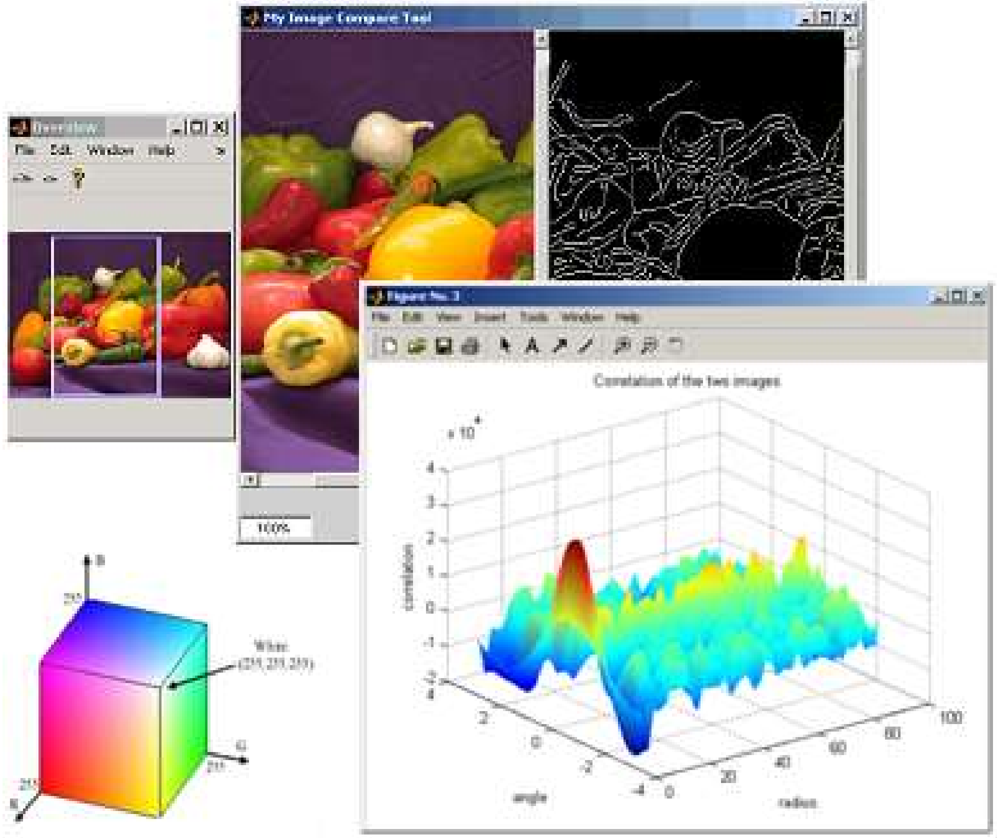
Digital Image Processing



Dr. Ajay Kumar Mahato
Assistant Professor
ECE Department
Gla University Mathura

DIGITAL IMAGE PROCESSING

LECTURE -13

Morphological Operations (Recap)

Morphological Operations

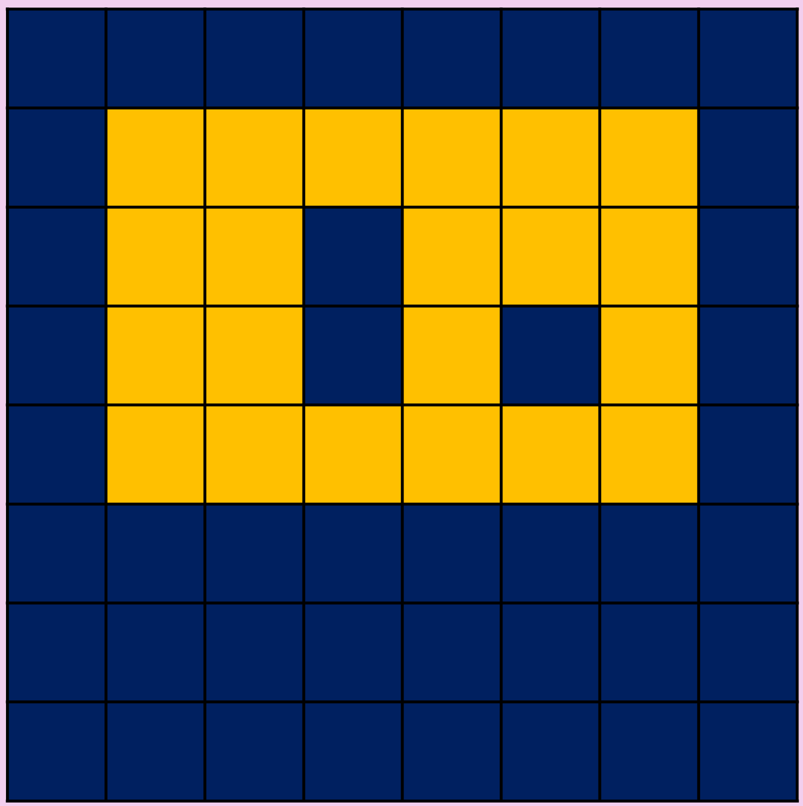
- Image Morphology is the study of shapes of the objects present in the image and extraction of image features. Image features are necessary for object recognition.

