

OUTLINE

- 1) Introduction
- 2) Objective
- 3) Some key words
- 4) How it works ?
- 5) Example

INTRODUCTION

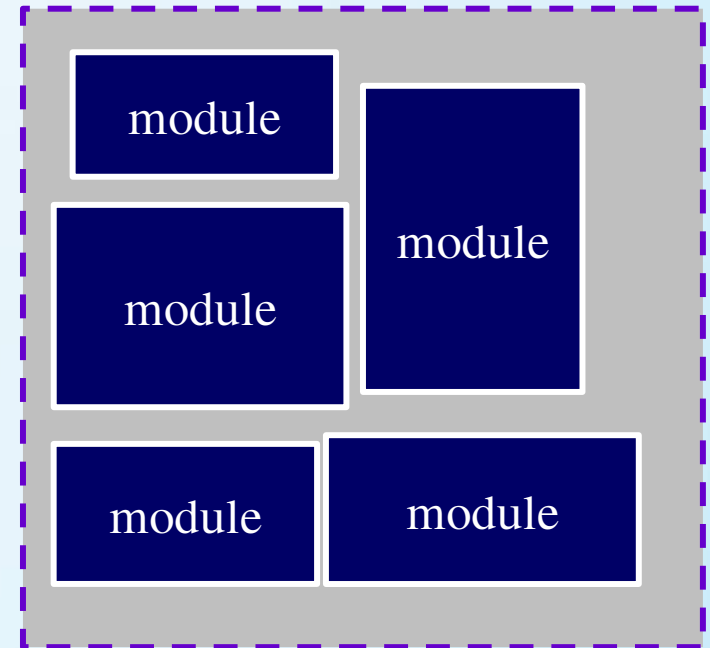
- ▶ Neighbor is a person who lives next to us
- ▶ Neighborhood means
 - a community in a town or city
 - created by social interaction between a group of people

INTRODUCTION

Neighbor – modules

City/Town – area

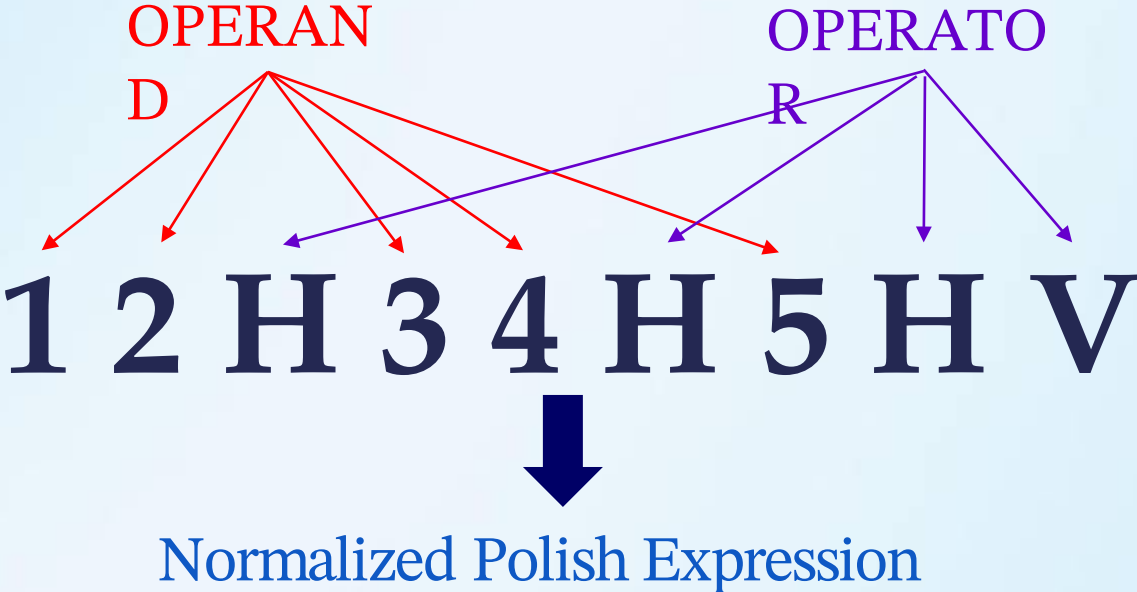
Neighborhood – interconnection
between the modules



OBJECTIVE

Changing a normalized polish expression through perturbation to obtain its corresponding slicing floorplans with zero dead space

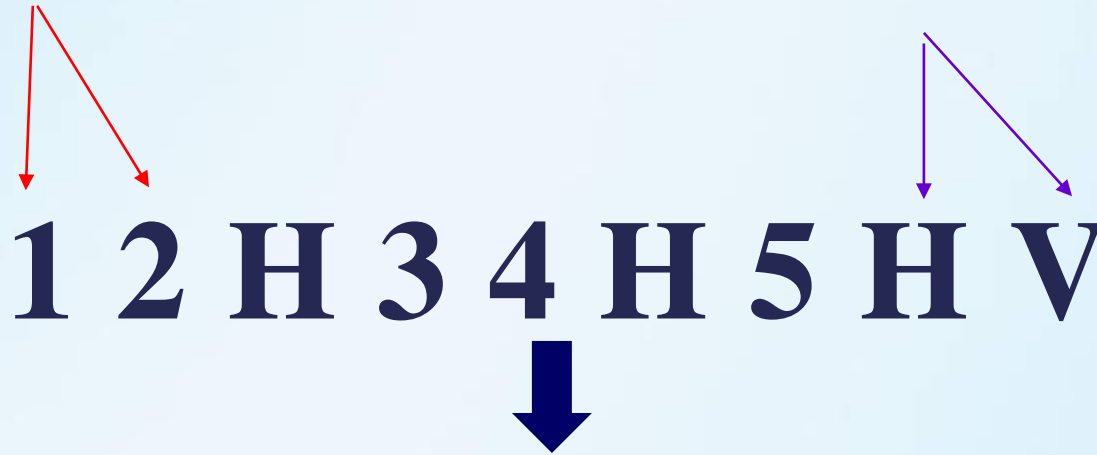
SOME KEY WORDS



SOME KEY WORDS

ADJACENT OPERAND

ADJACENT OPERATOR

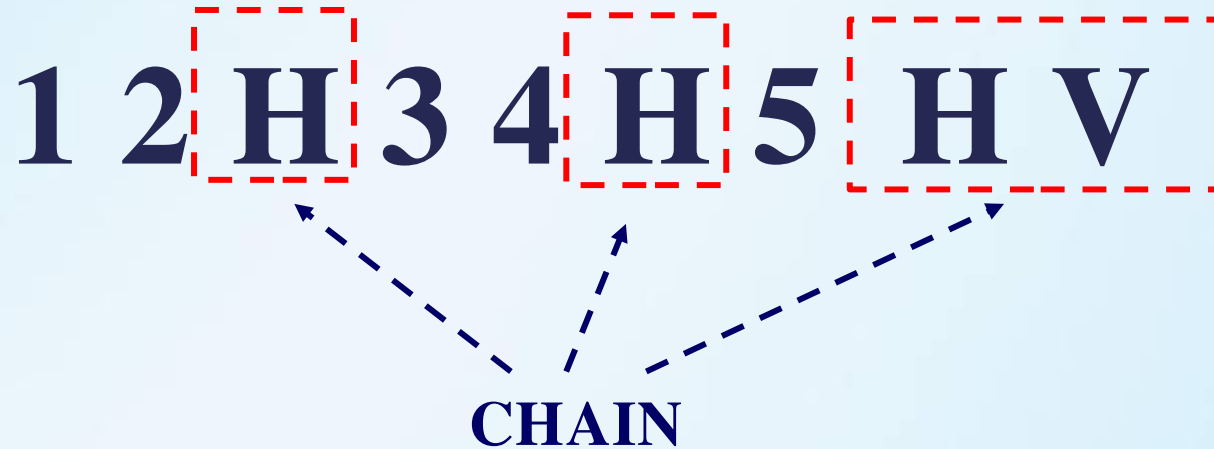


Normalized Polish Expression

Adjacent Operand : No operands between two operands

Adjacent Operator : No operands between two operators

SOME KEY WORDS



No Consecutive Operators of Same Type

HOW IT WORKS ?

3 types of moves –

M1 : Operand Swap

Swap two adjacent operands

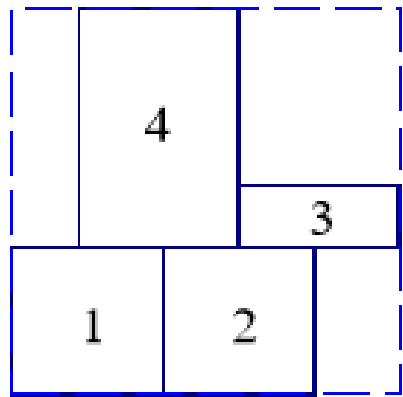
M2 : Chain Invert

Invert a chain by changing V to H and H to V

M3 : Operator/Operand Swap

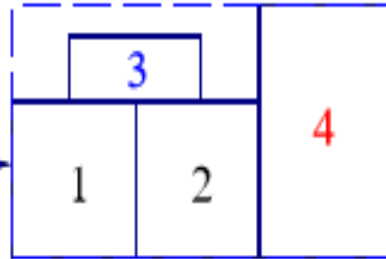
Swap two adjacent operands and operators

EXAMPLE



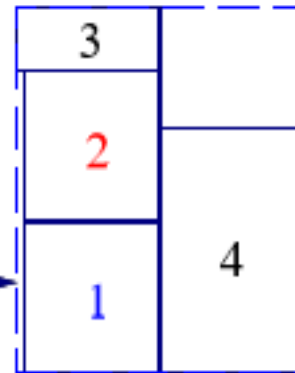
12V4H3V

M1



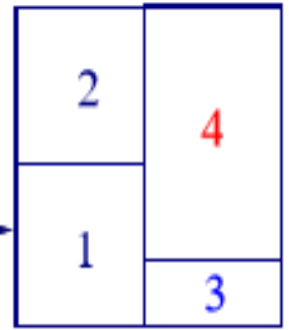
12V3H4V

M2



12H3H4V

M3



12H34HV

CHECKING LEGALITY

Balloting Property :

- M1 and M2 moves are OK
- M3 could make the number of operands not greater than that of operators, which violates the balloting property

CHECKING LEGALITY

Balloting Property :

	<i>1</i>	<i>2</i>	<i>H</i>	<i>3</i>	<i>4</i>	<i>H</i>	<i>5</i>	<i>H</i>	<i>V</i>
No. of operands	1	2	2	3	4	4	5	5	5
No. of operators	0	0	1	1	1	2	2	3	4

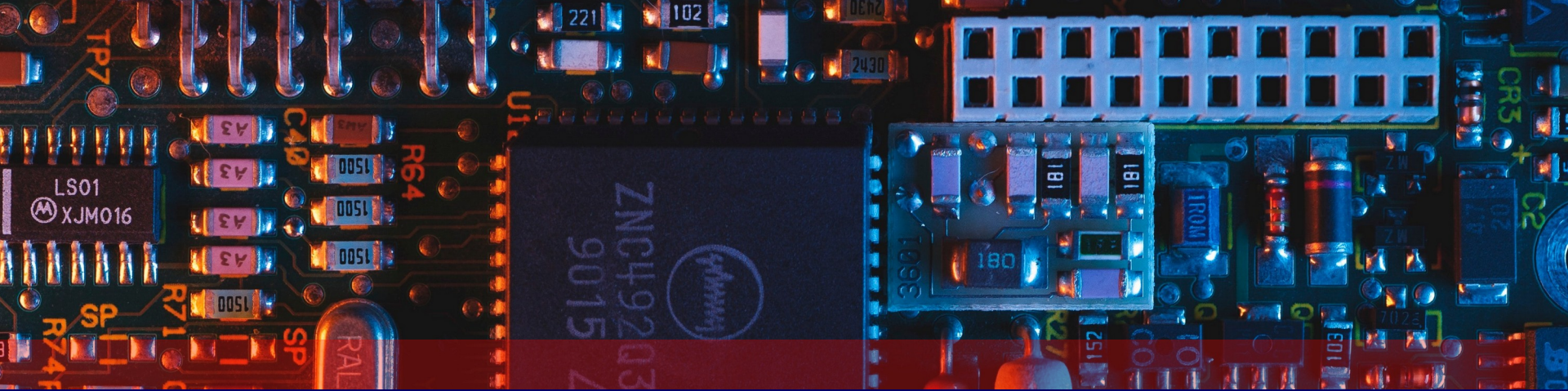
CHECKING LEGALITY

No Consecutive Operators :

