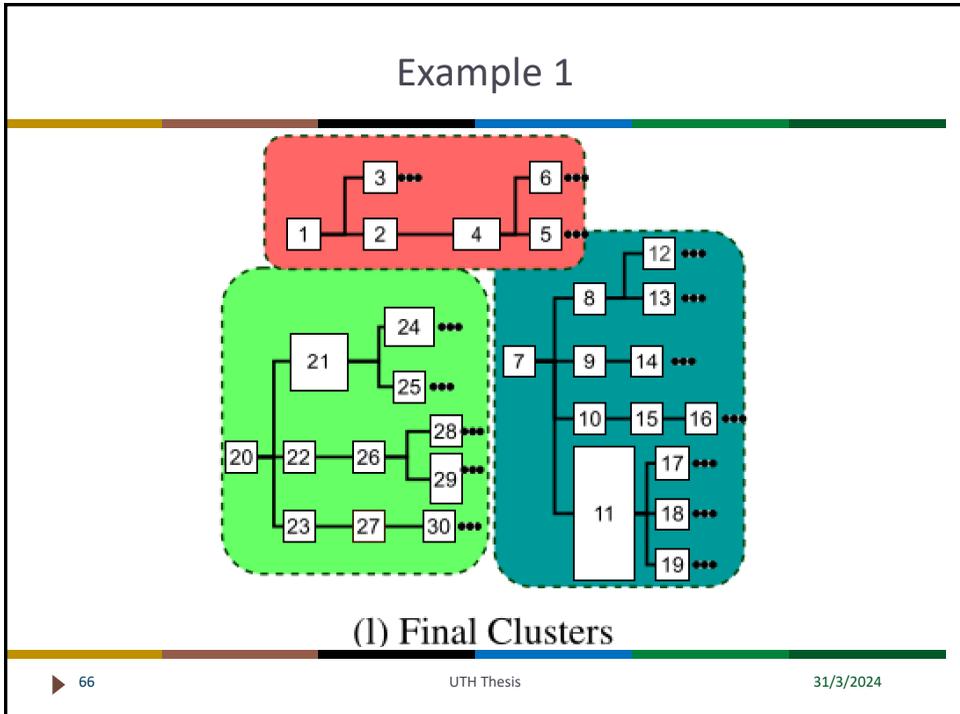
63



64



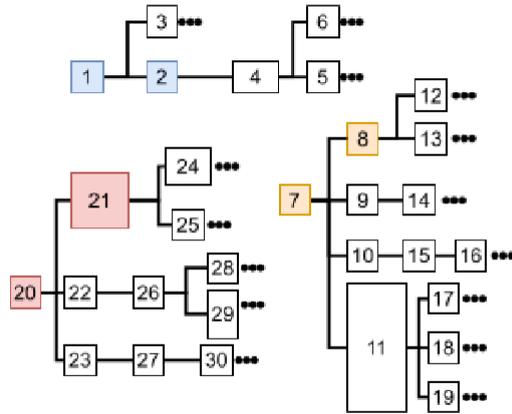
65



66



## Clustering Approach: MultiNet / Swarm



(b) Level 1 - 1st Step

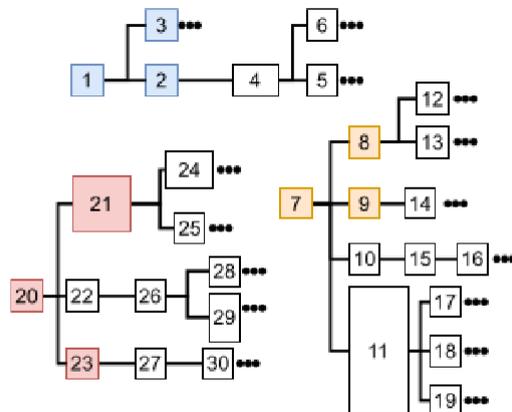
▶ 69

UTH Thesis

31/3/2024

69

## Clustering Approach: MultiNet / Swarm



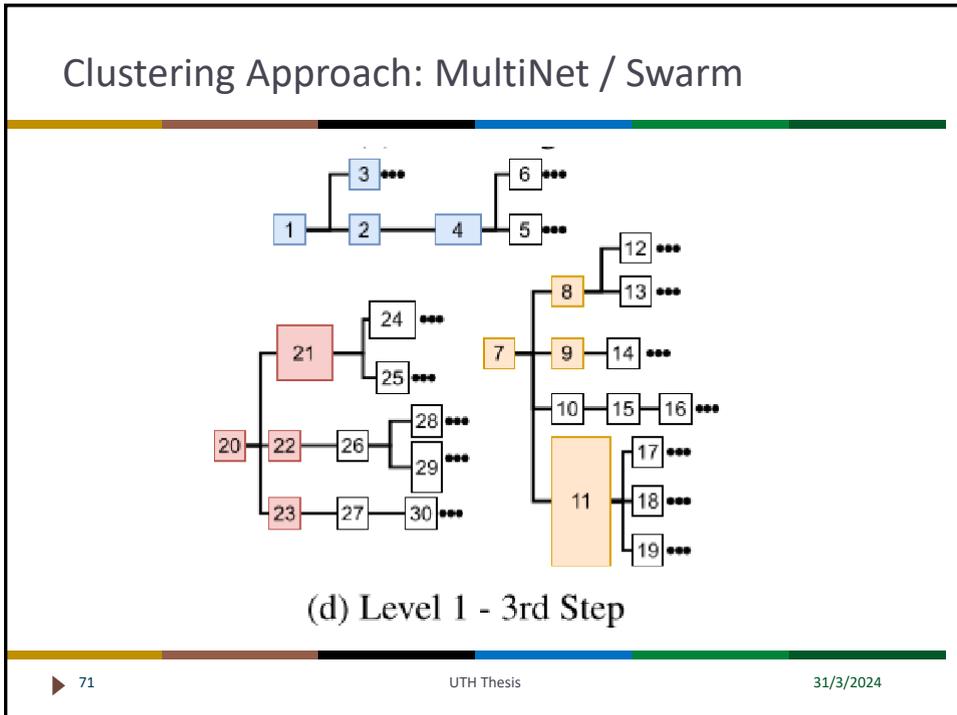
(c) Level 1 - 2nd Step

▶ 70

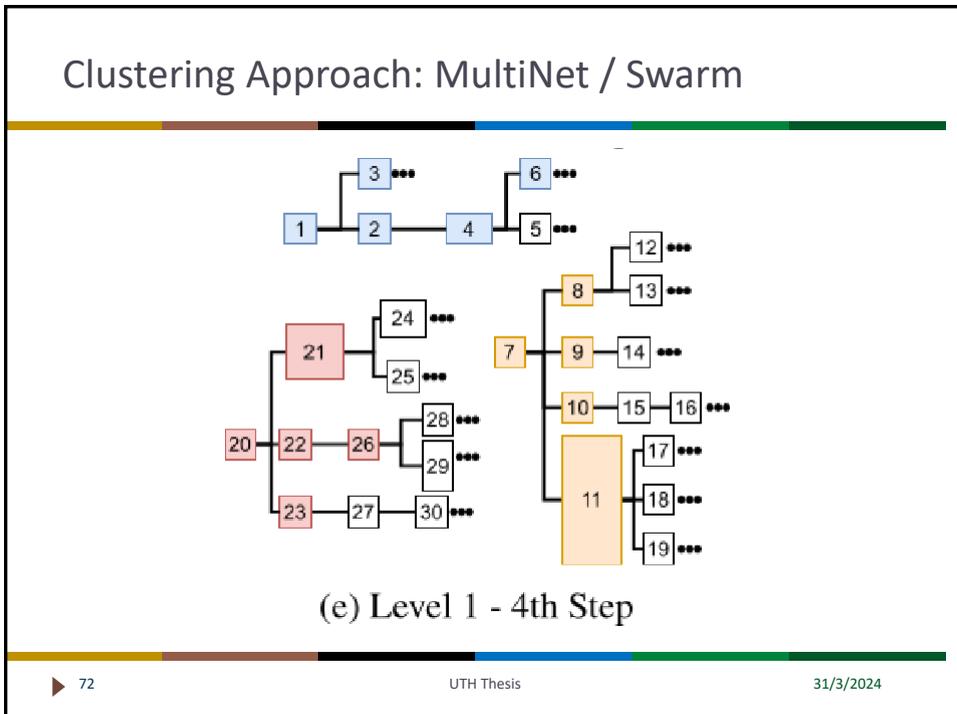
UTH Thesis

31/3/2024

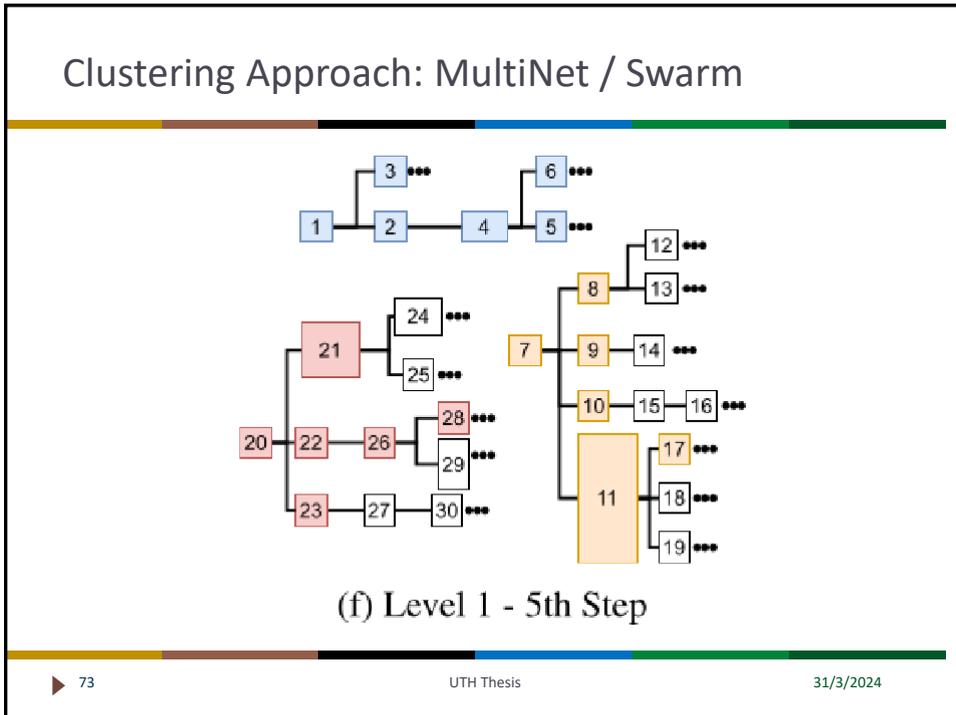
70



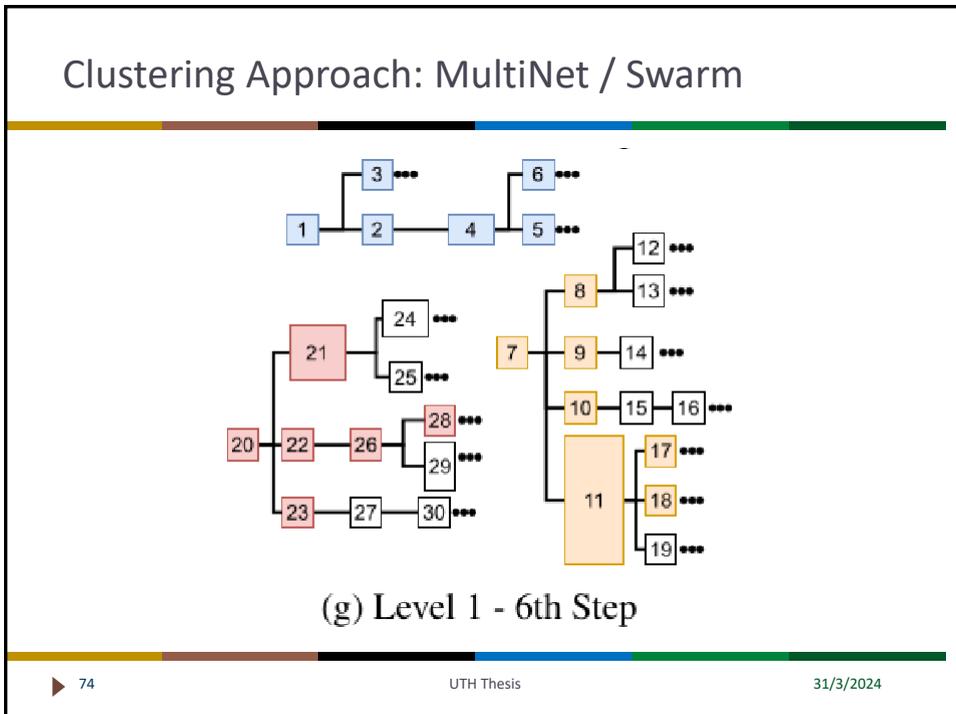
71



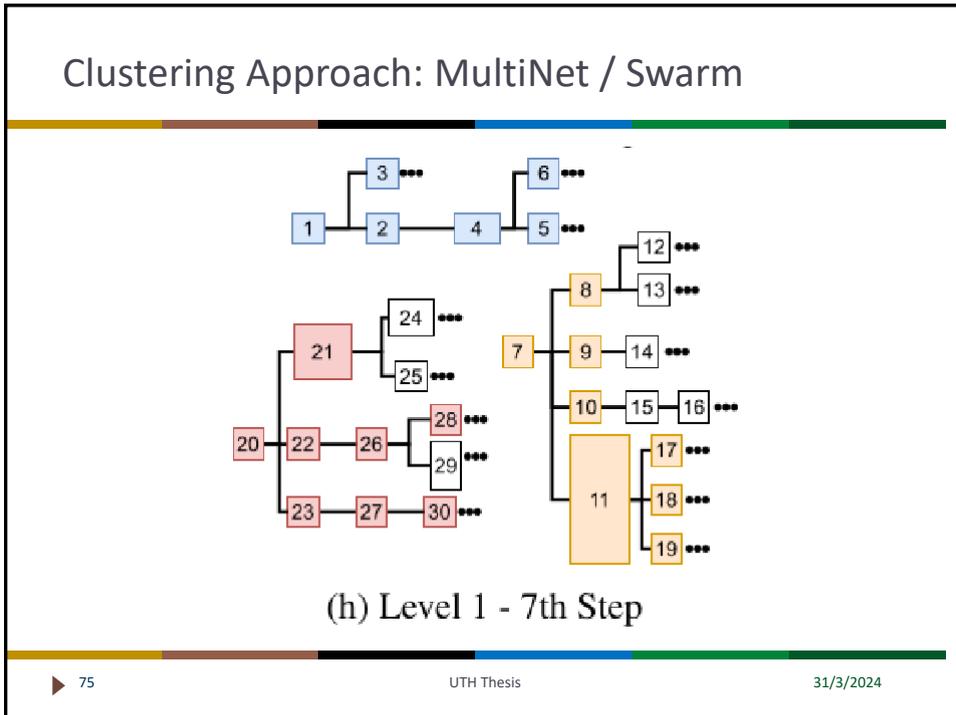
72