- We user the same word here in the context of mathematical morphology as a tool for ***extracting image components*** that are useful in the ***representation and description of region shape***, such as ***boundaries, skeletons*** and ***convex hull.***

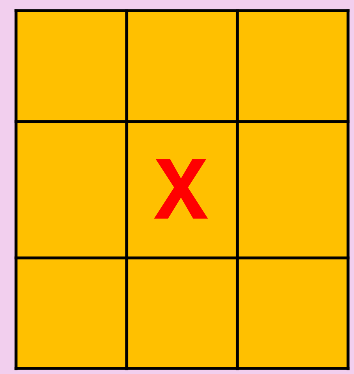
Dilation Operations

Perform $X \oplus B$

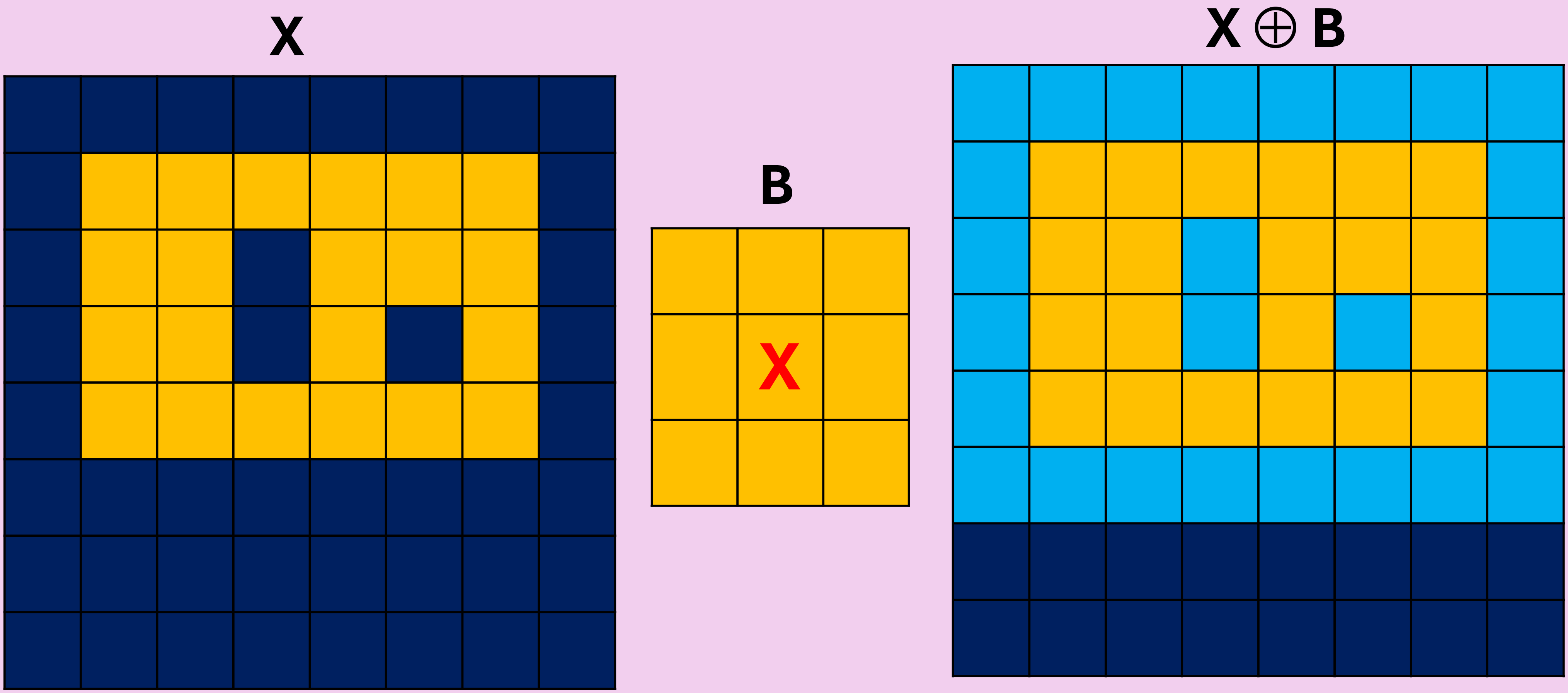
X



B



