High-throughput Electron Beam Direct Writing of VIA Layers by Character Projection using Character Sets Based on One-dimensional VIA Arrays with Area-efficient Stencil Design

Rimon IKENO[†], Takashi MARUYAMA[††], Tetsuya IIZUKA[†††], Satoshi KOMATSU[†], Makoto IKEDA[†], and Kunihiro ASADA[†]

[†]VLSI Design and Education Center (VDEC), The University of Tokyo,
[††]e-Shuttle, Inc.,
[†††]Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo

東京大学 THE UNIVERSITY OF TOKYO VDEC

Agenda

- **Introduction**
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- **Problems in applying CP to VIA layer exposure**
  - VIA number, density, and arrangement in actual designs
- **Approach to improve VIA CP throughput**
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- **Experimental results**
  - CP throughput evaluation
  - CP throughput improvement
- **Summary**

Agenda

- **Introduction**
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- **Problems in applying CP to VIA layer exposure**
  - VIA number, density, and arrangement in actual designs
- **Approach to improve VIA CP throughput**
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- **Experimental results**
  - CP throughput evaluation
  - CP throughput improvement
- **Summary**

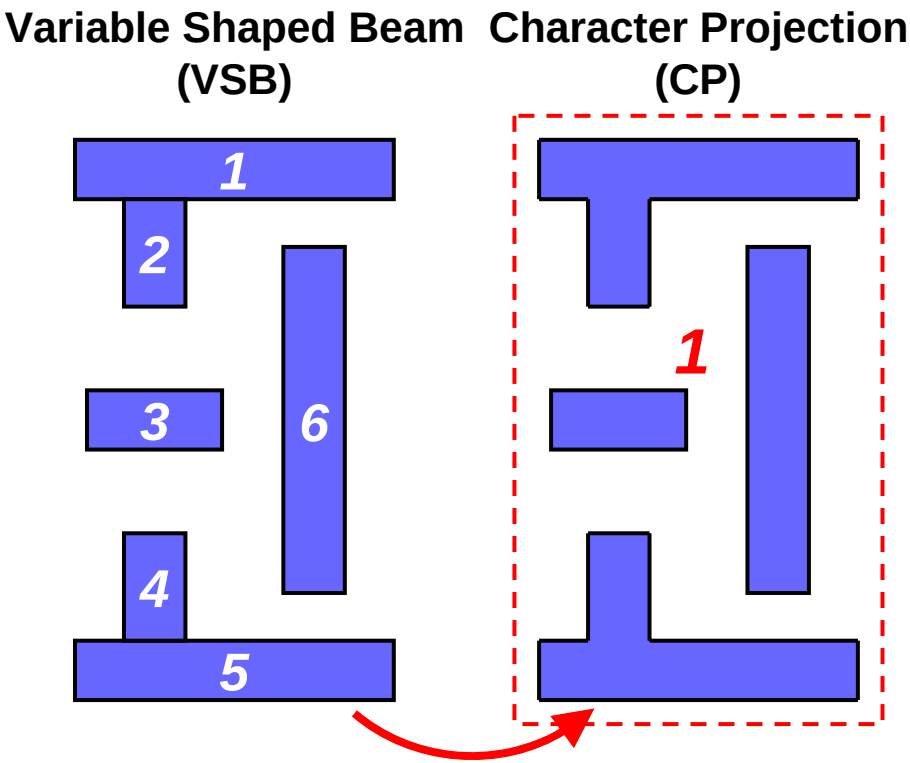
Introduction

- **Electron-Beam Direct Writing (EBDW)**
  - **Low-cost** ‘maskless’ exposure solution
    ← Mask cost explosion in high-resolution lithography
  - **Low exposure throughput**
    ← 1 rectangle / 1 EB shot (Variable Shaped Beam; VSB)
  - **Limited applications**: low-volume products, test chips, ...

- **Character Projection（CP）method in EBDW**
  - Promising approach for **high-throughput** EBDW
  - Shooting **multiple figures** at once as a **character**
  - **Characters** = “Frequently-appeared layout patterns”
    → Prepared on **character stencil**
  - For higher CP throughput → more figures in 1 EB shot
    → **More character varieties required!**

Overview of Character Projection (CP) EBDW

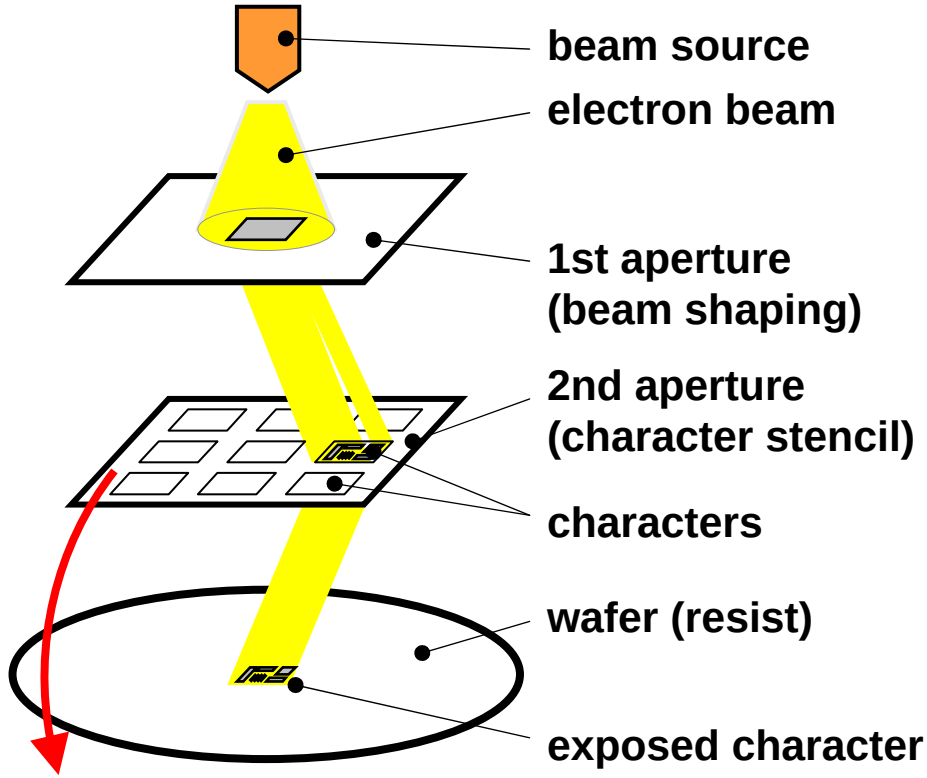
Shot count comparison



6x faster throughput (E_{CP} = 6.0)