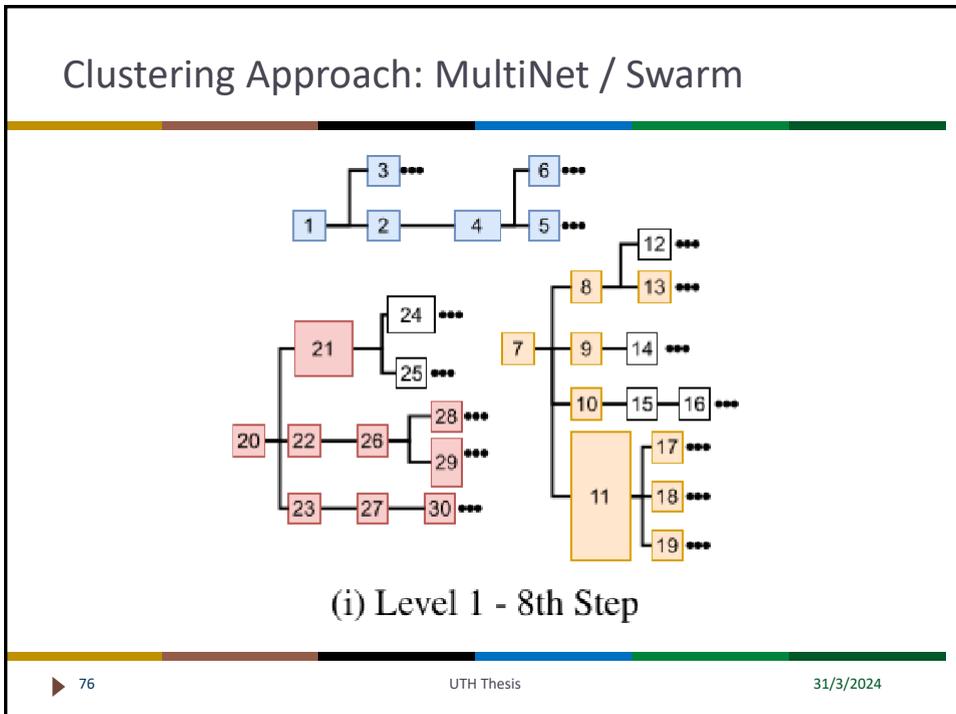
73



74



75



76

## Clustering Approach: MultiNet / Swarm

(j) Level 1 - Cluster Formation (Final Clusters)

▶ 77
UTH Thesis
31/3/2024

77

## Clustering Algorithm Comparison

Design	Levels	Level Area Ratio	Through Levels Area Ratio	Unclusterd Components
Industrial 1	14	104.49	1.59	2011
Industrial 2	18	802.89	1.56	246
b19	57	62.20	1.11	1
jpeg	26	1870.80	1.38	0
leon3	20	25.94	1.22	2
netcard	25	27.78	1.27	4250

Design	Levels	Level Area Ratio	Through Levels Area Ratio	Unclusterd Components
Industrial 1	9	10.43	1.01	1253