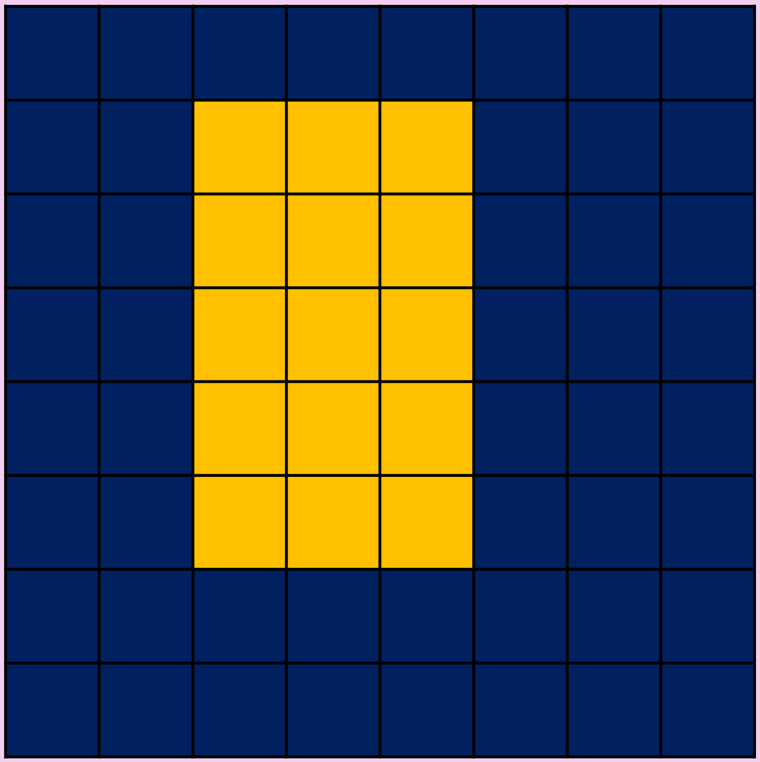
Dilation Operations



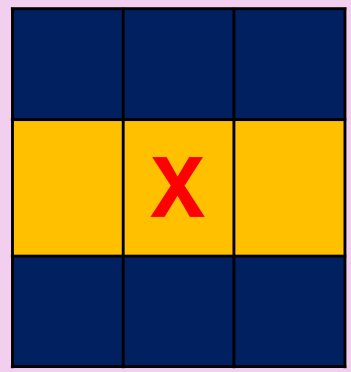
Erosion Operation

Perform X ⊖ B

X

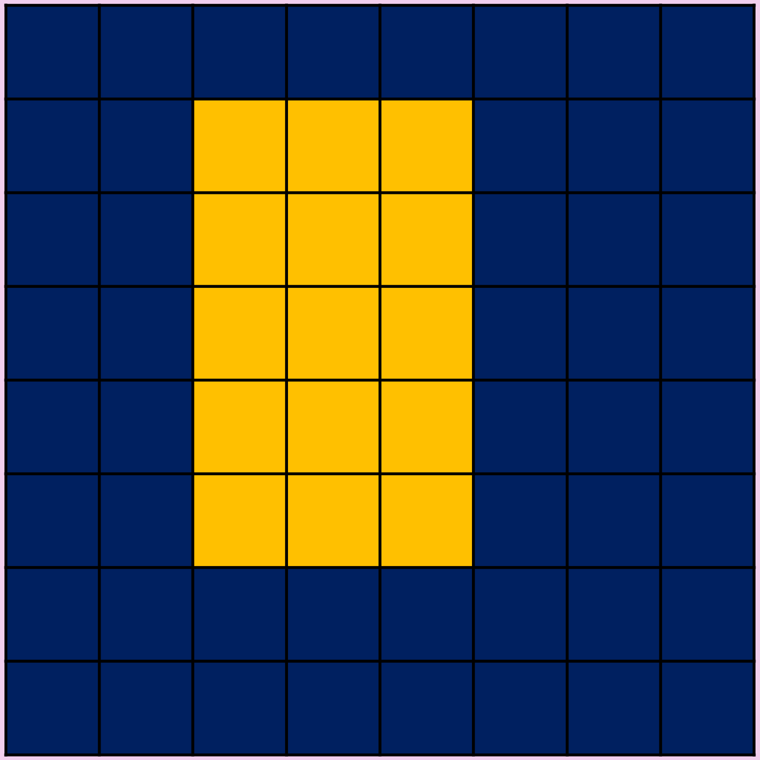


B

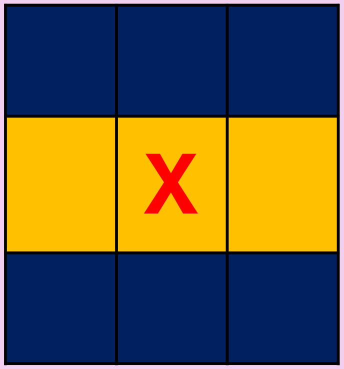