CP efficiency: E_{CP}
≡(VSB shot count)/(CP shot count)

CP EBDW equipment



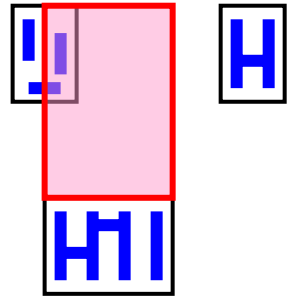
Limited stencil area:

→*Limited character variety (number)*

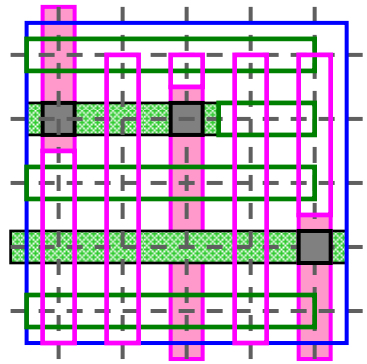
Constraint on throughput improvement

Earlier Efforts for Higher CP Throughput

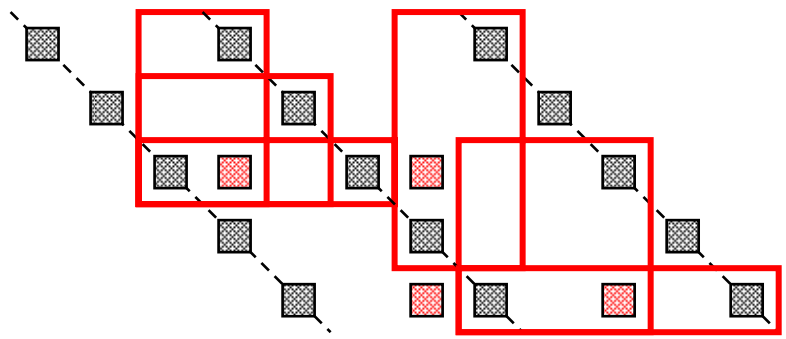
- **Cell layer (transistors & local wires)**
  - 1 cell/shot (major cells only) ➔ E_{CP} >10
  - >1 cells/shot (cell cluster → character) cluster extraction / rearrangement



- **Metal layers (interconnect routing)**
  - Tile routing: normalized wire segments
  - Matching to predefined characters



- **VIA layers (interconnect routing)**
  "High degree of freedom"
  - Area-efficient stencil design (Du et al./ASP-DAC2012)
    → Just 3 VIAs in 1 character
    ➔Further improvement?



Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- **Problems in applying CP to VIA layer exposure**
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary

VIA Placement Study in Actual Logic Designs

P&R results of example designs（65nm）

Design	Area [um^2]	Cell#	Cell pin#	VIA1#	VIA1 dens [um^{-2}]
8080 CPU	11,837	4,293	14,623	14,520	1.23
USB 2.0	29,309	7,725	28,414	28,301	0.97
AES Core	60,270	17,328	56,513	56,744	0.94
DCT Core	63,907	13,800	50,938	50,786	0.79
SPARC V8	1,191,590	229,580	929,474	927,562	0.78
SPARC T1	5,933,609	810,278	3,464,043	3,457,402	0.58

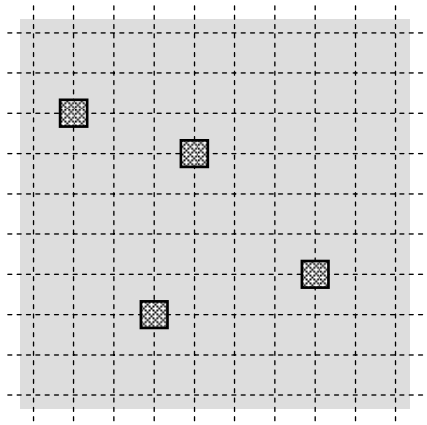
- **Observations: VIA# and density in actual designs**
  - VIA1# ~ cell pin# > cell# → requiring more E_{CP} than cells
  - VIA1 density ~ 1um^{-2} → ~1 VIA in 25 grids (1 grid=0.2um)

Problems in Applying CP to VIA Layers

- **Requirements for a VIA CP character set**
  - All possible combinations of multiple VIA arrangement
  - More VIAs in a character for higher CP throughput
  - Integrated within a limited stencil area
- **Reality of the VIA character set planning**

~1 VIA in 25 grids

A 4-VIA character occupies 25x4=100 grids in average



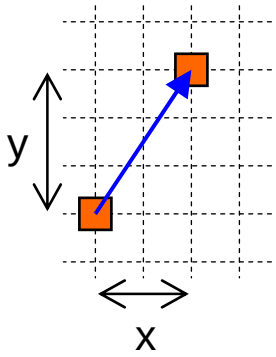
Character number = Combinatorial number of VIA placement