Industrial 2	8	10.13	1.18	231
b19	5	8.16	1.28	0
jpeg	5	10.72	1.27	0
leon3	5	15.09	1.33	0
netcard	25	6.67	0.97	1

▶ 78
UTH Thesis
31/3/2024

78

1<sup>st</sup> Paper

## Swarm - A VLSI Timing, Fanout-aware Clustering Algorithm

Christos P. Sotiriou , George Rafael Goudroumanis , Nikolaos Sketopoulos , and Christos Georgakidis   
 University of Thessaly, EECE Department, Volos, Greece  
 {chaotiriou, ggeorgios-r, sketopou, cgeorgakidis}@e-cc.uth.gr



▶ 79

UTH Thesis

31/3/2024

79

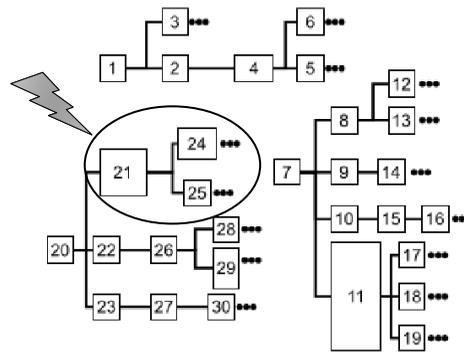
## Clustering Approach: Clusters Growth Techniques