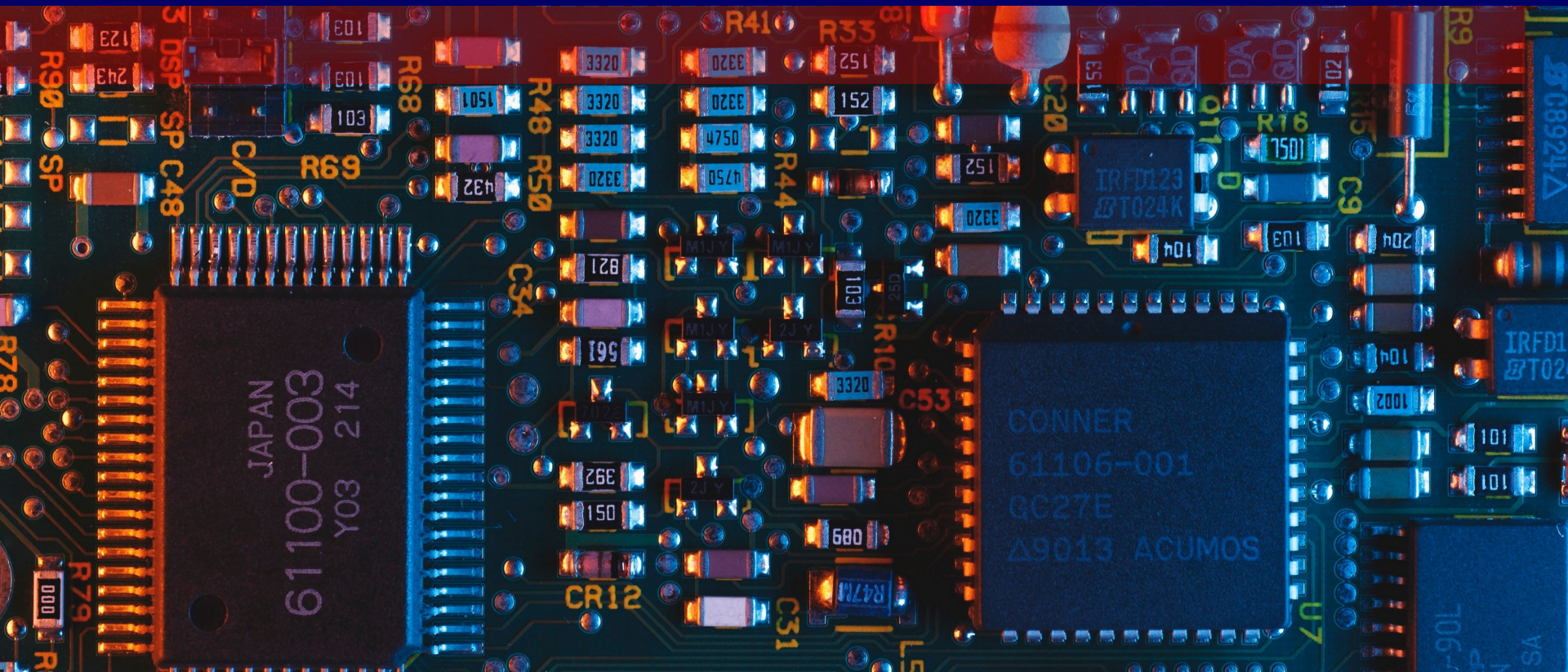
- M1 and M2 moves are OK
- M3 could generate two identical consecutive operators that violates the legal property of normalized polish expression



THANK YOU



Estimating cost of a floorplan



Topics will be discussed –

- » Introduction
- » Objective
- » Cost Function
- » Computation

INTRODUCTION

Cost of floorplanning :

- » The number of feasible solutions of a floorplanning
- » Measure the quality of floorplans

OBJECTIVE

Finding the best solution with minimized cost
for floorplanning
to optimize the performance of the circuits

Cost function

Two most commonly used cost criteria –

1. Minimizing Area
2. Minimizing total Wirelength

Cost function

» The function is (based on area and wirelength) –

$$\varphi = A + \lambda W$$

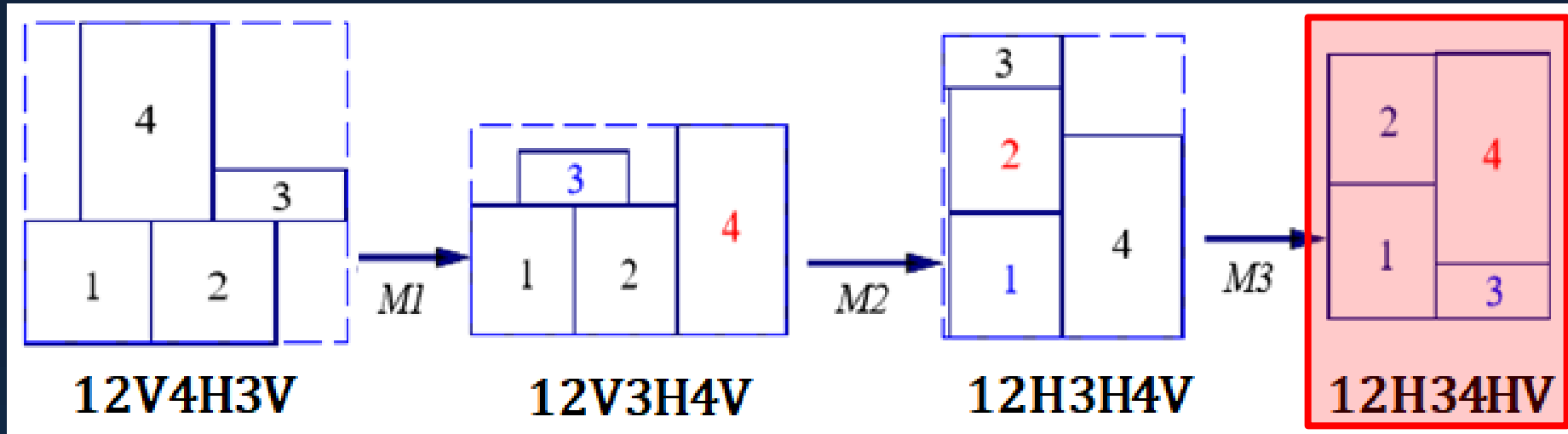
φ = cost for a specific floorplan

A = area of the smallest rectangle

W = overall wiring length

λ = user specified parameter

COMPUTING AREA



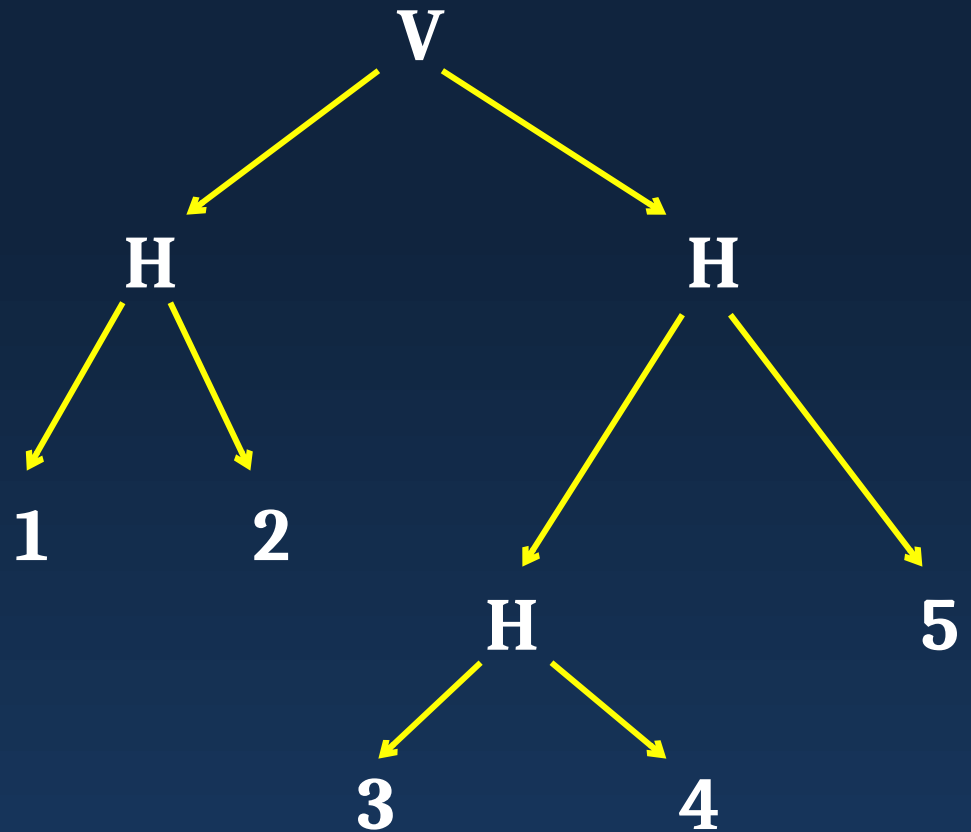
A (the minimized area with zero dead space) : 12H34HV

COMPUTING AREA

Step 1 : Constructing the corresponding skewed slicing tree

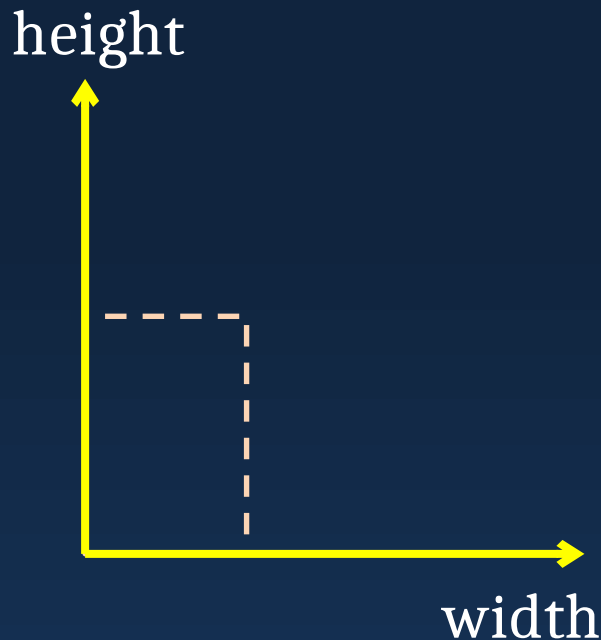
Normalized
Polish
Expression

12H34H5HV



COMPUTING AREA

Step 2 : Placing the dimensions of the modules from given information



Dimensions of the modules		
Module No.	Width (w)	Height (h)
1	4	6
2	4	4
3	3	4
4	4	4
5	3	4

Dimension of a module : (w , h)

COMPUTING AREA

Step 2 : Placing the dimensions of the modules from given information

One thing to notice –

- Module rotation is allowed
- Two types implementation

For example :

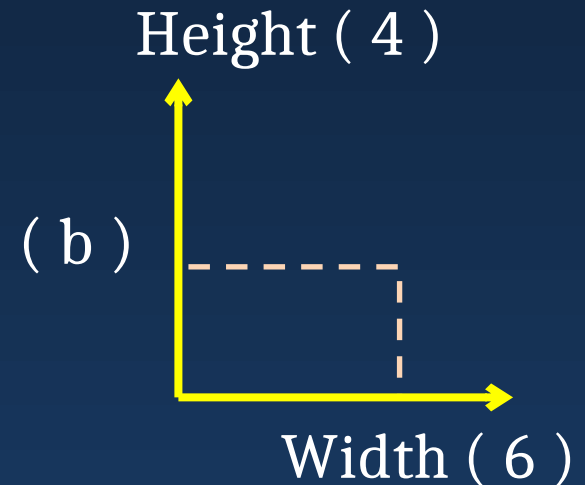
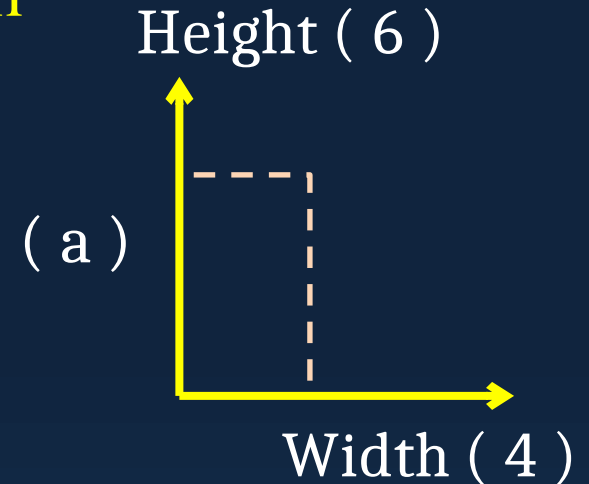
(a) from given table –

Dimension of module 1 : (4 , 6)