Erosion Operation

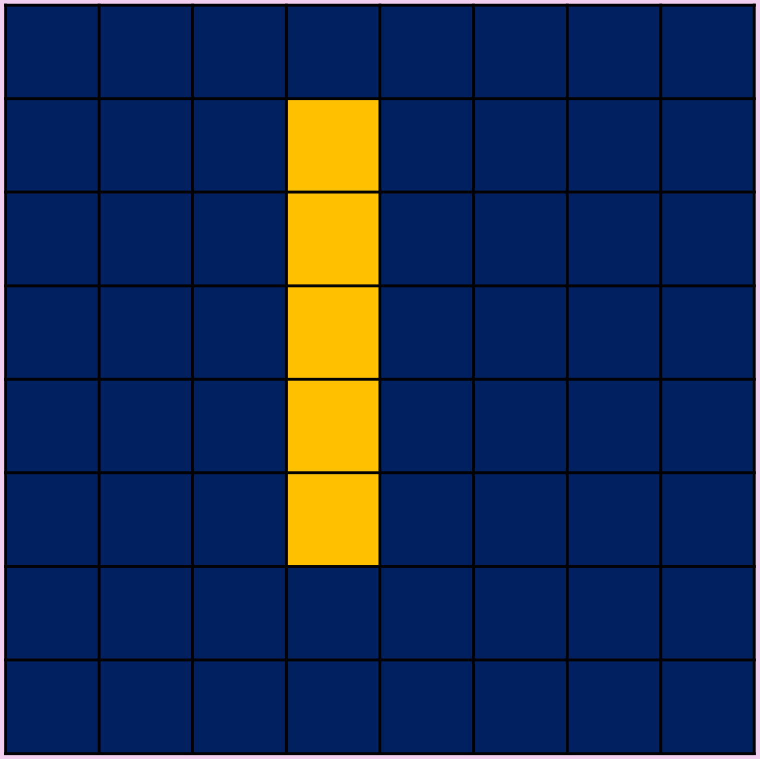
X



B



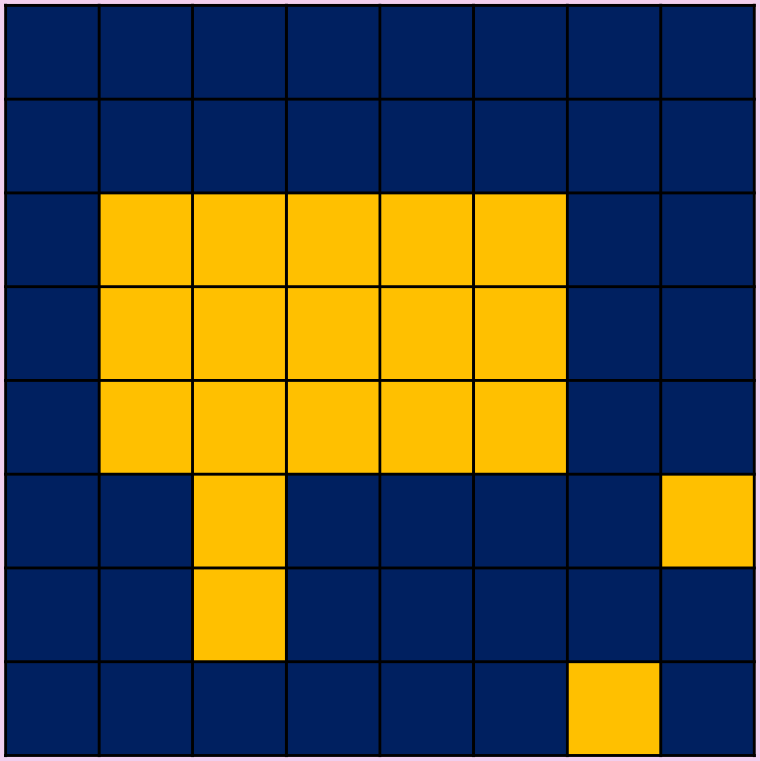
$X \ominus B$



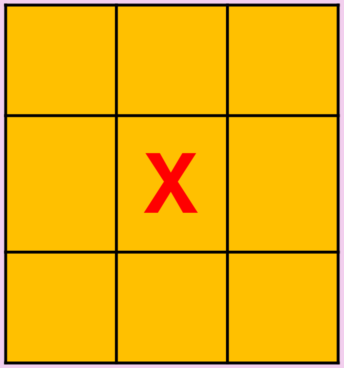
Erosion Operation