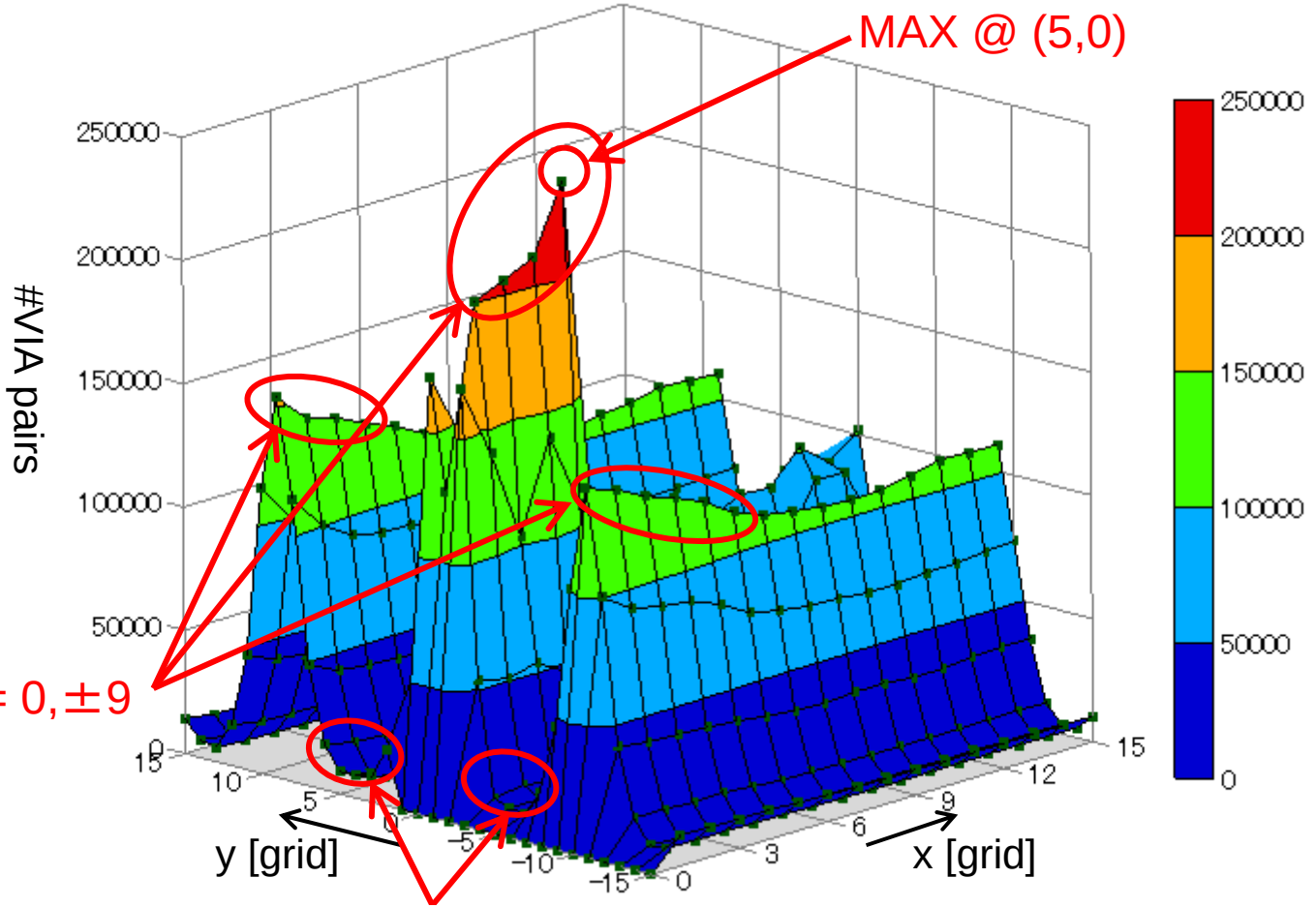
$${}_{100}C_4 = 3{,}921{,}225$$

  - All possible VIA placement: **character number explosion**
- **To achieve higher E_{CP} within stencil area limitation…**
  - Reduce character variations even with more VIAs
  - Detailed VIA placement study (*next page*)

Detailed Study on VIA Arrangement (VIA1)

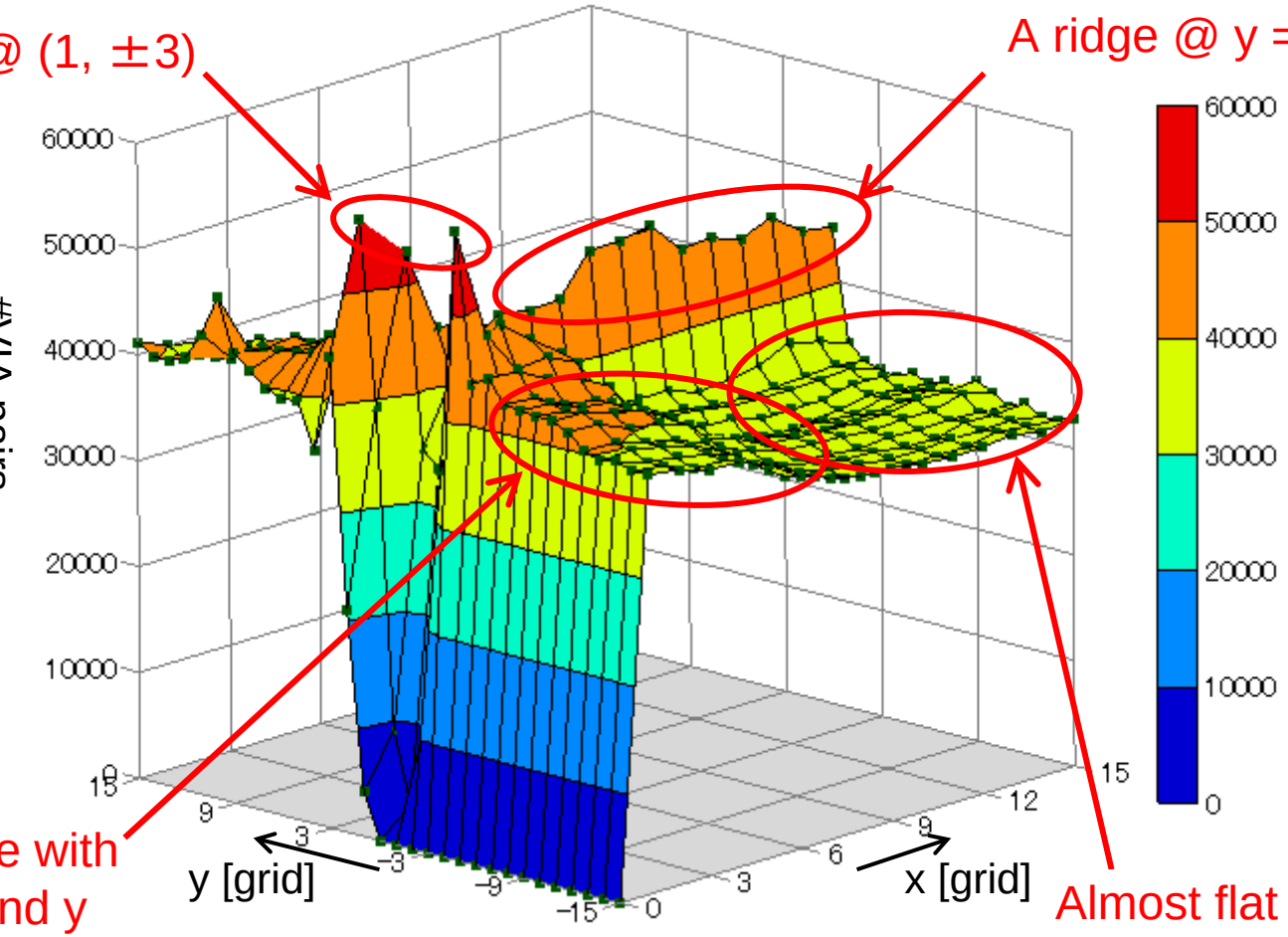
Relative positions of neighboring VIAs (USB 2.0, VIA1 layer)