(b) by rotating –

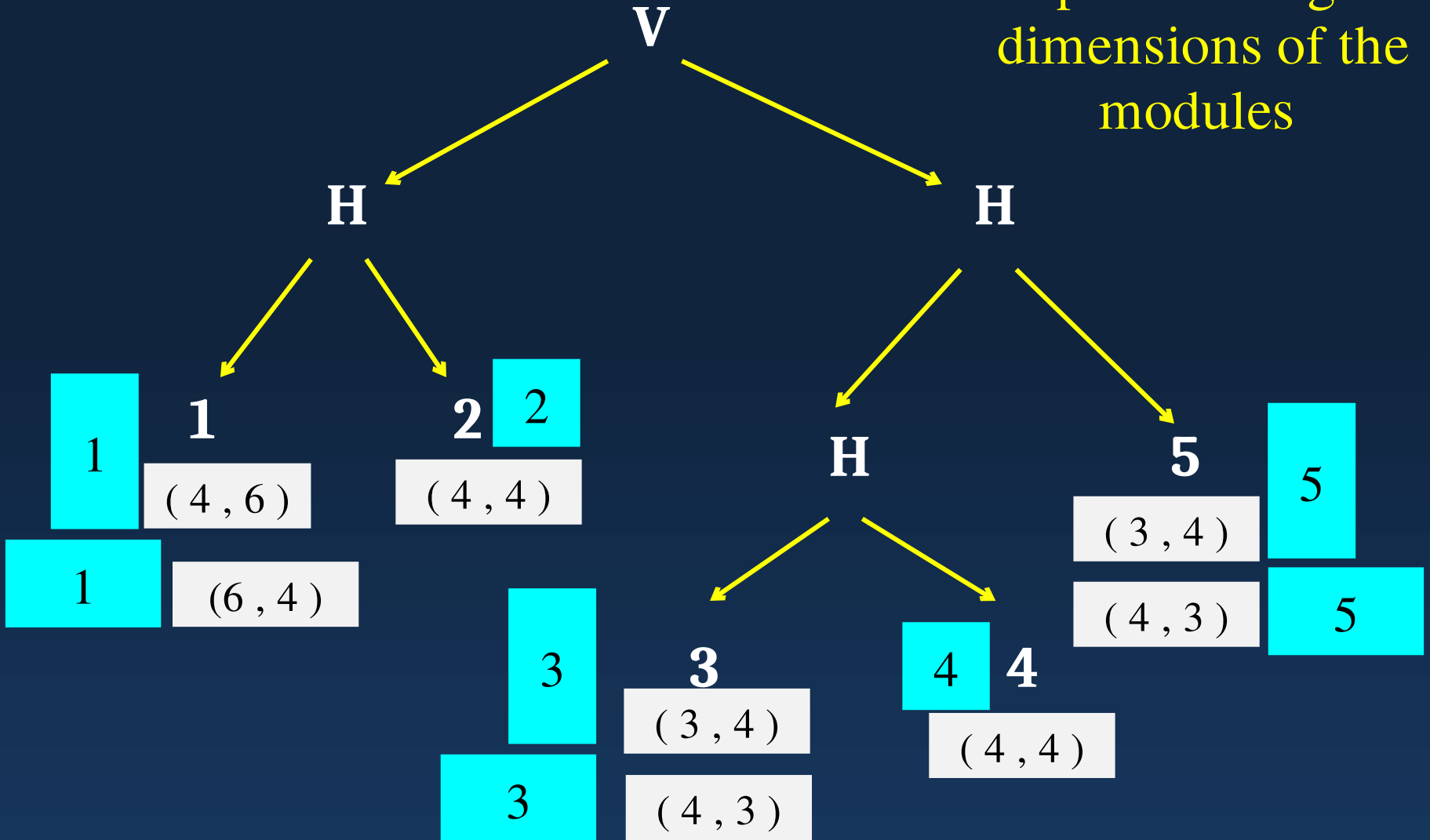
Dimension of module 1 : (6 , 4)

(width = height = 6 ; height = width = 4)



COMPUTING AREA

Step 2 : Placing the dimensions of the modules



COMPUTING AREA

Step 3 : Computing dimension based on the slicing tree

Let , left module = 1 , right module = 2

For **HORIZONTAL** cut :

Dimension of a module = (**max (w1 , w2) , h1+h2)**)

For **VERTICAL** cut :

Dimension of a module = (**w1+w2 , max (h1 , h2)**)

COMPUTING AREA

$$H = (\max (w1 , w2) , h1+h2)$$

34H

(1) Dimension of module 3 :

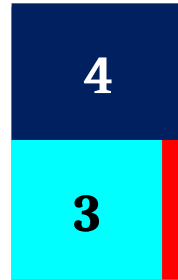
$$w1 = 3 , h1 = 4$$

Dimension of module 4 :

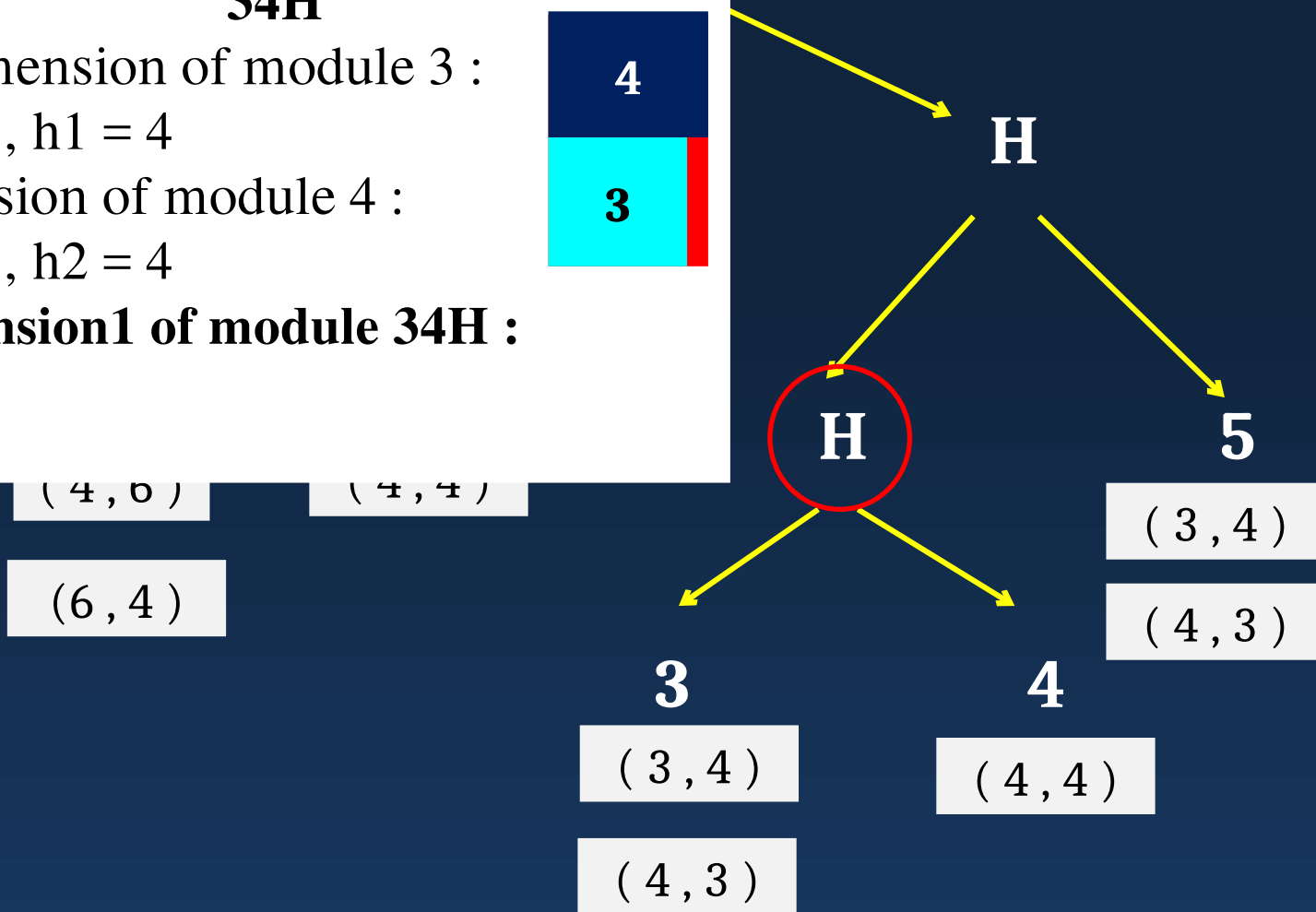
$$w2 = 4 , h2 = 4$$

Dimension1 of module 34H :

$$(4 , 8)$$



Step 3 : computing dimensions



COMPUTING AREA

$$H = (\max (w1 , w2) , h1+h2)$$

34H

(2) Dimension of module 3 :

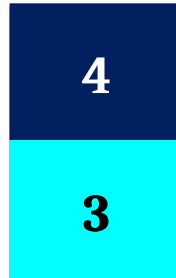
$$w1 = 4 , h1 = 3$$

Dimension of module 4 :

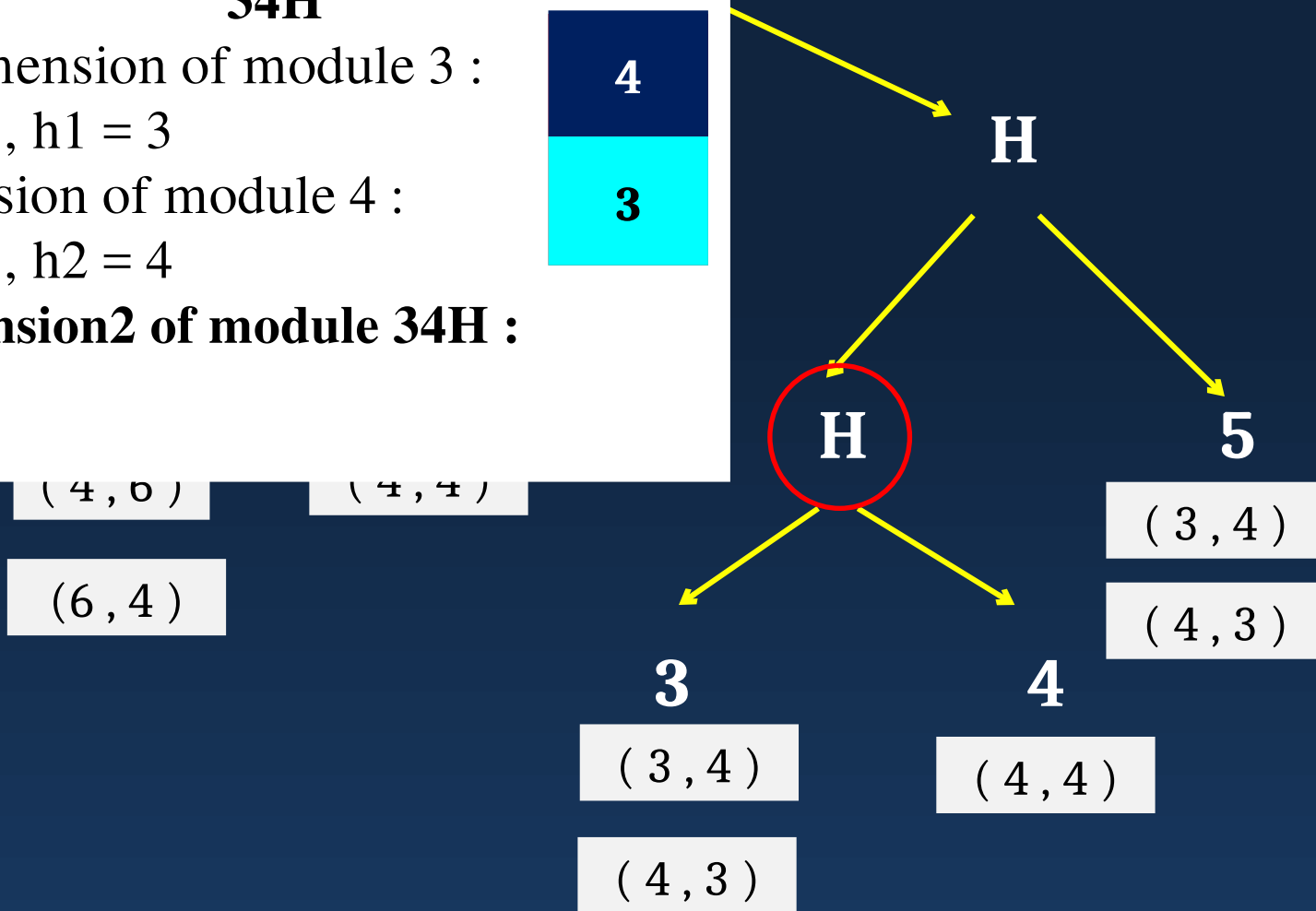
$$w2 = 4 , h2 = 4$$

Dimension2 of module 34H :

$$(4 , 7)$$



Step 3 : computing dimensions



COMPUTING AREA

$$H = (\max (w1 , w2) , h1+h2)$$

34H5H

(1) Dimension of module 34H :

$$w1 = 4 , h1 = 7$$

Dimension of module 4 :