▶ Group next candidate  
criticality notion

- ▶ Fanout
- ▶ Slack
- ▶ Delay

▶ Net next candidates  
ordering

- ▶ Increasing Order
- ▶ Decreasing Order



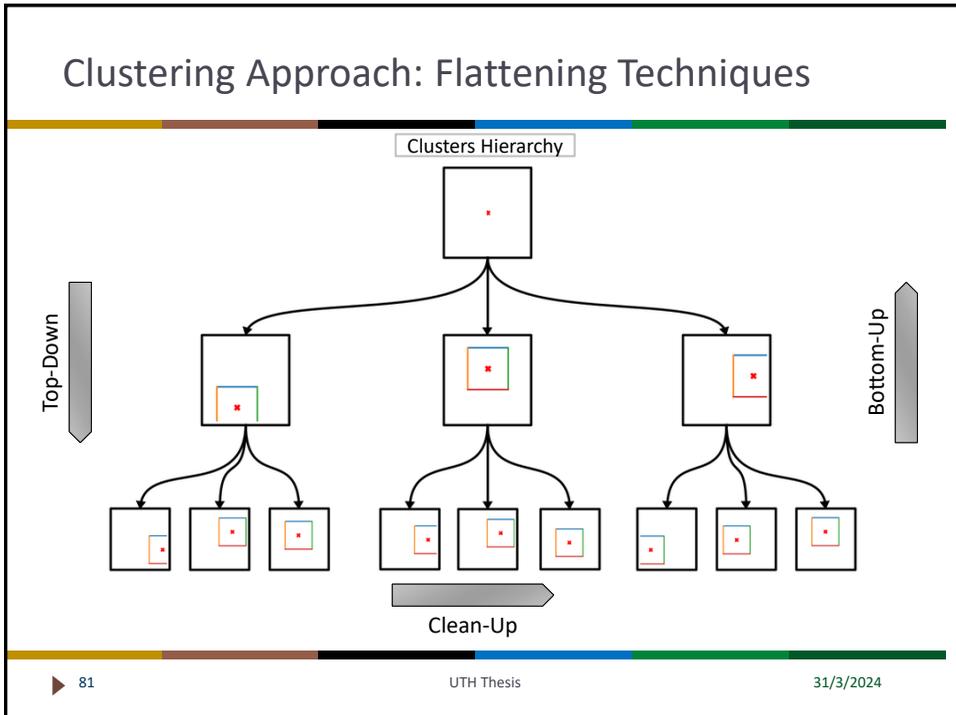
(a) Flat Design

▶ 80

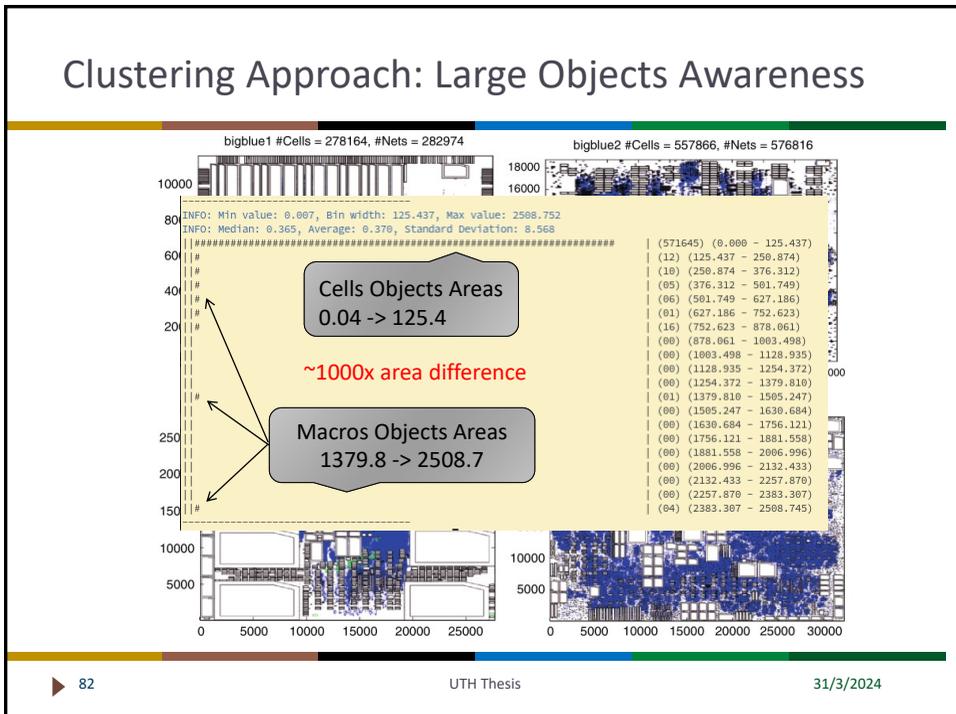
UTH Thesis

31/3/2024

80



81



82

## Results

Design	Large Objects Awareness				Novel Algorithm			
	Levels	Unclustered	AreaRatio	Exec Time	Levels	Unclustered	AreaRatio	Exec Time
adaptec1	9	863	94.28	44.916	9	664	93.44	35.695
adaptec3	14	3325	457.09	84.59	16	2012	403.30	74.35
adaptec4	15	4896	311.63	72.10	16	3228	458.44	105.663
adaptec5	17	5751	1351.18	180.74	18	3975	1643.95	271.471
bigblue1	10	1426	157.02	34.99	10	1060	216.59	45.695
newblue1	8	2260	145.14	30.383	8	2201	146.79	40.95
newblue3	14	10997	265.42	62.231	15	9837	306.20	65.92
newblue6	17	6953	601.53	330.33	18	4845	834.34	412.61

83 UTH Thesis 31/3/2024

83

## Our Contribution: Vcycle Flow

```

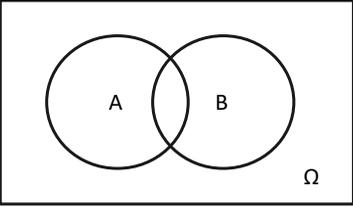
Input: Nodelist (Standard CGIC, Nets), Cluster Hierarchy, Partitions Number PN, Area Balance Factor ABF, Gain Type GT, Post Processing Optimization Phase PPOP, PM type FMT, Level Unfolding Strategy LUF.
Output: Set of Partitions, satisfying input parameters.
1: Hierarchy_Level = get_noclusters_maxlevel();
2: initial_phase = 1;
3: repeat
4:   if (initial_phase == 1) then
5:     objects = get_level_objects(Hierarchy_Level);
6:     initialise_recursive_binpartitioning_binary_tree(theNodelist, PN);
7:     status = recursive_binpartitioning_MF(objects, ABF, GT, FMT);
8:     if (status == -1) then
9:       /* error status, the algorithm must exit */
10:      break;
11:    else if (status == -2) then
12:      /* warning status, not enough objects, to fill in all partitions, in this level */
13:      Hierarchy_Level = Hierarchy_Level - 1;
14:      continue;
15:    else
16:      Hierarchy_Level = Hierarchy_Level - 1;
17:      initial_phase = 0
18:    end if
19:    optimise_area();
20:  end if
21:  unfold_partitions_level(Hierarchy_Level, Clusters Hierarchy, LUF);
22:  objects = get_level_objects(Hierarchy_Level);
23:  if (check_level_constraints()) == TRUE then
24:    status = heap_partitioning(objects, ABF, GT, FMT);
25:    if (status == -1) then
26:      /* error status, the algorithm must exit */
27:      break;
28:    else
29:      optimise_area();
30:      Hierarchy_Level = Hierarchy_Level - 1;
31:    end if
32:  end if
33: until (Hierarchy_Level == 0)
34: optimise_outsize(PPOP)
    
```