Detailed Study on VIA Arrangement (VIA2)

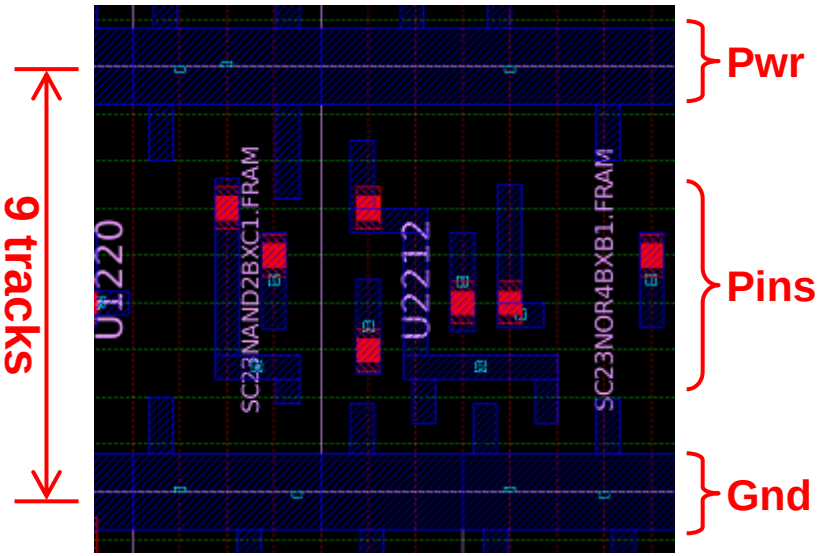
Relative positions of neighboring VIAs (USB 2.0, VIA2 layer)

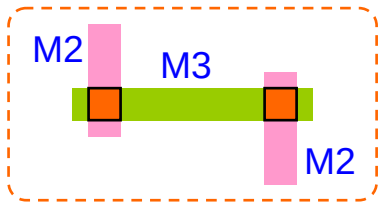


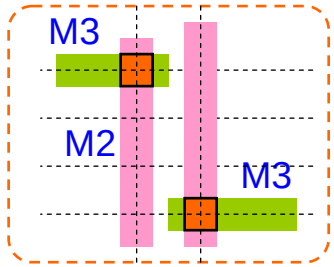
Considerations on VIA Arrangement

- **VIA1 layer (cell pin to interconnect):**
  ridges at y=0, ±9, valleys at y=±4~5
  - The library cell architecture (0.2um x 9 tracks) allows cell pin access (connection) to only 4 tracks in the center
  - VIAs are arranged as stripes to avoid power/ground lines
  - → VIA placement has strong correlations in the X direction
- **VIA2 layer**:
  a ridge at y=0, peaks at (1,±3)
  - Many cases of the right figures, but also other random cases

Library cell architecture & VIA1 placement for pin access





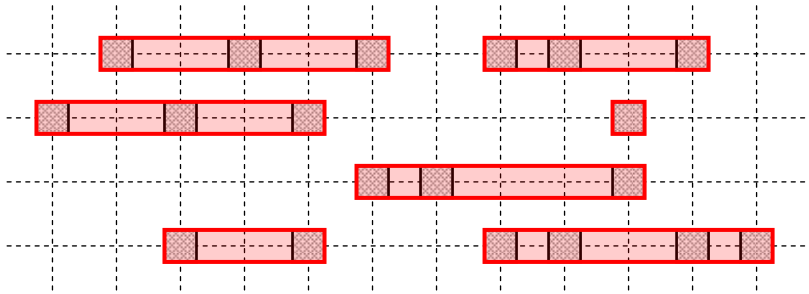


Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- **Approach to improve VIA CP throughput**
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary

One Dimensional VIA Array Characters

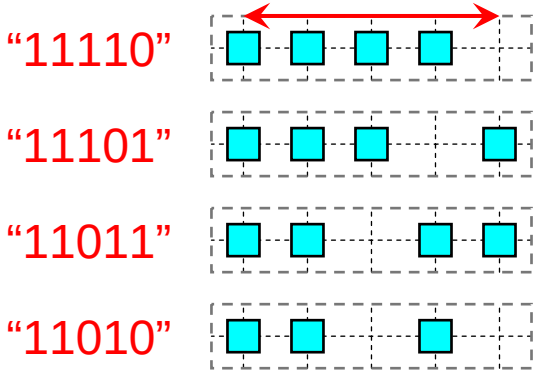
- **Only allow one-dimensional VIA arrays as characters**
  - VIA position candidates are reduced
    → **CP character variation decreases**



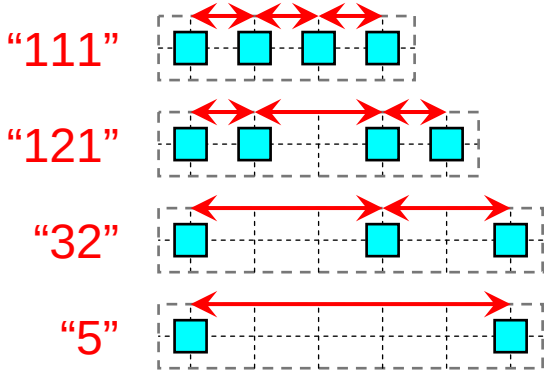
7 shots / 19 VIAs
(2.7 VIAs/shot)

- **Two options in VIA array character coding schemes**

Binary coding
(fixed character size)