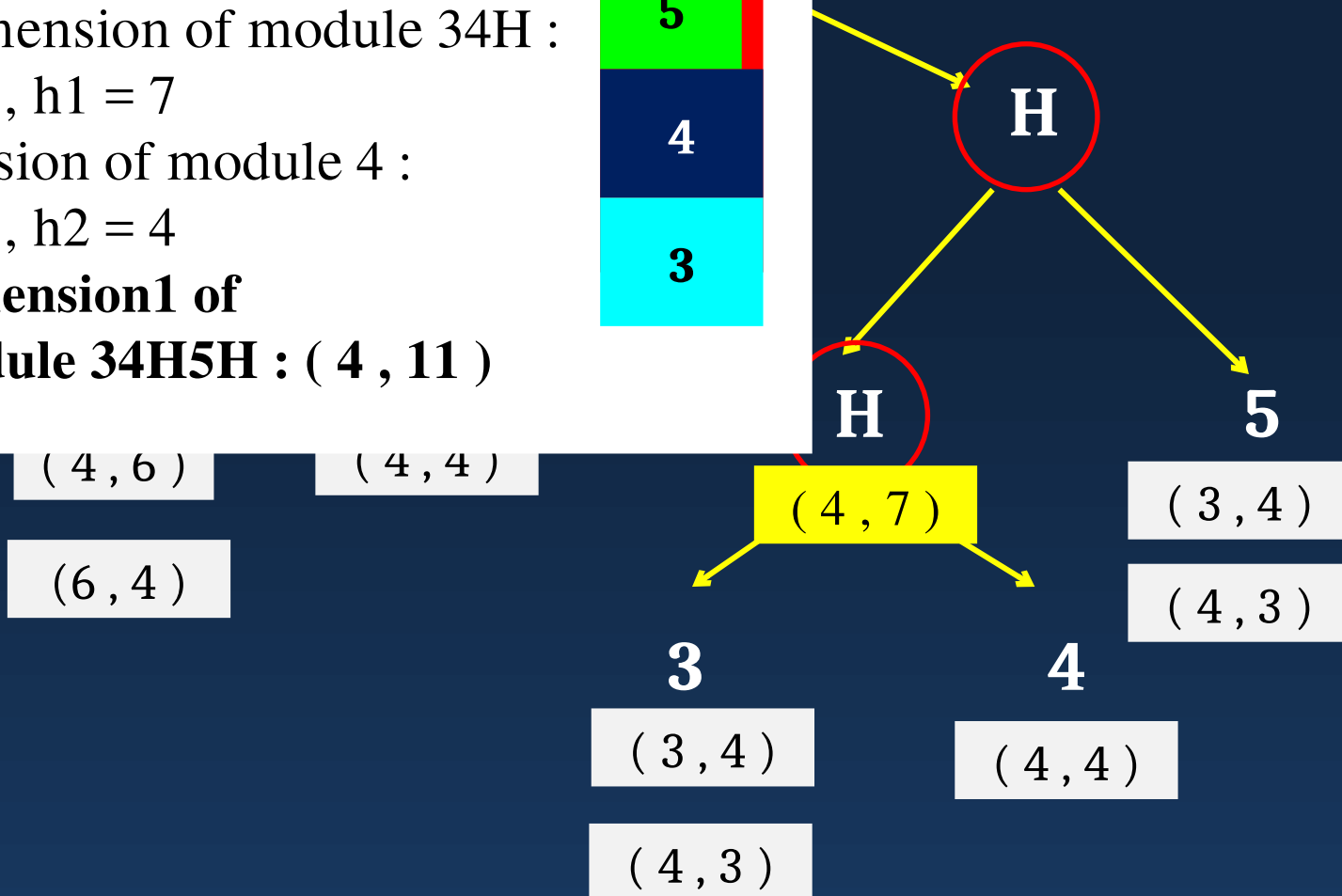
$$w2 = 3 , h2 = 4$$

Dimension1 of

module 34H5H : (4 , 11)



Step 3 : computing dimensions



COMPUTING AREA

$$H = (\max (w1 , w2) , h1+h2)$$

34H5H

(2) Dimension of module 3 :

$$w1 = 4 , h1 = 7$$

Dimension of module 4 :

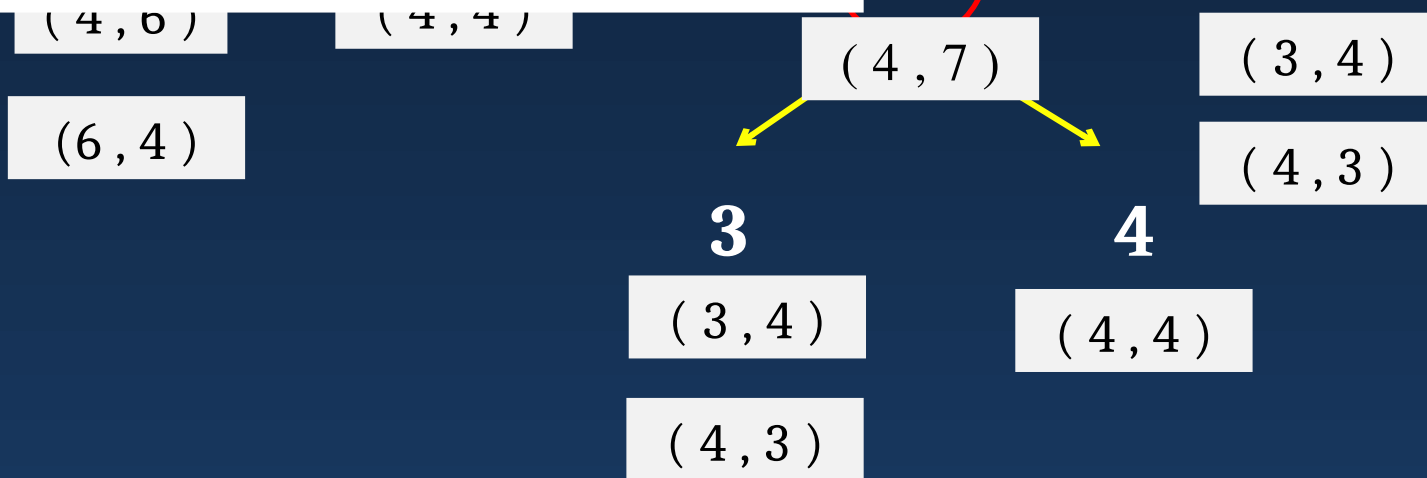
$$w2 = 4 , h2 = 3$$

Dimension2 of

module 34H5H : (4 , 10)

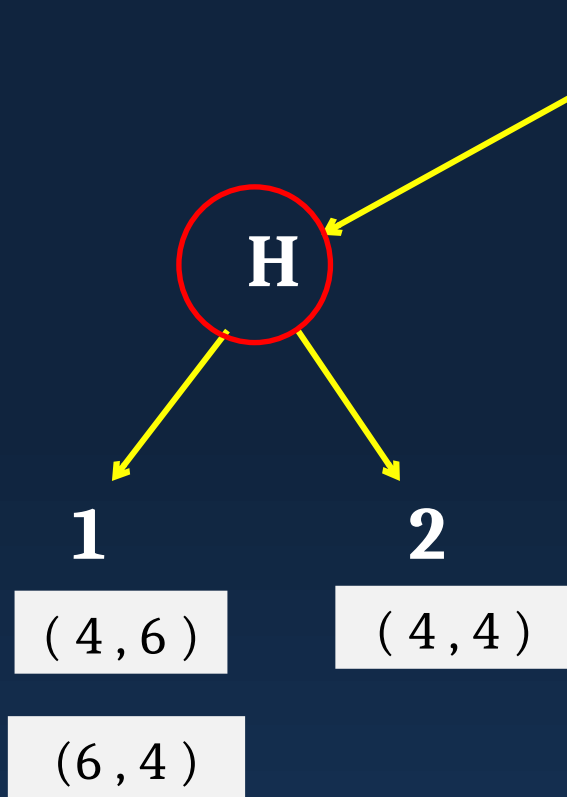


Step 3 : computing dimensions



COMPUTING AREA

Step 3 : computing dimensions



$$H = (\max (w1 , w2) , h1+h2)$$

12H

(1) Dimension of module 1 :

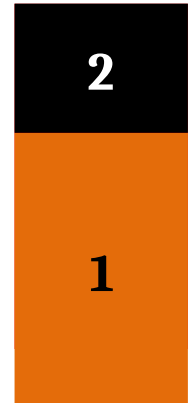
$$w1 = 4 , h1 = 6$$

Dimension of module 2 :

$$w2 = 4 , h2 = 4$$

Dimension1 of

module 12H : (4 , 10)



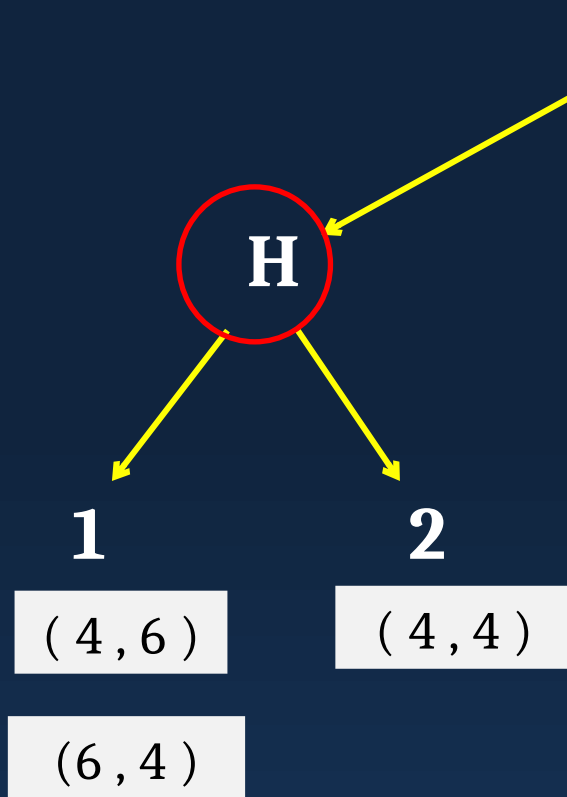
(3 , 4)

(4 , 4)

(4 , 3)

COMPUTING AREA

Step 3 : computing dimensions



$$H = (\max (w1 , w2) , h1+h2)$$

12H

(2) Dimension of module 1 :

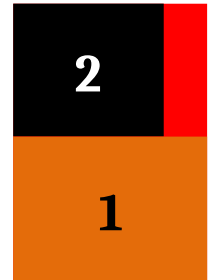
$$w1 = 6 , h1 = 4$$

Dimension of module 2 :

$$w2 = 4 , h2 = 4$$

Dimension1 of

$$\text{module } 12H : (6 , 8)$$



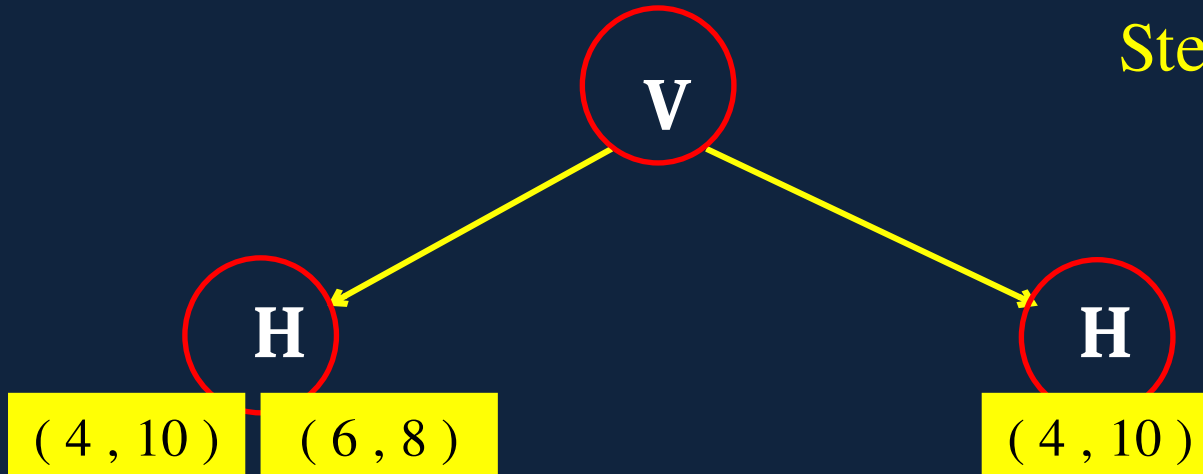
$$(3 , 4)$$

$$(4 , 4)$$

$$(4 , 3)$$

COMPUTING AREA

Step 3 : computing dimensions



1

$$V = (w1+w2 , \max (h1 , h2))$$

12H34H5HV

(4,

(1) Dimension of module 12H :