85 UTH Thesis 31/3/2024

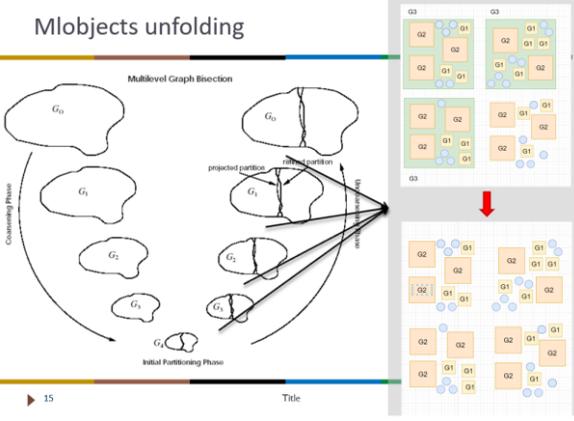
85



## Our Contribution: Unfolding



### Mobjects unfolding



1. All mobjects of the level ( $\Omega$ )
2. Mobjects on Cut (A)
3. Unfolded mobjects (B)
4. Unfolded mobjects **AND** Mobjects on Cut (A && B)
5. Unfolded mobjects **OR** Mobjects on Cut (A || B)

▶ 88
UTH Thesis
31/3/2024

88

## Our Contribution: kway partitioning

```

unfold_partitions_level(Hierarchy_Level, Clusters Hierarchy, LUF);
objects = get_level_objects(Hierarchy_Level);
if(check_level_constraints() == TRUE) then
    status = kway_partitioning(objects, ABF, GT, FMT)
    ~ if (status == -1) then
        /* error status, the algorithm must exit */
        break;
    else
        optimise_area();
        Hierarchy_Level = Hierarchy_Level - 1;
    ~ end if
end if

```



▶ 89
UTH Thesis
31/3/2024

89

## Results

Design	On Cut			On Cut and Unfolded			Unfolded on Cut			Unfolded		
	Exec Time	Cutsizes	Area Ratio	Exec Time	Cutsizes	Area Ratio	Exec Time	Cutsizes	Area Ratio	Exec Time	Cutsizes	Area Ratio
Industrial 1	20.765	20,010	3.498	22.857	18,703	3.500	18.694	22,392	3.499	18.694	21,651	3.498
Industrial 2	40.791	52,015	3.500	47.230	50,277	3.500	34.676	58,396	3.500	32.603	57,662	3.500
b19	42.080	29,794	3.500	63.471	28,228	3.500	58.187	38,188	3.500	47.243	35,222	3.500
jpeg	66.545	54,194	3.500	111.184	44,983	3.500	39.958	74,017	3.500	65.469	50,127	3.500
leon3	2272.985	111,116	3.500	2305.853	101,585	3.500	916.663	138,738	3.500	397.786	125,847	3.500
netcard	6376.440	249,056	3.500	5059.119	249,376	3.500	1808.874	288,981	3.500	255.838	279,523	3.500
adaptec1	-	-	-	969.061	40,116	3.500	286.331	65,678	3.500	308.040	42,902	3.500
adaptec2	-	-	-	971.795	59,365	13.335	681.017	83,585	12.475	319.581	70,001	12.564
adaptec3	-	-	-	2,346.700	74,249	10.668	6,483.530	122,083	10.653	2393.446	94,107	10.599
adaptec4	-	-	-	1,902.323	43,881	3.500	95.865	93,081	3.500	360.759	52,157	3.500
adaptec5	-	-	-	9,776.844	129,713	5.887	29,994.876	211,733	5.903	10,735.438	203,455	5.951

91 UTH Thesis 31/3/2024

91

## Our FM version

**Input:** Level Objects, Partitions Number  $PN$ .

**Output:** Objects list representing partitions.

```

1: initialise_heaps(PN);
2: extract_partition_cutsizes_and_gains_mobjects();
3: repeat
4:   sort_heaps();
5:   store_partitions_characteristics(&gain, &arearatio, &object);
6: repeat
7:   pop_larger_gain_heap_node();
8:   get_object_characteristics(&area, &destination_partition);
9:   if (check_object_movement_for_area_violations(arearatio) == TRUE) then
10:     store_violating_object(object);
11:     continue;
12:   else
13:     pop_all_other_identical_instances(object);
14:   end if
15:   reinsert_all_violating_objects_into_heaps();
16:   move_object(destination_partition);
17:   update_partitions_characteristics();
18:   store_movement_logistics();
19: until (FM_tentative_iterations_evaluation() == TRUE)
20: detect_minimum_cutsizes();
21: until (FM_iterations_evaluation() == TRUE)
22: return(create_objects_lists());

```

Time Complexity approaches  
 $O(((P-1)*N)^4)$

Space Complexity is  
 $O((P-1)N)$

92

UTH Thesis

31/3/2024

92

## Our Contribution: Gain strategies

### Nets Mode

**Gain = External - Internal**

- External: The object nets which all of their members are located at partition A apart from one object
- Internal: The mobject nets which all of their members are located at the same

### Flylines Mode

**Gain = External - Internal**

- External: The mobject connections towards the other partitions
- Internal: The mobject connections towards the same partition

▶ 93
UTH Thesis
31/3/2024

93

## Results

Design	per_flylines			per_nets		
	Exec Time	Cutsizes	Area Ratio	Exec Time	Cutsizes	Area Ratio
Industrial 1	17.405	22023	3.498	36.835	33570	3.500
Industrial 2	25.282	58243	3.500	475.696	65878	3.500
b19	31.015	38188	3.500	327.222	65027	3.500
jpeg	43.348	73463	3.500	500.872	95612	3.500
leon3	1050.561	138738	3.500	5023.114	203939	3.500
netcard	2168.491	288981	3.500	-	-	-
adaptec1	60.979	65678	3.500	-	-	-
adaptec2	835.725	83585	12.475	-	-	-

▶ 102
UTH Thesis
31/3/2024

102

## Partitioning Approach: FM algorithm efforts

FM heap creation type	Characteristics
Exhaustive Effort	<ul style="list-style-type: none"> <li>• Insert into heap every possible move of each mlobject</li> <li>• Perform FM algorithm using user defined exit conditions</li> </ul>
High Effort	<ul style="list-style-type: none"> <li>• Insert into heap as possible moves only the partitions in which a net is located</li> <li>• Perform FM algorithm using user defined exit conditions</li> </ul>
Normal Effort	<ul style="list-style-type: none"> <li>• Insert into heap as possible moves only 10 partitions with which a cell is most heavily connected</li> <li>• Perform FM algorithm using user defined exit conditions</li> </ul>
Low Effort	<ul style="list-style-type: none"> <li>• Insert into heap as possible moves only 1 partition with which a cell is most heavily connected</li> <li>• Perform only one FM iteration regardless of the user defined exit conditions</li> </ul>