Span coding
(limited character size & VIA#)



Area-efficient Character Stencil Design

Example
Span coding characters
VIA number: N =4
Max size: M = 6

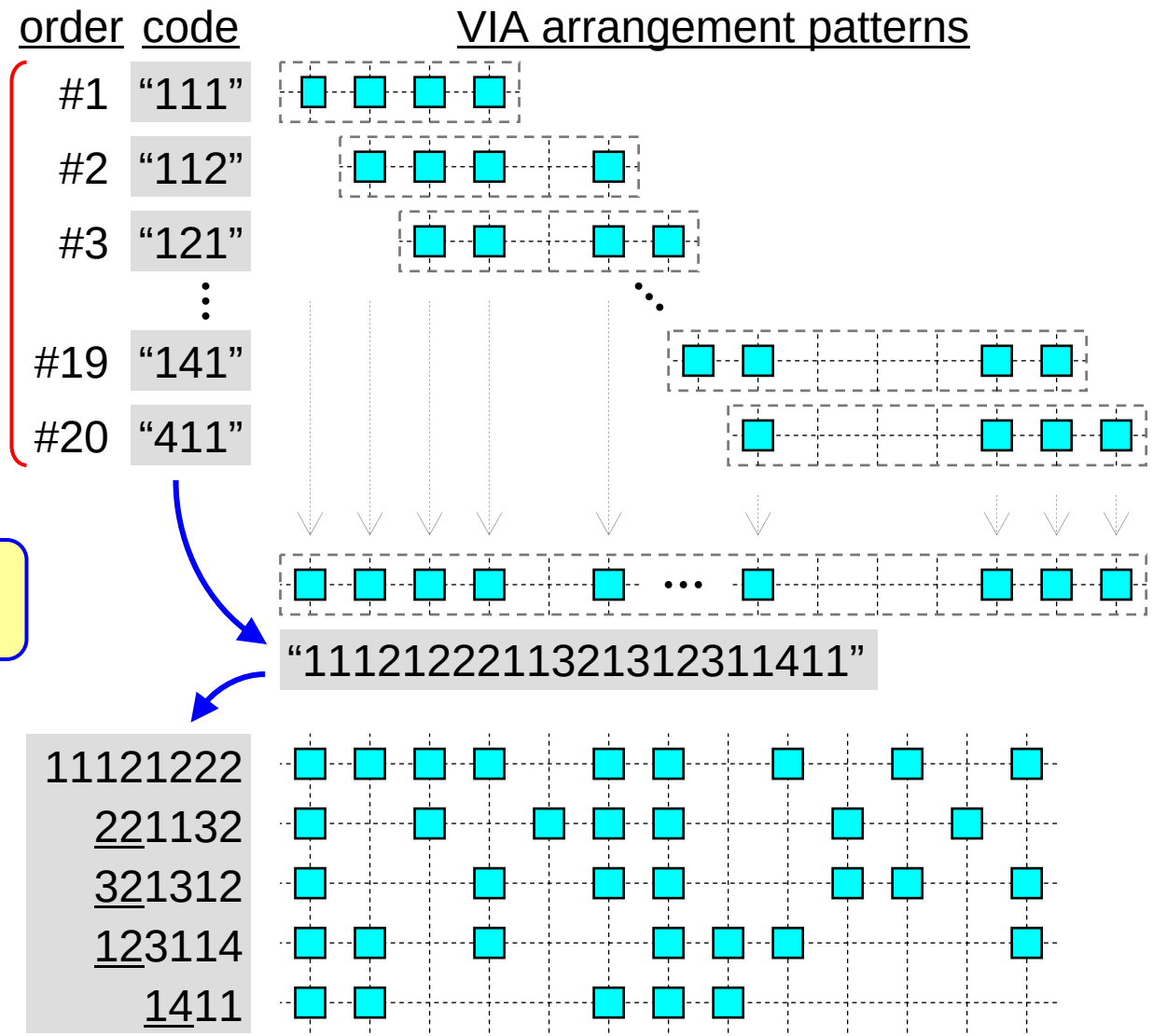
VIA array characters

Character ordering & superposition

Compressed VIA array

VIA array division & placement

Character stencil



Stencil Design under Area & Size Constraints

- **Character set planning for stencil design to be adopted to 14nm CP-EBDW equipment specifications**

Stencil area: 3.6 x 10^6 grids / Shot size: 30 grid x 30 grid

Binary coding character

- average character size = 1 grid (after compression)
- character number = 2^{M+1} (M: character size)
- $3.6 \times 10^6 < 2^{M+1}$ → **M=20**

Span coding character

- Some options of max VIA number (N) & max character size (M) combination →

Max VIA# (N)	Max char size (M)	Character number	Stencil area [grid]	Comp. ratio
7	30	621,616	2,894,892	5.80
8	26	726,206	2,625,930	6.65
9	24	880,970	2,689,536	7.43
10	23	1,097,790	2,989,405	8.13
12	22	1,401,292	3,319,648	9.19
21	21	1,048,576	2,330,190	9.62

Layout Constraints for More CP Improvement

- **Further improvement of CP throughput?**

Re-consideration on VIA arrangement:

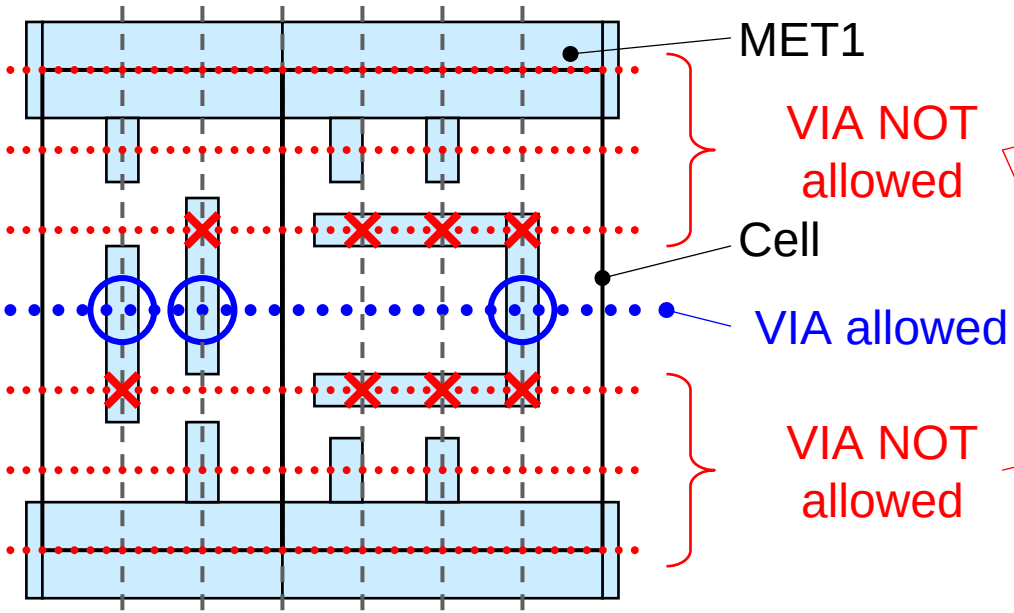
More VIAs on specific tracks

→ More VIAs in each shot & less shots for vacant rows

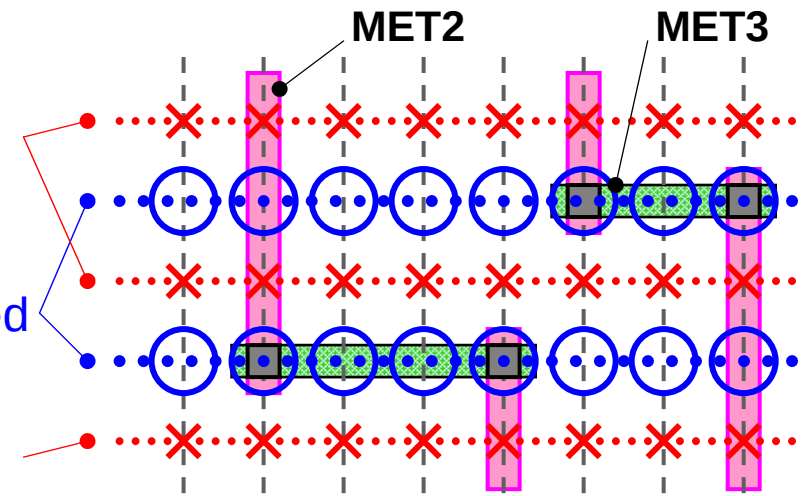
How?

Layout constraints for VIA placement (in detail routing)

VIA1 Layer constraint



VIA2 Layer constraint



Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- **Experimental results**
  - CP throughput evaluation
  - CP throughput improvement
- Summary

Experiments and CP Throughput Evaluation

- **Experiments:**
  - Automated place & route of 6 example circuits
  - ***with*** and ***without*** the **layout constraints**
    to evaluate impact of introducing the layout constraints

- **CP performance estimation:**
  - **Shot count estimation**

$$(\text{Gshot/wafer}) = \frac{10.0 \times 100}{(\text{layout area})} \times (\text{shot\#})$$

average shot area (100Gs/w)
× target shot count
= total shot area

- **Evaluation criteria**
  - **Shot count target** for practical use of CP-EBDW @14nm
    →**100~173 Giga shot/wafer**
    - 173 Gs/w: the minimum throughput requirement
    - 100 Gs/w: the target to allow margins of other factors

Results (1): CP Shot Count Estimation

VIA1 shot# with/wo layout constraint (OpenSPARC T1 Core)

			w/o layout constraints			With layout constraints		
Character set	N	M	Shot#	VIA# /shot	G shot /wafer	Shot#	VIA# /shot	G shot /wafer
Binary coding		20	1,677,663	2.06	282.7	1,032,441	3.32	174.0
Span coding	**7**	**30**	**1,301,248**	**2.66**	**219.3**	**858,330**	**3.99**	**144.7**
	8	26	1,437,692	2.40	242.3	916,924	3.74	154.5
	9	24	1,570,835	2.20	264.7	986,128	3.48	166.2
	10	23	1,597,913	2.16	269.3	994,780	3.44	167.7
	12	22	1,627,868	2.12	274.3	1,005,818	3.41	169.5
	21	21	1,651,097	2.09	278.3	1,017,748	3.37	171.5

- The layout constraints improved VIA#/shot (=CP efficiency) from 200~300 Gs/w range to 140~175 Gs/w range
- Span-coding (N=7, M=30) achieved the best performance

Results (2): CP Shot Count Summary

VIA1/2 shot# of 6 example circuits with layout constraints

	VIA1		VIA2	
Design	VIA# /shot	#shot /wafer	VIA# /shot	#shot /wafer
8080 CPU	5.94	205.7	3.11	311.9
USB 2.0	6.09	156.2	2.77	272.8
AES Core	5.64	161.6	2.76	266.7
DCT Core	4.92	159.2	2.52	248.3
SPARC V8	4.71	162.9	2.64	269.2
SPARC T1	3.99	144.7	2.31	236.1
Average	**4.21**	**148.1**	**2.38**	**242.1**

Span-coding:
max VIA#: N= 7
max size: M = 30

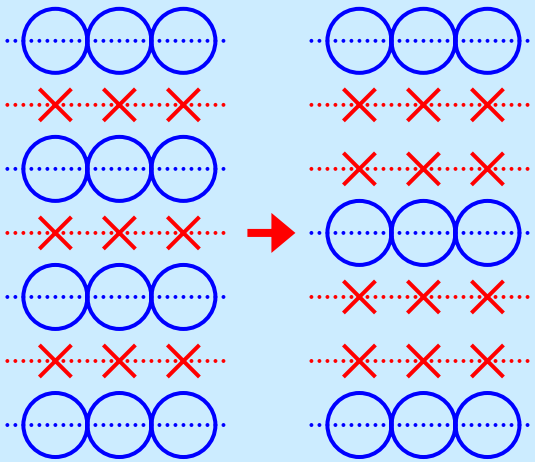
- VIA#/shot: 4~6 (VIA1), 2.3~3.1 (VIA2)
- Shot#/wafer 148.1 Gs/w (VIA1) → **Target achieved**
  242.1 Gs/w (VIA2) → **Not achieved**

Results (3): Improvement by Tighter Constraints

Further VIA2 throughput improvement?