(6,

w1 = 4 , h1 = 10

Dimension of module 34H5H :

w2 = 4 , h2 = 10

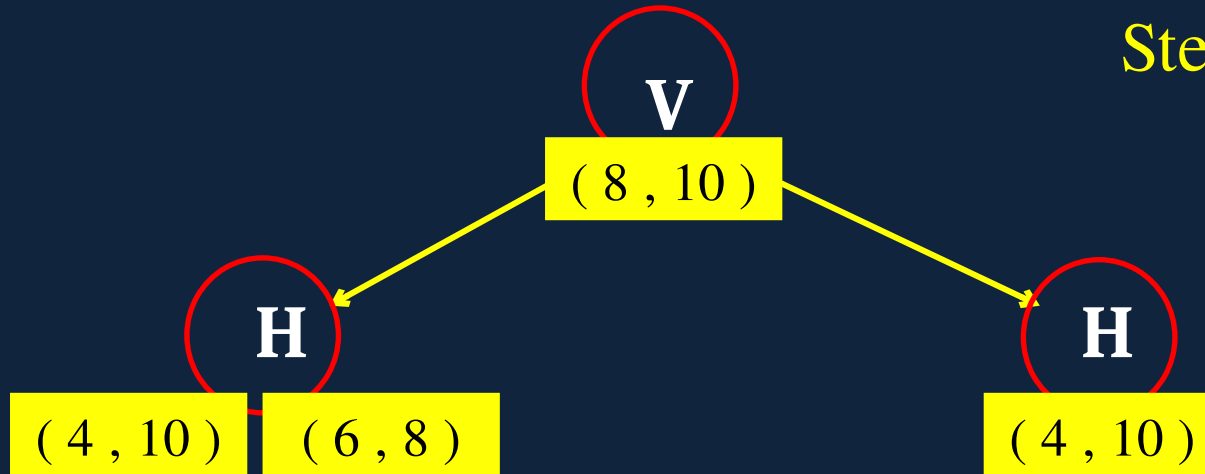
Dimension1 of module

12H : (8 , 10)



COMPUTING AREA

Step 3 : computing dimensions



1

$$V = (w1+w2, \max(h1, h2))$$

12H34H5HV

(4,

(1) Dimension of module 12H :

(6,

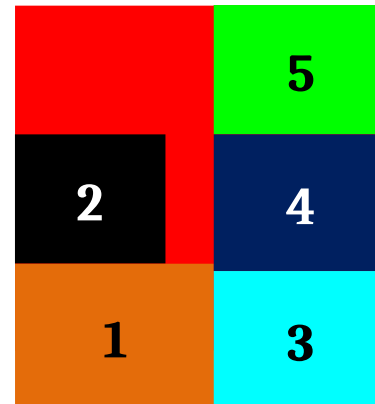
$w1 = 6, h1 = 8$

Dimension of module 34H5H :

$w2 = 4, h2 = 10$

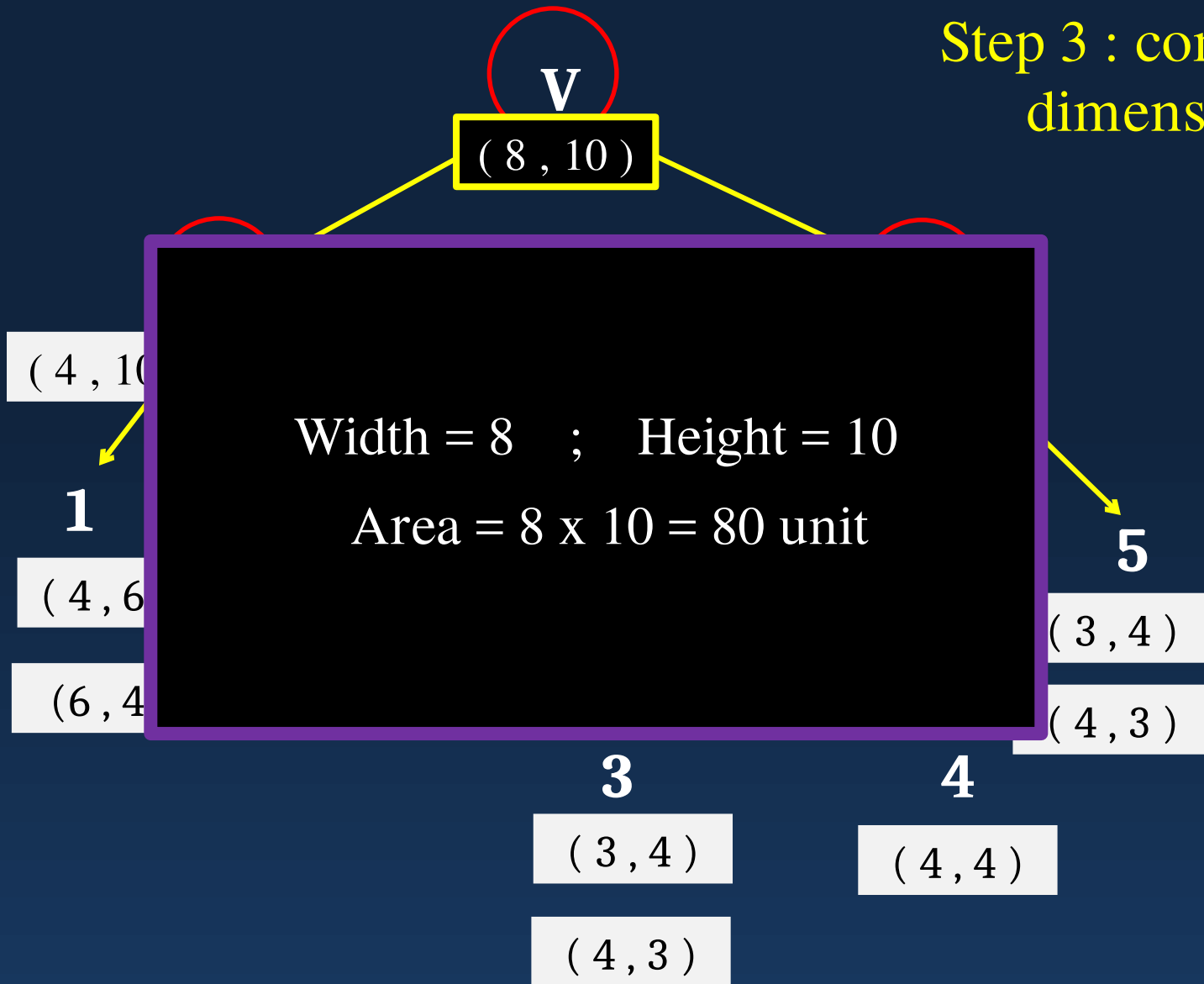
Dimension1 of module

12H : (10 , 10)



COMPUTING AREA

Step 3 : computing dimensions



COMPUTING wirelength

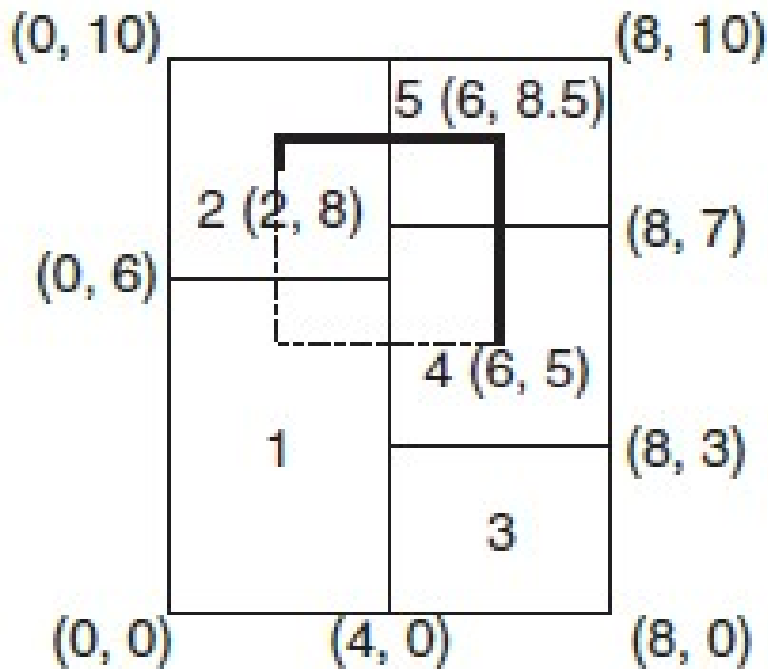
- » Only go through approximation
- » Actual wiring is not performed at the floorplan stage
- » **HPWL** : Popular approximation measurement

Half-perimeter wirelength

The half-perimeter length of the smallest bounding box that encloses all pins or the

Can be computed by use of the centers of modules

COMPUTING wirelength

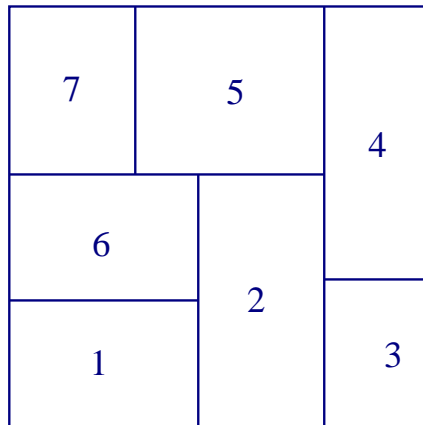


HPWL of a net connecting modules 2 , 4 and 5 :

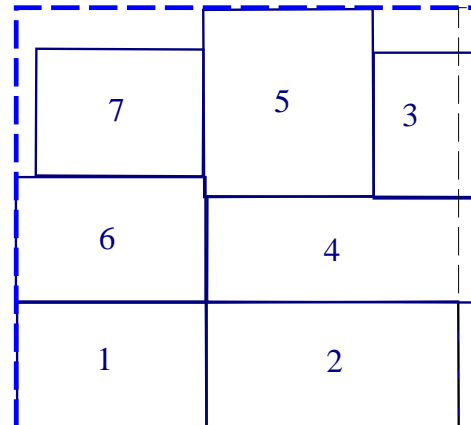
Height = 3.5 ; width = 4
HPWL = 3.5 + 4 = 7.5

Floorplanning

- Inputs to the floorplanning problem:
 - A set of blocks, fixed or flexible.
 - Pin locations of fixed blocks.
 - A netlist.
- Objectives: **Minimize area**, **reduce wirelength** for (critical) nets, **maximize routability**, determine shapes of flexible blocks