▶ 103
UTH Thesis
31/3/2024

103

## Results

Design	Log Output						Ratio																					
Industrial	<pre> [0.34m]INFO: Optimised area ratio: 3.499968, Execution time 1.002299, Iterations: 1 [0.30m]&gt;[0.34m]INFO: Performing k-way partitioning in level 1 [0.30m]&gt;[0.34m]INFO: Starting K-way Cutszie optimisation... [0.30m]&gt;[0.34m]INFO: Considerable Mlobjects 297204, Movable mlobjects 81010 [0.30m]&gt;[0.34m]INFO: Considerable Mlobjects 297204, Movable mlobjects 81010           </pre>						99																					
Industrial: b19	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Iteration</th> <th>Cutszie</th> <th>Area Ratio</th> <th>Area Ratio Error</th> <th>Performed Swaps</th> <th>Tested Swaps</th> <th>Execution Time</th> </tr> </thead> <tbody> <tr> <td>Init</td> <td>223854</td> <td>3.500</td> <td>-----</td> <td>-----</td> <td>-----</td> <td>-----</td> </tr> <tr> <td>1</td> <td>223555</td> <td>3.500</td> <td>0.000</td> <td>8810</td> <td>276274741</td> <td>313.360</td> </tr> </tbody> </table>						Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time	Init	223854	3.500	-----	-----	-----	-----	1	223555	3.500	0.000	8810	276274741	313.360	00
Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time																						
Init	223854	3.500	-----	-----	-----	-----																						
1	223555	3.500	0.000	8810	276274741	313.360																						
jpeg	<pre> [0.34m]INFO: Nets Based Multi-Level Partitions Cutszie: Per_CCs 202509, Per_Flyline 223460 [0.30m]&gt;[0.34m]INFO: Multi-Level Partitions Cutszie: Per_CCs 171161, Per_Flyline 223555 [0.30m]&gt;[0.90m]REPORT: K-way Partitioning CPU Time = 315.552 sec [0.30m]DEBUG: Unfolded mlobjects size: 512448           </pre>						00																					
leon3	<pre> [0.34m]INFO: Optimizing mlpartitions area ratio [0.30m]DEBUG: Exit condition: Area ratio achieved           </pre>						00																					
netcard	<pre> [0.34m]INFO: Optimised area ratio: 3.499936, Execution time 1.534325, Iterations: 1 [0.30m]&gt;[0.34m]INFO: Performing k-way partitioning in level 0 [0.30m]&gt;[0.34m]INFO: Starting K-way Cutszie optimisation... [0.30m]&gt;[0.34m]INFO: Considerable Mlobjects 674721, Movable mlobjects 162626 [0.30m]&gt;[0.34m]INFO: Considerable Mlobjects 674721, Movable mlobjects 162626           </pre>						00																					
adaptecl	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Iteration</th> <th>Cutszie</th> <th>Area Ratio</th> <th>Area Ratio Error</th> <th>Performed Swaps</th> <th>Tested Swaps</th> <th>Execution Time</th> </tr> </thead> <tbody> <tr> <td>Init</td> <td>223555</td> <td>3.500</td> <td>-----</td> <td>-----</td> <td>-----</td> <td>-----</td> </tr> <tr> <td>1</td> <td>223065</td> <td>3.500</td> <td>0.000</td> <td>21366</td> <td>551782358</td> <td>920.703</td> </tr> </tbody> </table>						Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time	Init	223555	3.500	-----	-----	-----	-----	1	223065	3.500	0.000	21366	551782358	920.703	00
Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time																						
Init	223555	3.500	-----	-----	-----	-----																						
1	223065	3.500	0.000	21366	551782358	920.703																						
adaptecl2	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Iteration</th> <th>Cutszie</th> <th>Area Ratio</th> <th>Area Ratio Error</th> <th>Performed Swaps</th> <th>Tested Swaps</th> <th>Execution Time</th> </tr> </thead> <tbody> <tr> <td>Init</td> <td>223555</td> <td>3.500</td> <td>-----</td> <td>-----</td> <td>-----</td> <td>-----</td> </tr> <tr> <td>1</td> <td>223065</td> <td>3.500</td> <td>0.000</td> <td>21366</td> <td>551782358</td> <td>920.703</td> </tr> </tbody> </table>						Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time	Init	223555	3.500	-----	-----	-----	-----	1	223065	3.500	0.000	21366	551782358	920.703	75
Iteration	Cutszie	Area Ratio	Area Ratio Error	Performed Swaps	Tested Swaps	Execution Time																						
Init	223555	3.500	-----	-----	-----	-----																						
1	223065	3.500	0.000	21366	551782358	920.703																						

▶ 104
UTH Thesis
31/3/2024

104

## Results

Design	low effort	normaleffort	high effort						
	<pre> @x1b [0:30m@x1b [0:34mINFO: Performing k-way partitioning on level 1 @x1b [0:30m@x1b [0:34mINFO: Starting k-way Cutszie optimisation ... @x1b [0:30m@x1b [0:34mINFO: Considerable Mobjects 297204, Movable mobjects 35403 @x1b [0:30m Iteration   Cutszie   Area Ratio   Area Ratio Error   Performed Swaps   Tested Swaps   Execution Time Init        88527     3.500       -----           -----           -----           ----- 1          82733     3.500       0.000             16260             220636            0.775 2          81744     3.500       0.000             14689             388893            0.785 3          81519     3.500       0.000             14393             484330            0.889 4          81448     3.500       0.000             14375             686472            0.996 @x1b [0:34mINFO: Nets Based Multi-Level Partitions Cutszie: Per_CCs 75700, Per_Flyline 81411 @x1b [0:30m@x1b [0:34mINFO: Multi-Level Partitions Cutszie: Per_CCs 65474, Per_Flyline 81448 @x1b [0:30m@x1b [0:90mREPORT: K-way Partitioning CPU Time = 5.275 sec @x1b [0:30mDEBUG: Unfolded mobjects size 512448 @x1b [0:34mINFO: Skip level 0 because the movable objects are too few (65902 / 674721) @x1b [0:30m@x1b [0:34mINFO: Nets Based Multi-Level Partitions Cutszie: Per_CCs 75700, Per_Flyline 81411 </pre>								
adaptec2	123.594	150,659	14.840	698.191	83,585	938.914	12.573	83,199	12.475

▶ 105

UTH Thesis

31/3/2024

105

## Partitioning Approach: FM algorithm versions

### Classic

- ▣ Until cutszie increment is detected perform FM Iterations
  - ▣ Create heap containing possible moves sorted by decreasing gain value
  - ▣ Until heap is empty perform FM tentative moves based on its sorting

### Greedy

- ▣ Until cutszie increment is detected perform FM Iterations
  - ▣ Create heap containing possible moves sorted by decreasing gain value
  - ▣ Until a negative gain is detected perform FM tentative moves based on heap sorting