Perform X ⊖ B

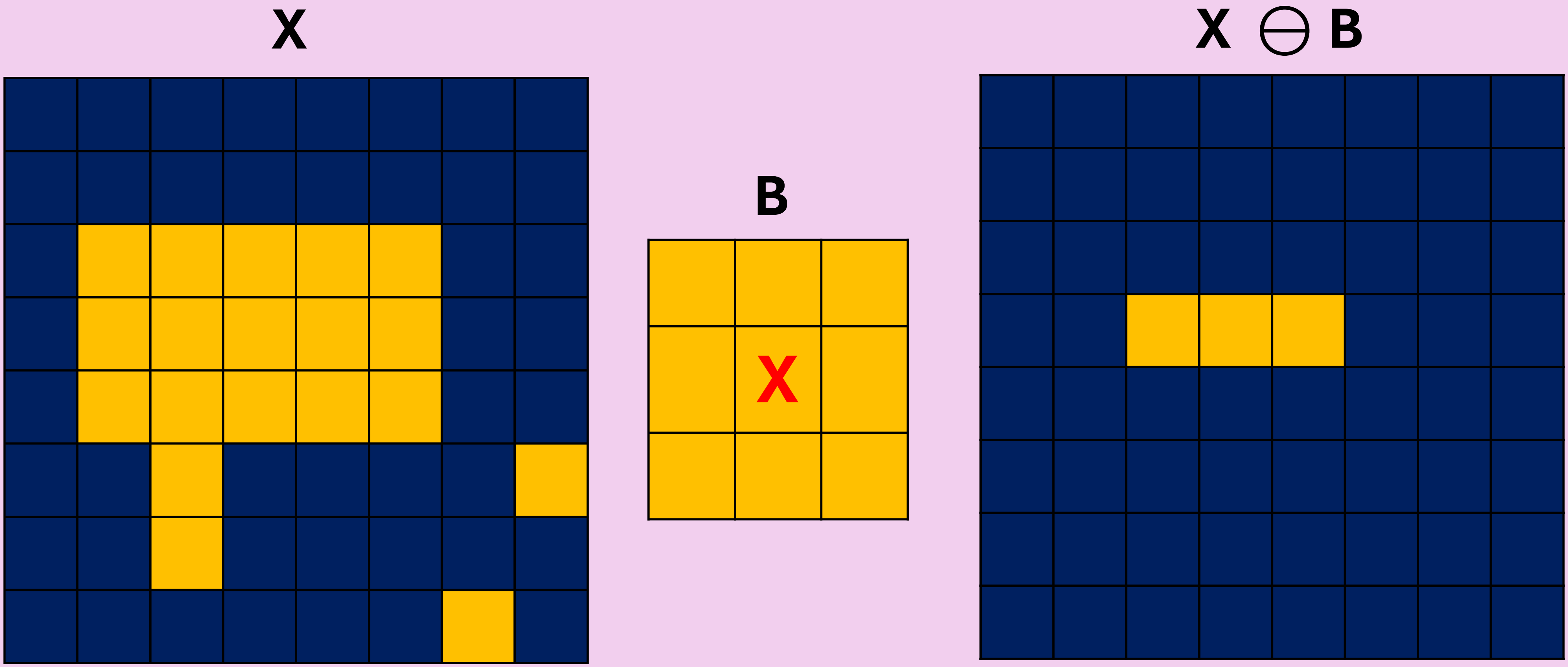
X



B



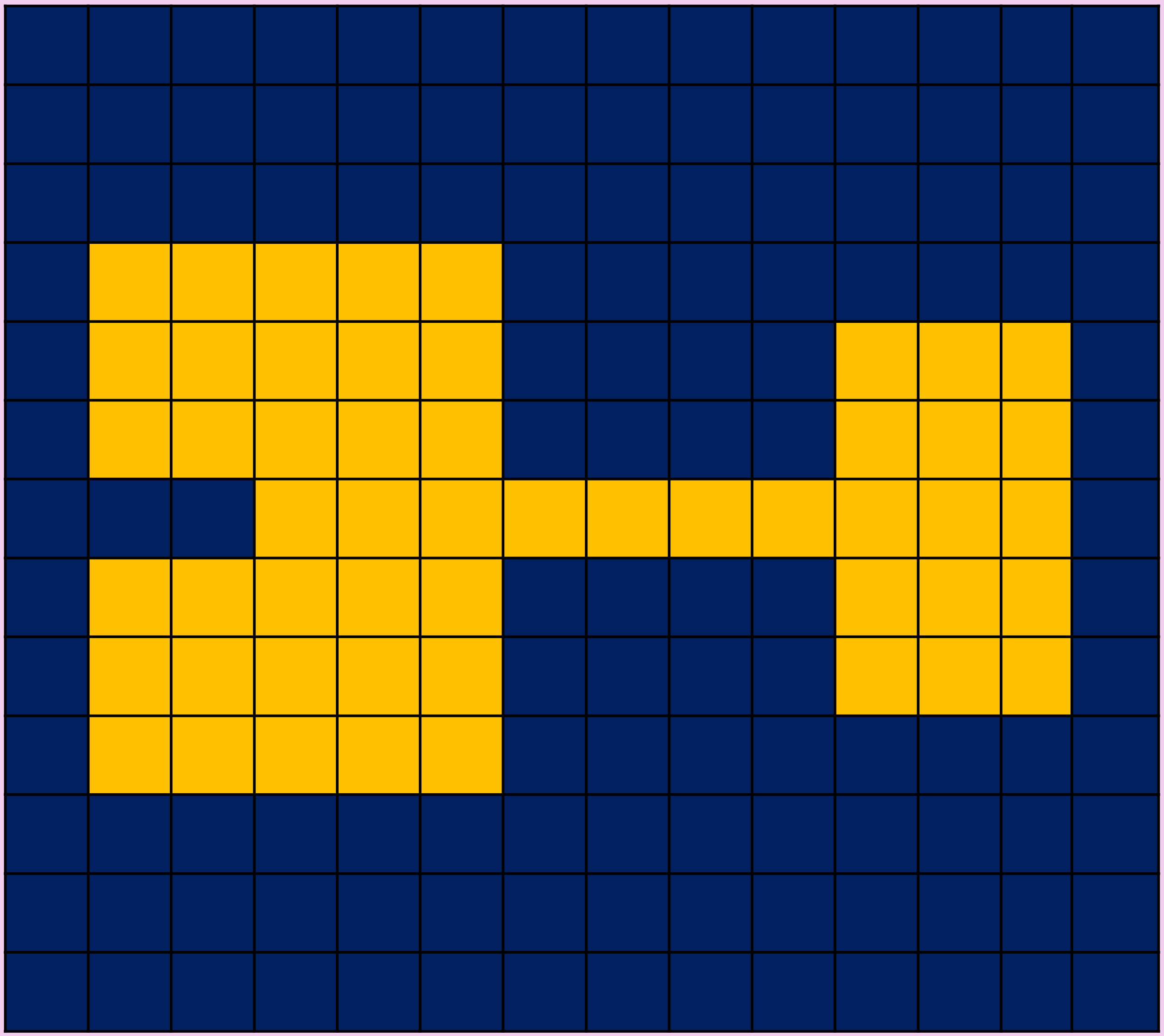
Erosion Operation



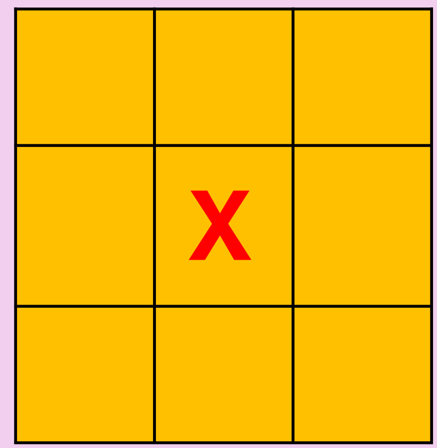
Opening Operation