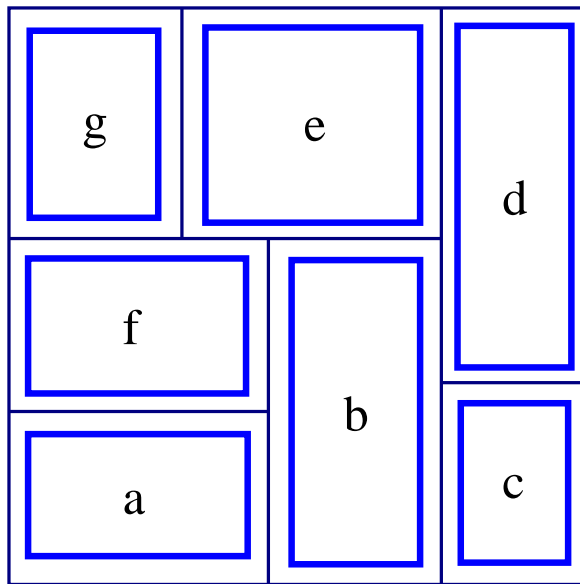



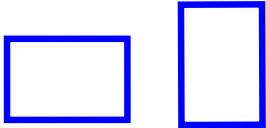
An optimal floorplan,
in terms of area

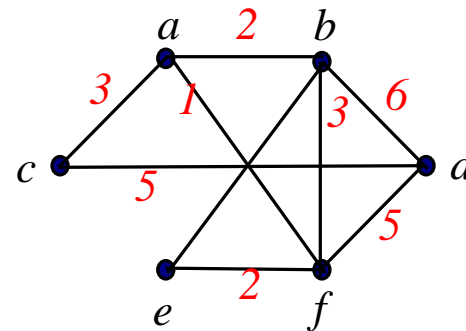


A non-optimal floorplan

Floorplan Design

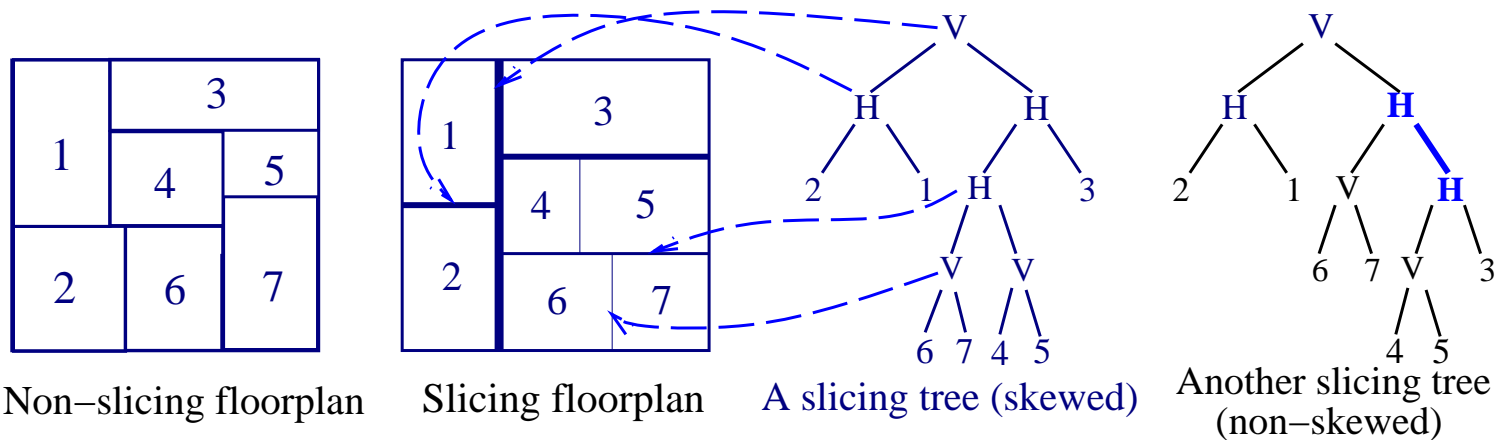


- *Modules:* 
- *Area:* $A=xy$
- *Aspect ratio:* $r \leq y/x \leq s$
- *Rotation:* 
- *Module connectivity*



Floorplanning: Terminology

- **Rectangular dissection:** Subdivision of a given rectangle by a finite # of horizontal and vertical line segments into a finite # of non-overlapping rectangles.
- **Slicing structure:** a rectangular dissection that can be obtained by repetitively subdividing rectangles horizontally or vertically.
- **Slicing tree:** A binary tree, where each internal node represents a vertical cut line or horizontal cut line, and each leaf a basic rectangle.
- **Skewed slicing tree:** One in which no node and its **right** child are the same.




Floorplan Design by Simulated Annealing

- Related work
 - Wong & Liu, “A new algorithm for floorplan design,” DAC’86.
 - * Consider slicing floorplans.
 - Wong & Liu, “Floorplan design for rectangular and L-shaped modules,” ICCAD’87.
 - * Also consider L-shaped modules.
 - Wong, Leong, Liu, *Simulated Annealing for VLSI Design*, pp. 31–71, Kluwer academic Publishers, 1988.
- Ingredients: solution space, neighborhood structure, cost function, annealing schedule?

Solution Representation

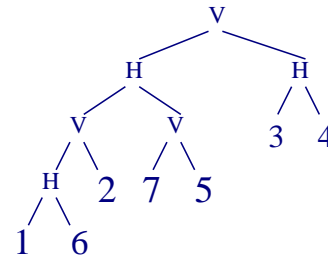
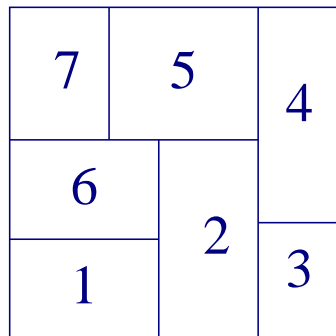
- An expression $E = e_1e_2\dots e_{2n-1}$, where $e_i \in \{1, 2, \dots, n, H, V\}$, $1 \leq i \leq 2n - 1$, is a **Polish expression** of length $2n - 1$ iff
 - every operand j , $1 \leq j \leq n$, appears exactly once in E ;
 - (the balloting property)** for every subexpression $E_i = e_1 \dots e_i$, $1 \leq i \leq 2n - 1$, $\#operands > \#operators$.

1 6 H 3 5 V 2 H V 7 4 H



 $\# \text{ of operands} = 4 \dots\dots = 7$
 $\# \text{ of operators} = 2 \dots\dots = 5$

- Polish expression \longleftrightarrow Postorder traversal.
- ijH : rectangle i on bottom of j ; ijV : rectangle i on the left of j .



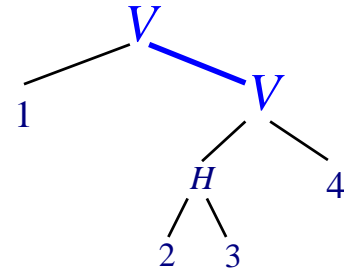
$E = 16H2V75VH34HV$

$E = 16+2*75*+34+*$

Postorder traversal of a tree!

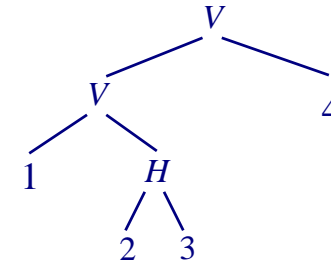
Solution Representation (cont'd)

1	3	4
	2	



$E = 123H4VV$

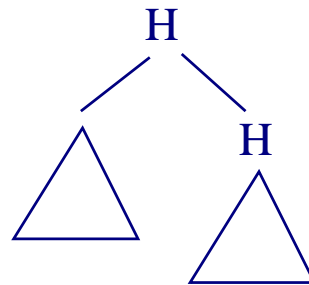
non-skewed!



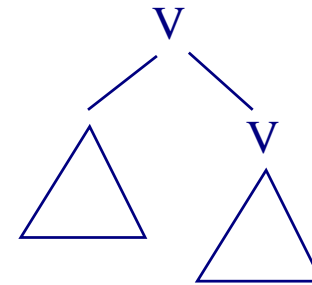
$E = 123HV4V$

skewed!

Non-skewed cases



..... HH

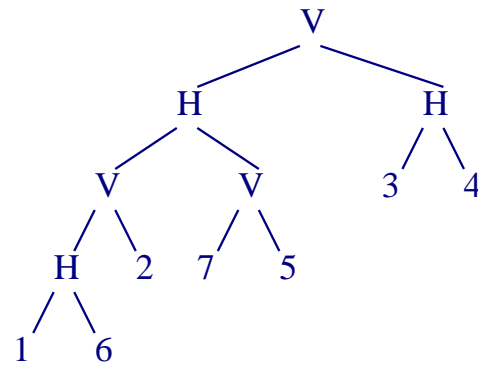
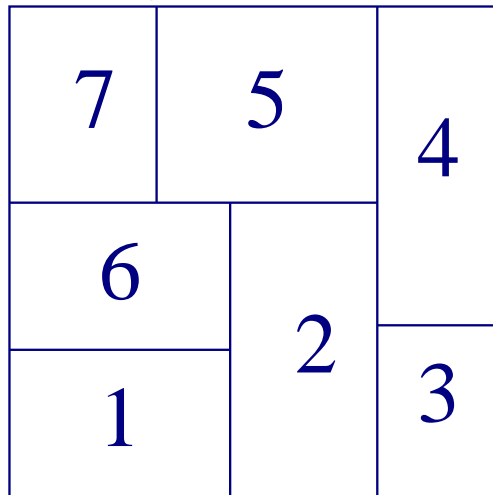


..... VV

- **Question:** How to eliminate ambiguous representation?

Normalized Polish Expression

- A Polish expression $E = e_1e_2\dots e_{2n-1}$ is called **normalized** iff E has no consecutive operators of the same type (H or V).
- Given a **normalized** Polish expression, we can construct a **unique** rectangular slicing structure.

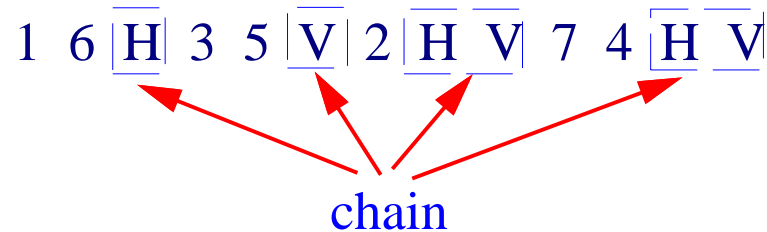


$E = 16H2V75VH34HV$

A normalized Polish expression

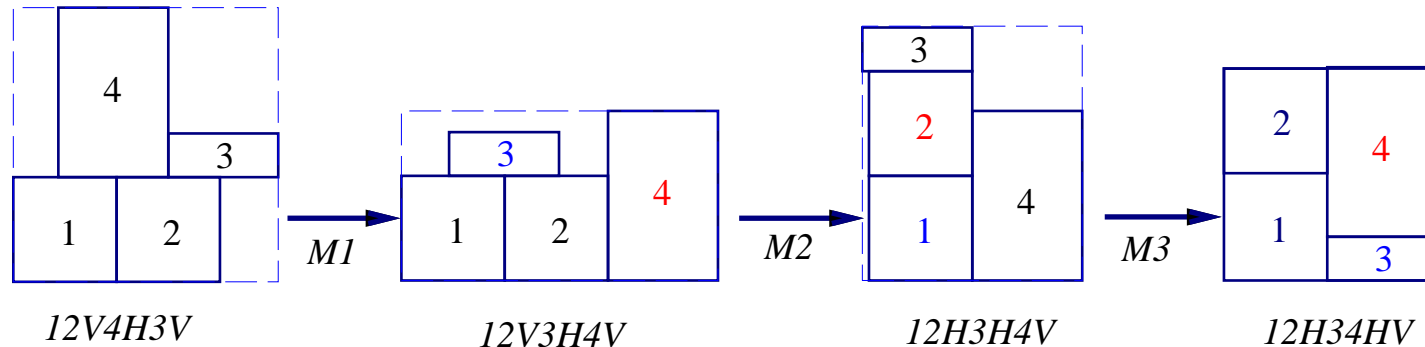
Neighborhood Structure

- **Chain:** $HVHVH\dots$ or $VHVHV\dots$



- **Adjacent:** 1 and 6 are adjacent operands; 2 and 7 are adjacent operands; 5 and V are adjacent operand and operator.
- 3 types of moves:
 - $M1$ (**Operand Swap**): Swap two adjacent operands.
 - $M2$ (**Chain Invert**): Complement some chain ($\bar{V} = H, \bar{H} = V$).
 - $M3$ (**Operator/Operand Swap**): Swap two adjacent operand and operator.

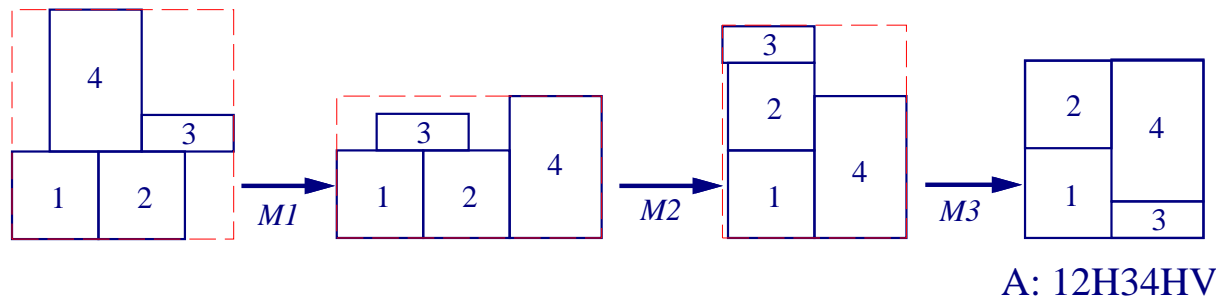
Effects of Perturbation



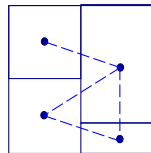
- **Question:** The balloting property holds during the moves?
 - $M1$ and $M2$ moves are OK.
 - **Check the $M3$ moves!** Reject “illegal” $M3$ moves.
- **Check $M3$ moves:** Assume that the M_3 move swaps the operand e_i with the operator e_{i+1} , $1 \leq i \leq k - 1$. Then, the swap will not violate the balloting property iff $2N_{i+1} < i$.
 - N_k : # of operators in the Polish expression $E = e_1e_2 \dots e_k, 1 \leq k \leq 2n - 1$.