### Early Exit

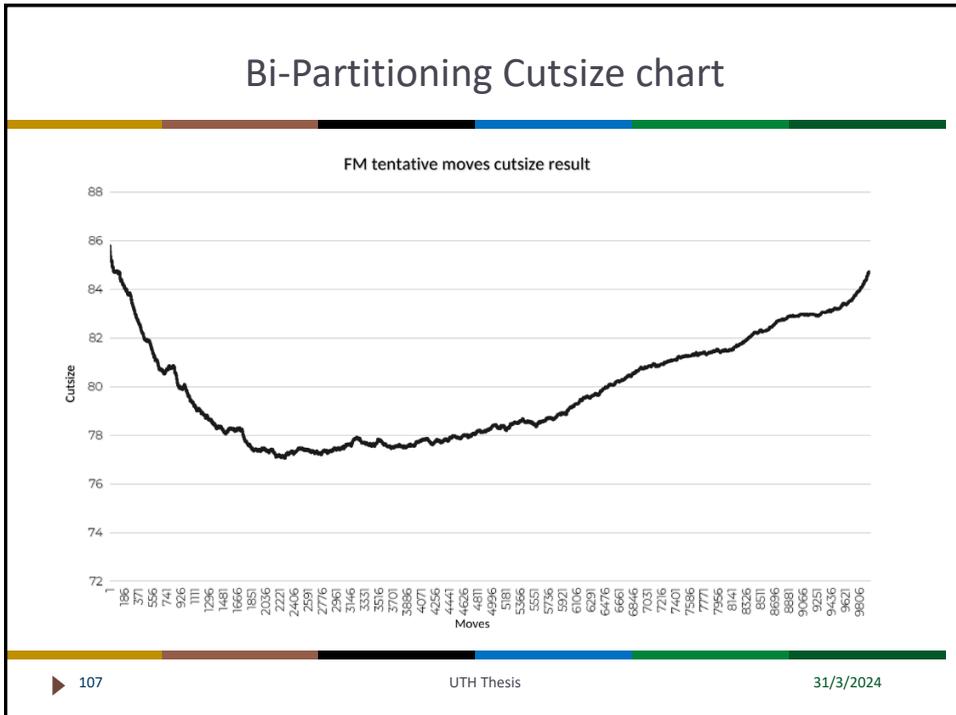
- ▣ Until user specified cutszie reduction percentage is achieved (cutszie reduction less than 1%)
  - ▣ Create heap containing possible moves sorted by decreasing gain value
  - ▣ Until user defined cutszie increment percentage is reached (cutszie after a tentative move > 20% of minimum) perform FM tentative moves based on heap sorting

▶ 106

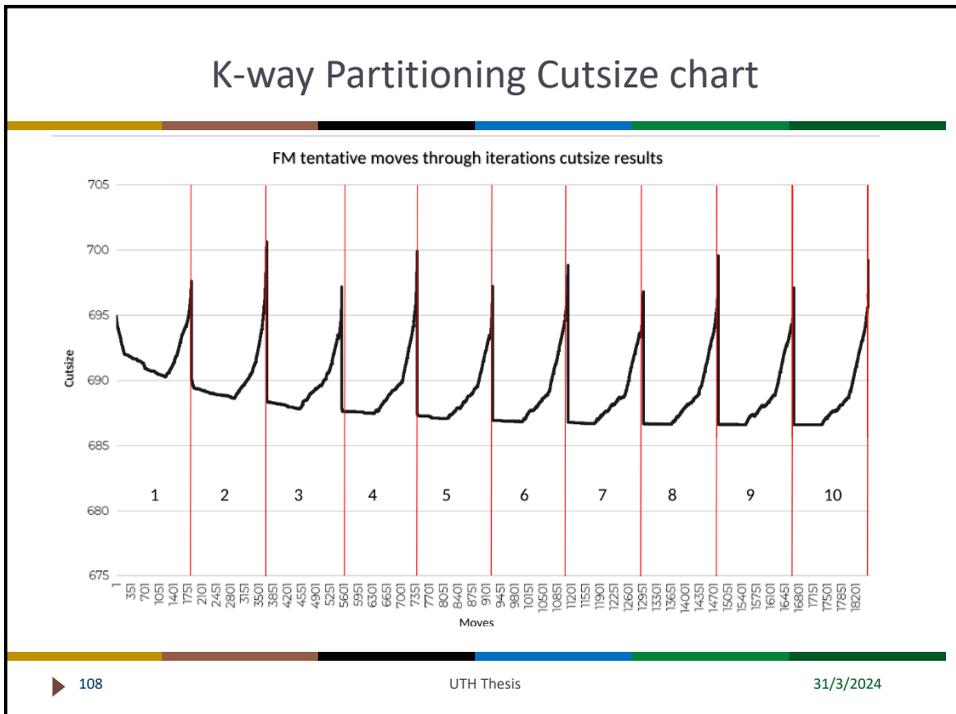
UTH Thesis

31/3/2024

106



107



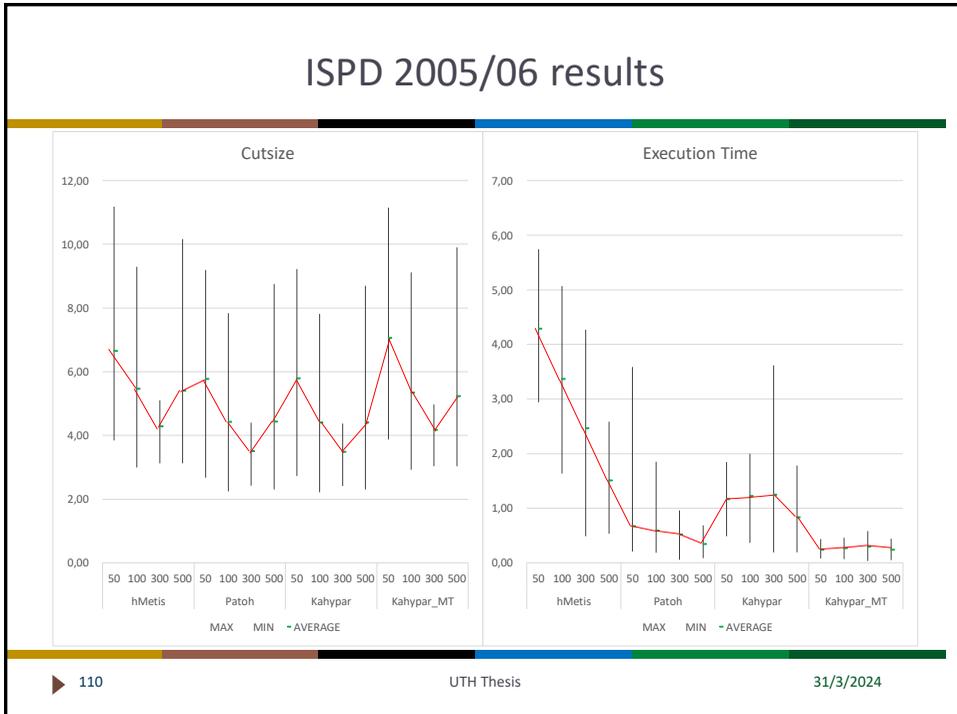
108

## Thesis Outline

1. Definitions 
2. Applications 
3. Existing tools 
4. Our contribution 
5. **Evaluation** 
6. Conclusions 
7. Future work 