VIA2 track utility

50% → **33%**



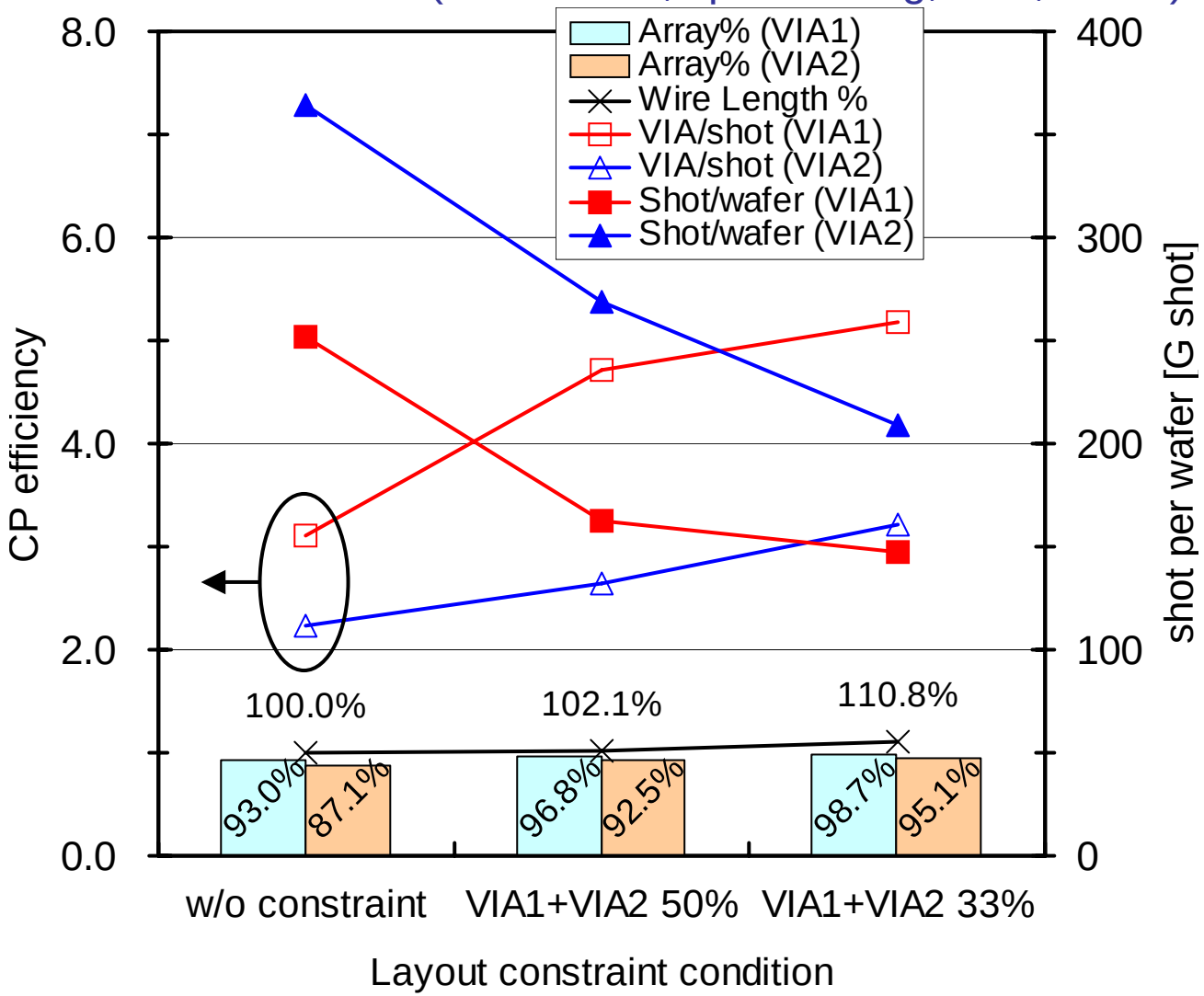
VIA2 shot/wafer

270 Gs/w → **208 Gs/w**

Wire length increase

2% → **11%**

(SPARC V8, span-coding, N=7, M=30)



Agenda

- **Introduction**
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- **Problems in applying CP to VIA layer exposure**
  - VIA number, density, and arrangement in actual designs
- **Approach to improve VIA CP throughput**
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- **Experimental results**
  - CP throughput evaluation
  - CP throughput improvement
- **Summary**

Summary

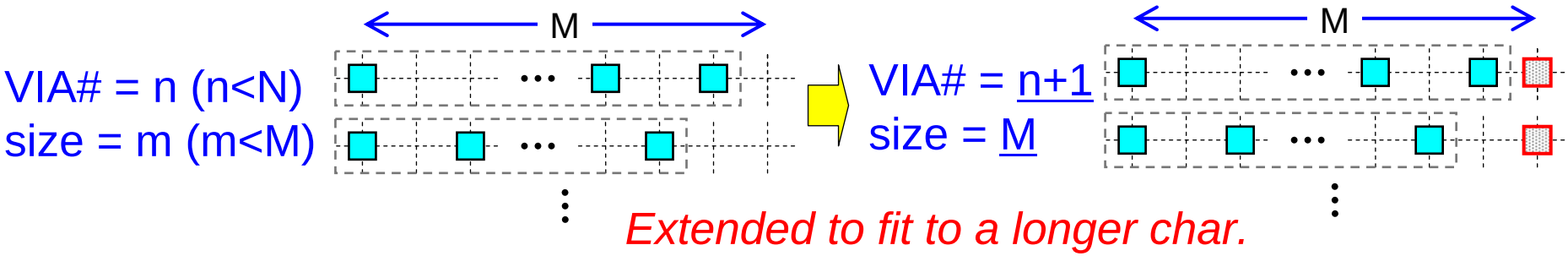
- **Difficulty in applying CP-EBDW to VIA layers**
  - Character explosion when increasing VIA number
  - VIA # / density / arrangement studies in actual designs
- **One-dimensional VIA array characters**
  - Character number suppression by limited arrangement
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- **Experimental results and conclusions**
  - **Good CP throughput** by span coding character set
  - **Improved CP throughput** with little interconnect impact by tighter layout constraints
  - VIA1: <150 Gs/w → **target achieved** (100~173 Gs/w)
  - VIA2: ~200 Gs/w → slightly missed / tuning constraints?