Cost Function

- $\Phi = A + \lambda W$.
 - A : area of the smallest rectangle
 - W : overall wiring length
 - λ : user-specified parameter

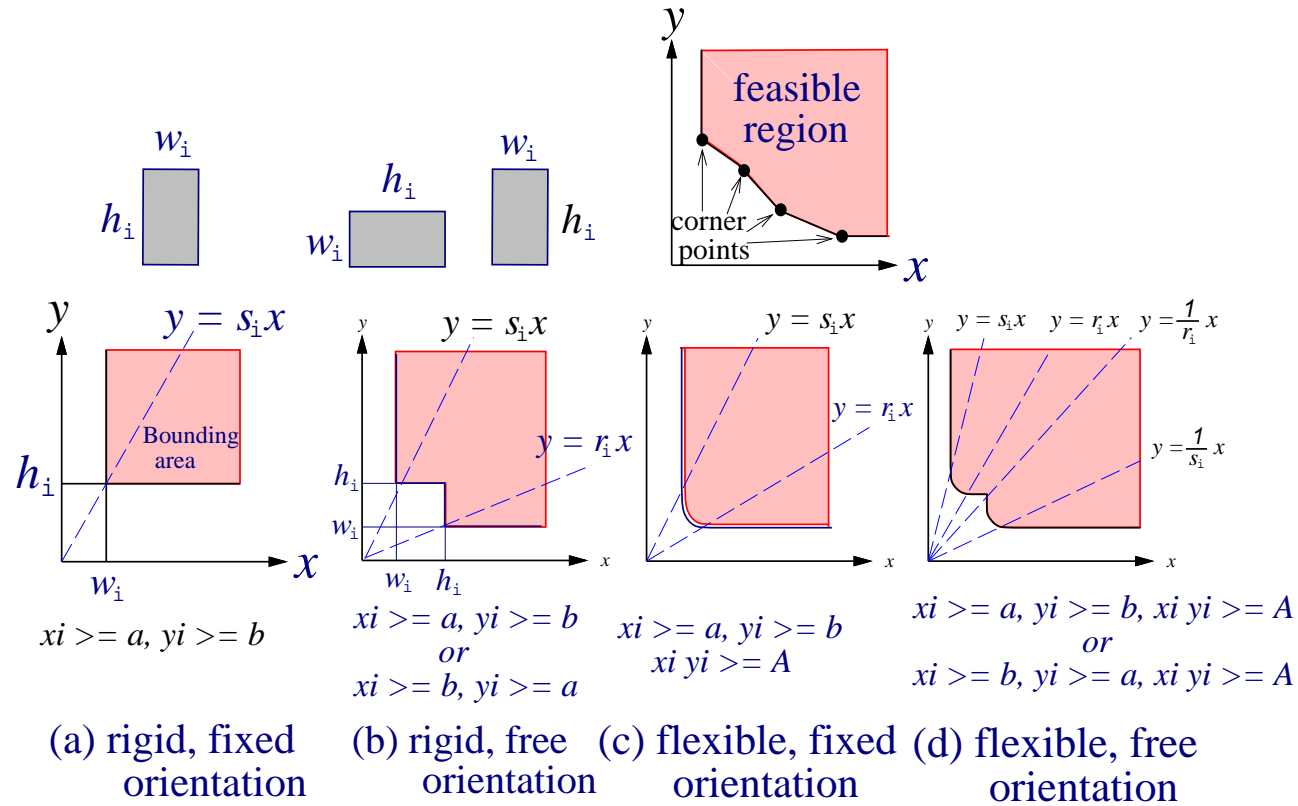


- $W = \sum_{ij} c_{ij} d_{ij}$.
 - c_{ij} : # of connections between blocks i and j .
 - d_{ij} : center-to-center distance between basic rectangles i and j .

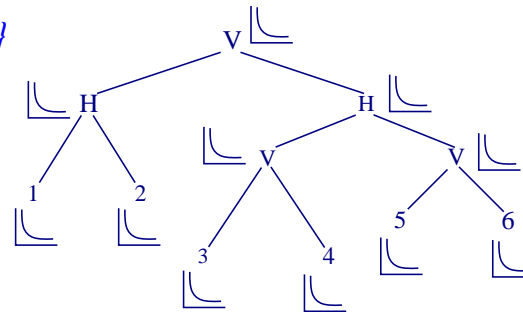
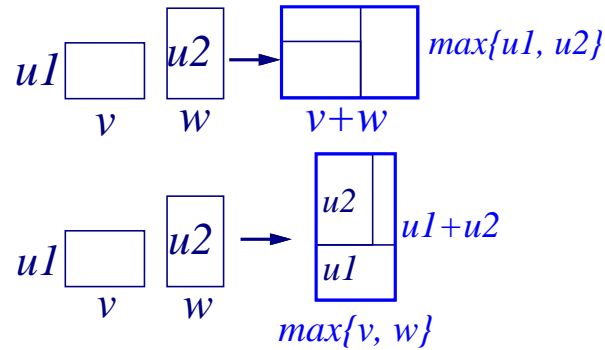
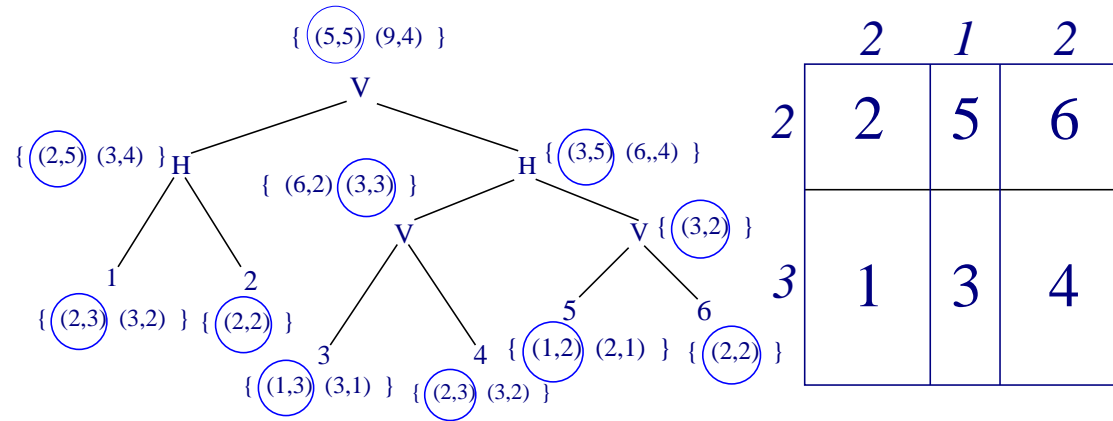


Cost Evaluation: Shape Curves

- Shape curves correspond to different kinds of constraints where the shaded areas are feasible regions.



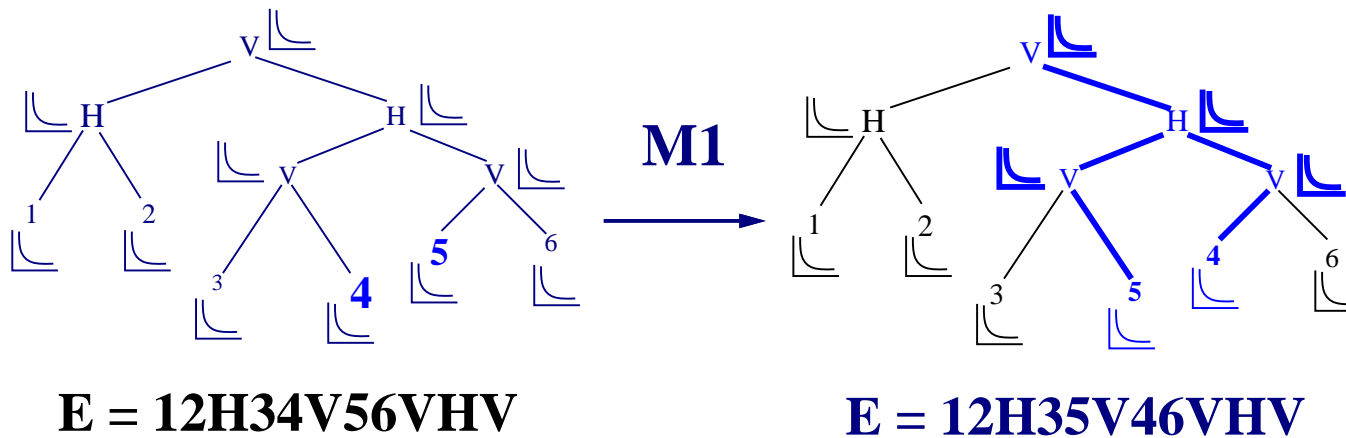
Area Computation



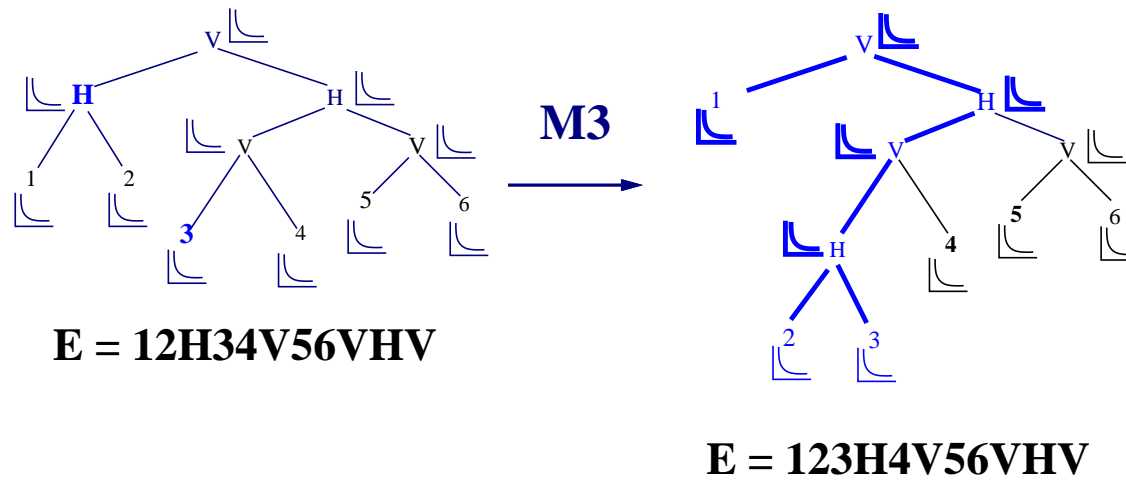
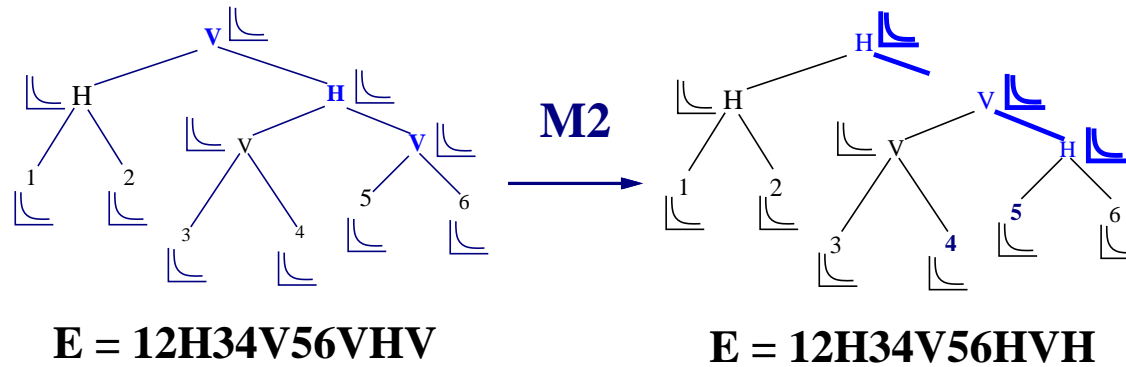
- Wiring cost?

Incremental Computation of Cost Function

- Each move leads to only a minor modification of the Polish expression.
- At most **two paths** of the slicing tree need to be updated for each move.



Incremental Computation of Cost Function (cont'd)



Annealing Schedule

- Initial solution: $12V3V \dots nV$.

1	2	3		n
----------	----------	----------	--	----------

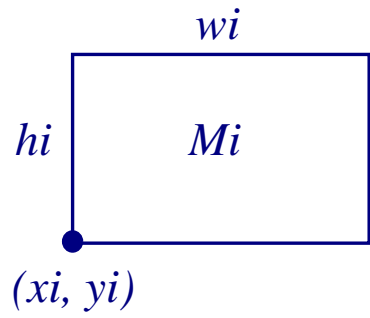
- $T_i = r^i T_0, i = 1, 2, 3, \dots; r = 0.85$.
- At each temperature, try kn moves ($k = 5-10$).
- Terminate the annealing process if
 - # of accepted moves $< 5\%$,
 - temperature is low enough, or
 - run out of time.