▶ 109 UTH Thesis 31/3/2024

109



110

# Physical Design and Timing Oriented Metrics

times x (other / Amass)													
		CutSize				AreaRatio				ExecutionTime			
		50	100	300	500	50	100	300	500	50	100	300	500
hmetis	ISPD06	6.65	5.46	4.29	5.40	3.26	6.24	10.14	3.69	3.99	3.36	2.46	1.51
	ISPD11	3.99	3.02	2.90	2.95	8.37	12.82	11.33	4.38	2.40	2.16	1.50	1.16
	DAC12	3.95	3.23	3.07	3.21	9.40	14.45	10.53	1.78	2.08	1.68	1.38	1.07
	ICCAD15	4.79	3.28	2.35	2.25	5.72	8.07	9.75	4.63	2.09	1.88	1.32	0.90
Patoh	ISPD06	5.77	4.43	3.50	4.43	9.16	1530.30	1277.05	3261.35	0.67	0.59	0.52	0.35
	ISPD11	2.38	1.79	1.87	1.96	4455.35	2157.42	420.45	174.37	0.99	1.27	1.27	0.95
	DAC12	2.39	1.92	1.93	2.06	11065.08	6792.42	4467.30	1530.78	0.90	1.31	0.88	0.80
	ICCAD15	3.31	2.31	1.72	1.67	3396.28	22040.44	12408.99	10977.83	1.02	1.15	0.81	0.54
Kahypar	ISPD06	5.79	4.40	3.49	4.40	11.12	76.64	187.23	268.06	1.17	1.22	1.25	0.83
	ISPD11	2.38	1.75	1.80	1.88	1152.26	10.64	1916.99	673.49	2.01	3.30	5.01	4.98
	DAC12	2.37	1.89	1.88	2.00	838.90	79.69	1300.21	533.32	2.48	3.53	4.52	5.34
	ICCAD15	3.22	2.22	1.66	1.62	19.21	369.73	8158.89	3942.13	4.52	8.25	21.15	28.76
Kahypar_MT	ISPD06	7.06	5.34	4.17	5.24	5.05	7.19	7.77	8.00	0.24	0.27	0.30	0.24
	ISPD11	3.67	2.60	2.45	2.48	11.90	9.09	6.01	4.40	0.21	0.26	0.35	0.35
	DAC12	3.70	2.79	2.55	2.63	12.28	11.53	7.09	2.60	0.17	0.20	0.27	0.31
	ICCAD15	4.73	3.23	2.30	2.20	6.25	7.25	6.29	5.35	0.10	0.11	0.12	0.12
		Top 1000				top 1000 Delay				Top 1000 Slack			
		50	100	300	500	50	100	300	500	50	100	300	500
	hMetis	1.03	0.77	0.63	0.70	7.58	4.78	4.59	4.26	4.52	4.18	4.21	4.18
	Patoh	0.57	0.37	0.26	0.25	2.95	2.22	2.27	2.23	2.97	3.11	3.62	3.77
	Kahypar	0.53	0.35	0.25	0.26	3.02	2.07	2.09	2.05	3.02	3.05	3.31	3.34
	Kahypar_MT	1.06	0.72	0.57	0.59	6.85	4.93	4.46	4.19	4.39	4.18	4.17	4.15

## 2<sup>nd</sup> Paper

# AMASS: Multi-Level Timing Driven Partitioning Algorithm

George Rafael Goudroumanis <sup>ORCID</sup>, Nikolaos Sketopoulos <sup>ORCID</sup>, Christos P. Sotiriou <sup>ORCID</sup>, and Stavros Simoglou  
 University of Thessaly, EECE Department, Volos, Greece  
 {ggeorgios-r, sketopou, chsotiriou, ssimoglou}@e-cc.uth.gr



## Thesis Outline

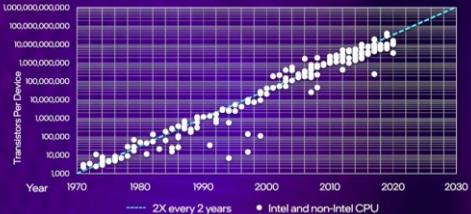
1. Definitions 
2. Applications 
3. Existing tools 
4. Our contribution 
5. Evaluation 
6. Conclusions 
7. Future work 

▶ 113
UTH Thesis
31/3/2024

113

## Conclusions

### Moore's Law is alive and well



Year 1970 1980 1990 2000 2010 2020 2030

--- 2X every 2 years    • Intel and non-Intel CPU

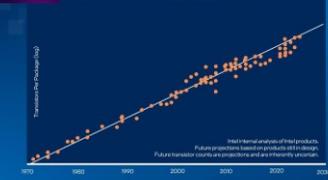
Moore's law isn't really a law in the legal sense or even a proven theory in the scientific sense (such as  $E = mc^2$ ). Rather, it was an observation by the late Gordon Moore in 1965 while he was working at Fairchild Semiconductor: the number of transistors on a microchip (as they were called in 1965) doubled about every year.

Moore went on to co-found Intel Corporation and his observation became the driving force behind the semiconductor technology revolution at Intel and elsewhere.



Gordon Moore in 1965

- ▶ MultiLevel flow will become the new standrad flow
- ▶ Partitioning will become again mandatory in the foreseeable future
- ▶ Area ratio and execution time will be the two key evaluation points of the algorithms
- ▶ Cloud computing will shortly introduced as necessity exploiting the advantages of ML flow



Year 1970 1980 1990 2000 2010 2020 2030

Aspiring to 1 Trillion transistors in 2030

- ✓ RibbonFET
- ✓ PowerVia
- ✓ High-NA
- ✓ 2.5D/3D packaging

intel

▶ 114
UTH Thesis
31/3/2024

114

## Future Work

- ▶ Integrate more advanced algorithms into clustering and partitioning phases
  - ▶ Clustering step
    - ▶ AI/ML oriented algorithms
    - ▶ Network flow based algorithms
  - ▶ Post processing partitioning features like
    - ▶ Gate replication to reduce cutsize
    - ▶ Path distribution removal
    - ▶ Hierarchical ML flow, exploiting the native partitioning of the RTL modules
- ▶ Optimise data structures and update operations to address many millions instances designs

▶ 115

UTH Thesis

31/3/2024

115

## Questions



▶ 116

UTH Thesis

31/3/2024

116

Thank you all for your time



thank  
you  
all

Adobe Stock | #17939402