Thank you!

Appendices

Required Character Sets for Span Coding

- Characters with size<M are parts of characters with size=M



Extended to fit to a longer char.

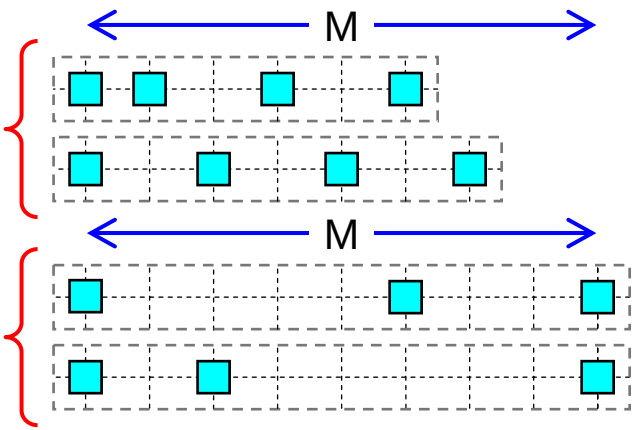
- Required character sets are only following two groups

VIA number		2	3	…	N-2	N-1	N
Character size	<M	○	○	…	○	○	○
	M	○	○	…	○	○	○

Group 1: VIA# = N, size ≤M

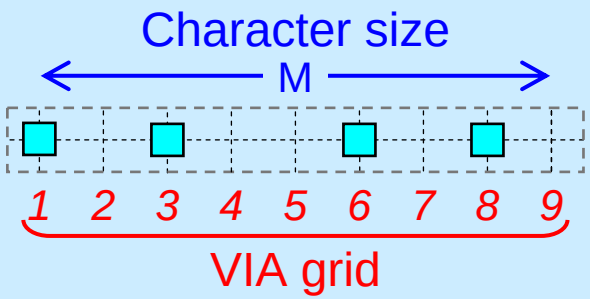
Group 2:VIA# <N, size = M

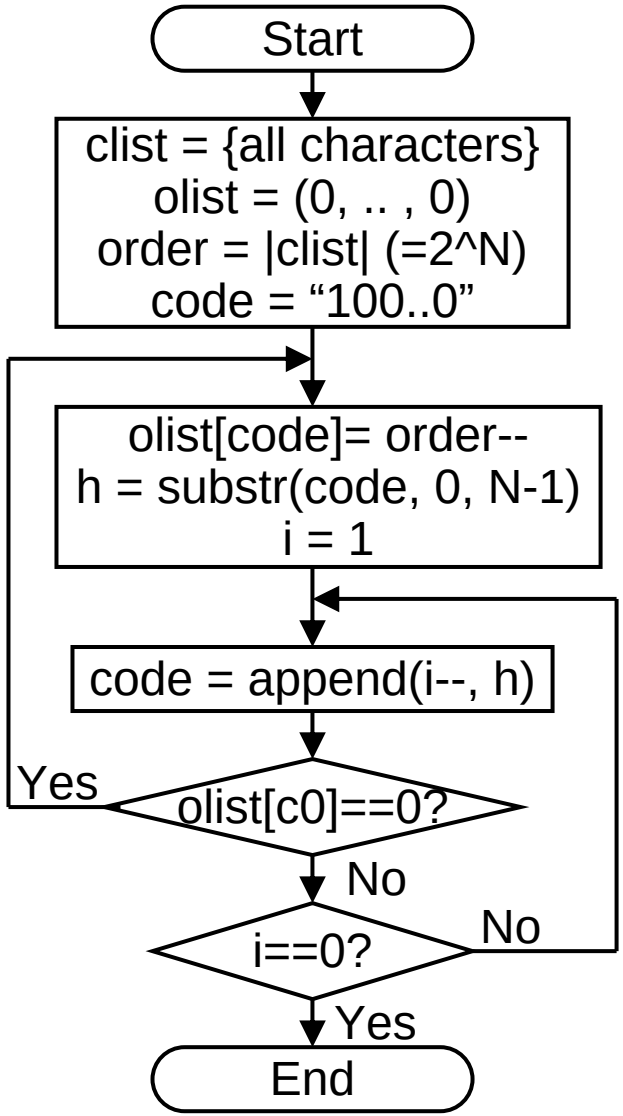
Example: N=4, M=8



Example of Character Ordering Algorithm

Example
Binary coding characters
Max VIA number: N
(= Char size : M + 1)





- *Gen character list*
- *Initialize order storage*
- *Get the last code and its order (= char#)*
- *Set character order*
- *Extract next char head*
- *Next char tail candidate*
- *Gen next char candidate*
- *If not ordered yet, go back to set the order*
- *If already ordered, try another candidate*

Details of Character Ordering in Page 15

Span coding character set (VIA number: N =4, Max size: M = 6)

No.	Character			Order																					
1	1	1	1	1	1	1	1																		
2	1	1	2	2		1	1	2																	
3	1	1	3	9									1	1	3										
4	1	1	4	18																	1	1	4		
5	1	2	1	3			1	2	1																
6	1	2	2	5					1	2	2														
7	1	2	3	15														1	2	3					
8	1	3	1	13												1	3	1							
9	1	3	2	10										1	3	2									
10	1	4	1	19																		1	4	1	
11	2	1	1	8								2	1	1											
12	2	1	2	4				2	1	2															
13	2	1	3	12											2	1	3								
14	2	2	1	7							2	2	1												
15	2	2	2	6						2	2	2													
16	2	3	1	16															2	3	1				
17	3	1	1	17																3	1	1			
18	3	1	2	14													3	1	2						
19	3	2	1	11											3	2	1								
20	4	1	1	20																			4	1	1
Compressed VIA array					1	1	1	2	1	2	2	2	1	1	3	2	1	3	1	2	3	1	1	4	1	1

Results (4): VIA Character Usage Trend vs. Area

(SPARC V8, span-coding, N=7, M=30)

Ratio of actually-used characters as a function of design area (cumulative statistics by multiple ICC runs)

VIA character usage is almost proportional to (design area)$^{1/2}$

VIA characters appear randomly

All possible VIA arrangement must be prepared as CP characters