Algorithm: Simulated_Annealing_Floorplanning(P, ϵ, r, k)

```
1 begin
2  $E \leftarrow 12V3V4V \dots nV$ ; /* initial solution */
3  $Best \leftarrow E$ ;  $T_0 \leftarrow \frac{\Delta_{avg}}{\ln(P)}$ ;  $M \leftarrow MT \leftarrow uphill \leftarrow 0$ ;  $N = kn$ ;
4 repeat
5    $MT \leftarrow uphill \leftarrow reject \leftarrow 0$ ;
6   repeat
7      $SelectMove(M)$ ;
8     Case  $M$  of
9        $M_1$ :  $Select\ two\ adjacent\ operands\ e_i\ and\ e_j$ ;  $NE \leftarrow Swap(E, e_i, e_j)$ ;
10       $M_2$ :  $Select\ a\ nonzero\ length\ chain\ C$ ;  $NE \leftarrow Complement(E, C)$ ;
11       $M_3$ :  $done \leftarrow FALSE$ ;
12      while not ( $done$ ) do
13         $Select\ two\ adjacent\ operand\ e_i\ and\ operator\ e_{i+1}$ ;
14        if ( $e_{i-1} \neq e_{i+1}$ ) and ( $2N_{i+1} < i$ ) then  $done \leftarrow TRUE$ ;
15         $NE \leftarrow Swap(E, e_i, e_{i+1})$ ;
16         $MT \leftarrow MT + 1$ ;  $\Delta cost \leftarrow cost(NE) - cost(E)$ ;
17        if ( $\Delta cost \leq 0$ ) or ( $Random < e^{\frac{-\Delta cost}{T}}$ )
18        then
19          if ( $\Delta cost > 0$ ) then  $uphill \leftarrow uphill + 1$ ;
20           $E \leftarrow NE$ ;
21          if  $cost(E) < cost(best)$  then  $best \leftarrow E$ ;
22        else  $reject \leftarrow reject + 1$ ;
23 until ( $uphill > N$ ) or ( $MT > 2N$ );
24  $T = rT$ ; /* reduce temperature */
25 until ( $\frac{reject}{MT} > 0.95$ ) or ( $T < \epsilon$ ) or  $OutOfTime$ ;
26 end
```

Floorplanning by Mathematical Programming

- Sutanthavibul, Shragowitz, and Rosen, “An analytical approach to floorplan design and optimization,” 27th DAC, 1990.
- Notation:
 - w_i, h_i : width and height of module M_i .
 - (x_i, y_i) : coordinate of the lower left corner of module M_i .
 - $a_i \leq w_i/h_i \leq b_i$: aspect ratio w_i/h_i of module M_i . (Note: We defined aspect ratio as h_i/w_i before.)
- Goal: Find a mixed **integer linear programming (ILP)** formulation for the floorplan design.
 - **Linear** constraints? Objective function?



$$\text{Area} = h_i * w_i$$
$$\text{Aspect ratio} = w_i / h_i$$

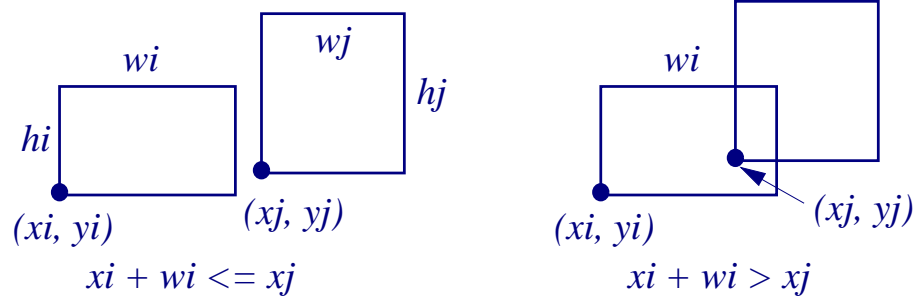
Nonoverlap Constraints

- Two modules M_i and M_j are nonoverlap, if at least one of the following linear constraints is satisfied (cases encoded by p_{ij} and q_{ij}):

M_i to the left of M_j :	$x_i + w_i \leq x_j$	p_{ij}	q_{ij}
M_i below M_j :	$y_i + h_i \leq y_j$	0	1
M_i to the right of M_j :	$x_i - w_j \geq x_j$	1	0
M_i above M_j :	$y_i - h_j \geq y_j$	1	1

- Let W, H be upper bounds on the floorplan width and height, respectively.
- Introduce two 0,1 variables p_{ij} and q_{ij} to denote that one of the above inequalities is enforced; e.g., $p_{ij} = 0, q_{ij} = 1 \Rightarrow y_i + h_i \leq y_j$ is satisfied.

$$\begin{aligned}
 x_i + w_i &\leq x_j + W(p_{ij} + q_{ij}) \\
 y_i + h_i &\leq y_j + H(1 + p_{ij} - q_{ij}) \\
 x_i - w_j &\geq x_j - W(1 - p_{ij} + q_{ij}) \\
 y_i - h_j &\geq y_j - H(2 - p_{ij} - q_{ij})
 \end{aligned}$$



Cost Function & Constraints

- Minimize $Area = xy$, **nonlinear!** (x, y : width and height of the resulting floorplan)
- How to fix?
 - Fix the width W and minimize the height y !
- Four types of constraints:
 1. no two modules overlap ($\forall i, j : 1 \leq i < j \leq n$);
 2. each module is enclosed within a rectangle of width W and height H ($x_i + w_i \leq W, y_i + h_i \leq H, 1 \leq i \leq n$);
 3. $x_i \geq 0, y_i \geq 0, 1 \leq i \leq n$;
 4. $p_{ij}, q_{ij} \in \{0, 1\}$.
- w_i, h_i are known.

Mixed ILP for Floorplanning

Mixed ILP for the floorplanning problem with rigid, fixed modules.

$$\begin{array}{ll}
 \min & y \\
 \text{subject to} & \\
 & x_i + w_i \leq W, \quad 1 \leq i \leq n \quad (1) \\
 & y_i + h_i \leq y, \quad 1 \leq i \leq n \quad (2) \\
 & x_i + w_i \leq x_j + W(p_{ij} + q_{ij}), \quad 1 \leq i < j \leq n \quad (3) \\
 & y_i + h_i \leq y_j + H(1 + p_{ij} - q_{ij}), \quad 1 \leq i < j \leq n \quad (4) \\
 & x_i - w_j \geq x_j - W(1 - p_{ij} + q_{ij}), \quad 1 \leq i < j \leq n \quad (5) \\
 & y_i - h_j \geq y_j - H(2 - p_{ij} - q_{ij}), \quad 1 \leq i < j \leq n \quad (6) \\
 & x_i, y_i \geq 0, \quad 1 \leq i \leq n \quad (7) \\
 & p_{ij}, q_{ij} \in \{0, 1\}, \quad 1 \leq i < j \leq n \quad (8)
 \end{array}$$

- Size of the mixed ILP: for n modules,
 - # continuous variables: $O(n)$; # integer variables: $O(n^2)$; # linear constraints: $O(n^2)$.
 - Unacceptably huge program for a large n ! (How to cope with it?)
- Popular LP software: LINDO, Ip_solve, etc.

Mixed ILP for Floorplanning (cont'd)

Mixed ILP for the floorplanning problem: rigid, freely oriented modules.

min y

subject to

$$x_i + r_i h_i + (1 - r_i) w_i \leq W, \quad 1 \leq i \leq n \quad (9)$$

$$y_i + r_i w_i + (1 - r_i) h_i \leq y, \quad 1 \leq i \leq n \quad (10)$$

$$x_i + r_i h_i + (1 - r_i) w_i \leq x_j + M(p_{ij} + q_{ij}), \quad 1 \leq i < j \leq n \quad (11)$$

$$y_i + r_i w_i - (1 - r_i) h_i \leq y_j + M(1 + p_{ij} - q_{ij}), \quad 1 \leq i < j \leq n \quad (12)$$

$$x_i - r_j h_j + (1 - r_j) w_j \geq x_j - M(1 - p_{ij} + q_{ij}), \quad 1 \leq i < j \leq n \quad (13)$$

$$y_i - r_j w_j - (1 - r_j) h_j \geq y_j - M(2 - p_{ij} - q_{ij}), \quad 1 \leq i < j \leq n \quad (14)$$

$$x_i, y_i \geq 0, \quad 1 \leq i \leq n \quad (15)$$

$$p_{ij}, q_{ij} \in \{0, 1\}, \quad 1 \leq i < j \leq n \quad (16)$$

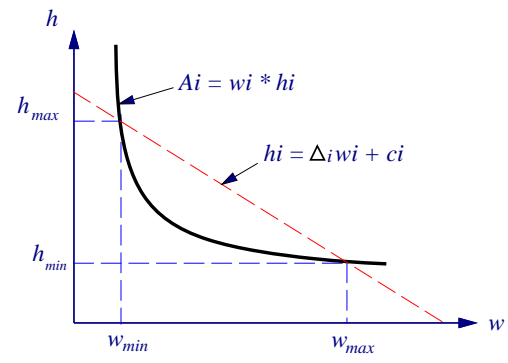
- For each module i with free orientation, associate a 0-1 variable r_i :
 - $r_i = 0$: 0° rotation for module i .
 - $r_i = 1$: 90° rotation for module i .
- $M = \max\{W, H\}$.

Flexible Modules

- Assumptions: w_i, h_i are unknown; area lower bound: A_i .
- Module size constraints: $w_i h_i \geq A_i$; $a_i \leq \frac{w_i}{h_i} \leq b_i$.
- Hence, $w_{min} = \sqrt{A_i a_i}$, $w_{max} = \sqrt{A_i b_i}$, $h_{min} = \sqrt{\frac{A_i}{b_i}}$, $h_{max} = \sqrt{\frac{A_i}{a_i}}$.
- $w_i h_i \geq A_i$ nonlinear! How to fix?
 - Can apply a first-order approximation of the equation: a line passing through (w_{min}, h_{max}) and (w_{max}, h_{min}) .

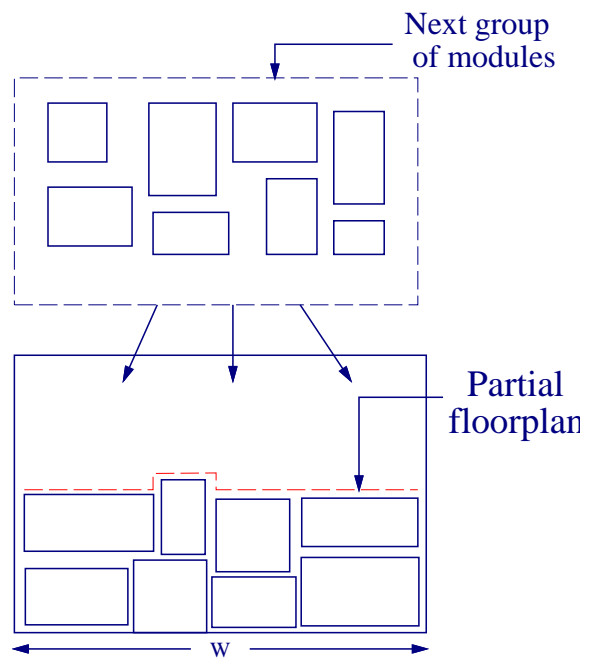
$$\begin{aligned} h_i &= \Delta_i w_i + c_i & / * y = mx + c * / \\ \Delta_i &= \frac{h_{max} - h_{min}}{w_{min} - w_{max}} & / * slope * / \\ c_i &= h_{max} - \Delta_i w_{min} & / * c = y_0 - mx_0 * / \end{aligned}$$

- Substitute $\Delta_i w_i + c_i$ for h_i to form linear constraints (x_i, y_i, w_i are unknown; $\Delta_i, \Delta_j, c_i, c_j$ can be computed as above).



Reducing the Size of the Mixed ILP

- Time complexity of a mixed ILP: exponential!
- Recall the large size of the mixed ILP: # variables, # constraints: $O(n^2)$.
 - How to fix it?
- Key: Solve a partial problem at each step (successive augmentation)
- Questions:
 - How to select next subgroup of modules? \Rightarrow linear ordering based on connectivity.
 - How to minimize the # of required variables?



Reducing the Size of the Mixed ILP (cont'd)

- Size of each successive mixed ILP depends on (1) # of modules in the next group; (2) “size” of the partially constructed floorplan.
- Keys to deal with (2)
 - Minimize the problem size of the partial floorplan.
 - Replace the already placed modules by a set of covering rectangles.
 - # rectangles is usually much smaller than # placed modules.