- Opening Operation: $A \circ B = (A \ominus B) \oplus B$

A



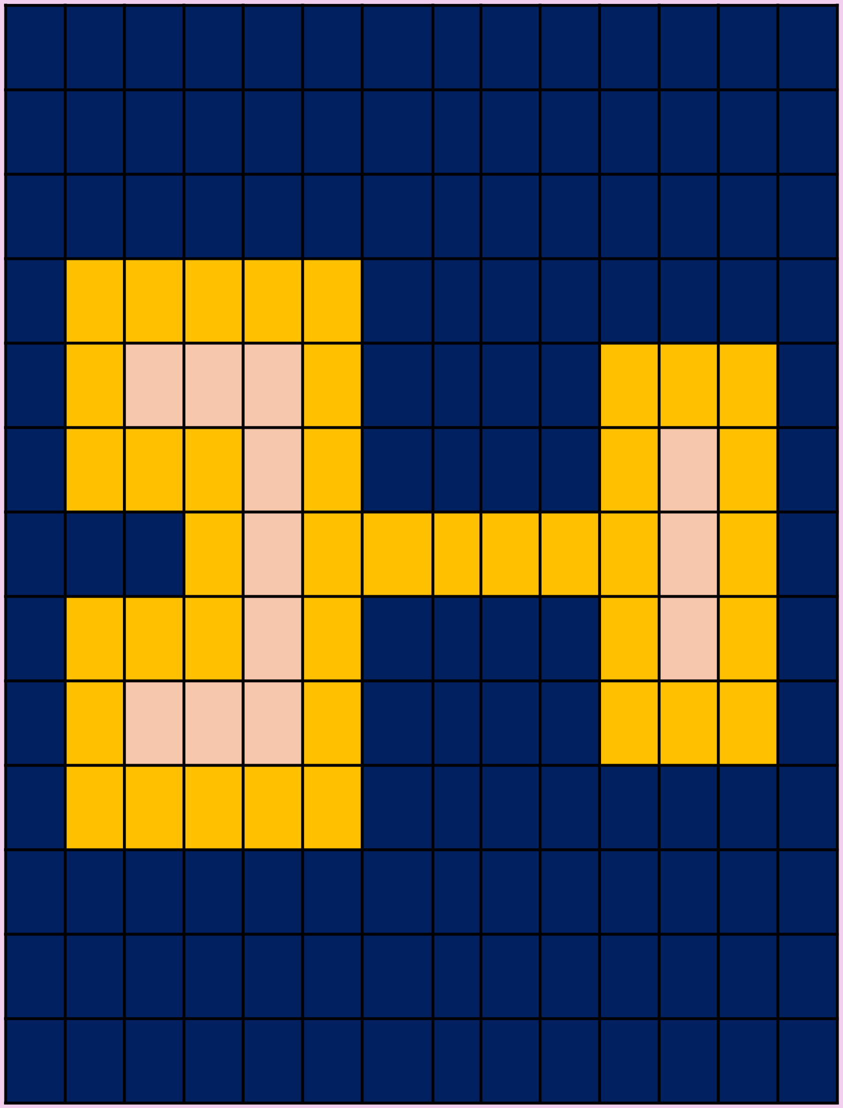
B



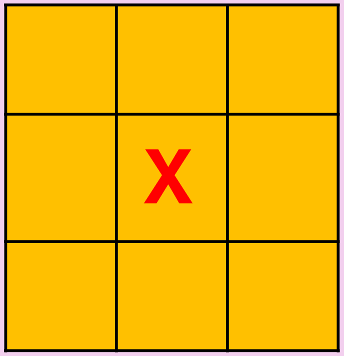
Opening Operation

$$A \circ B = (A \ominus B) \oplus B$$

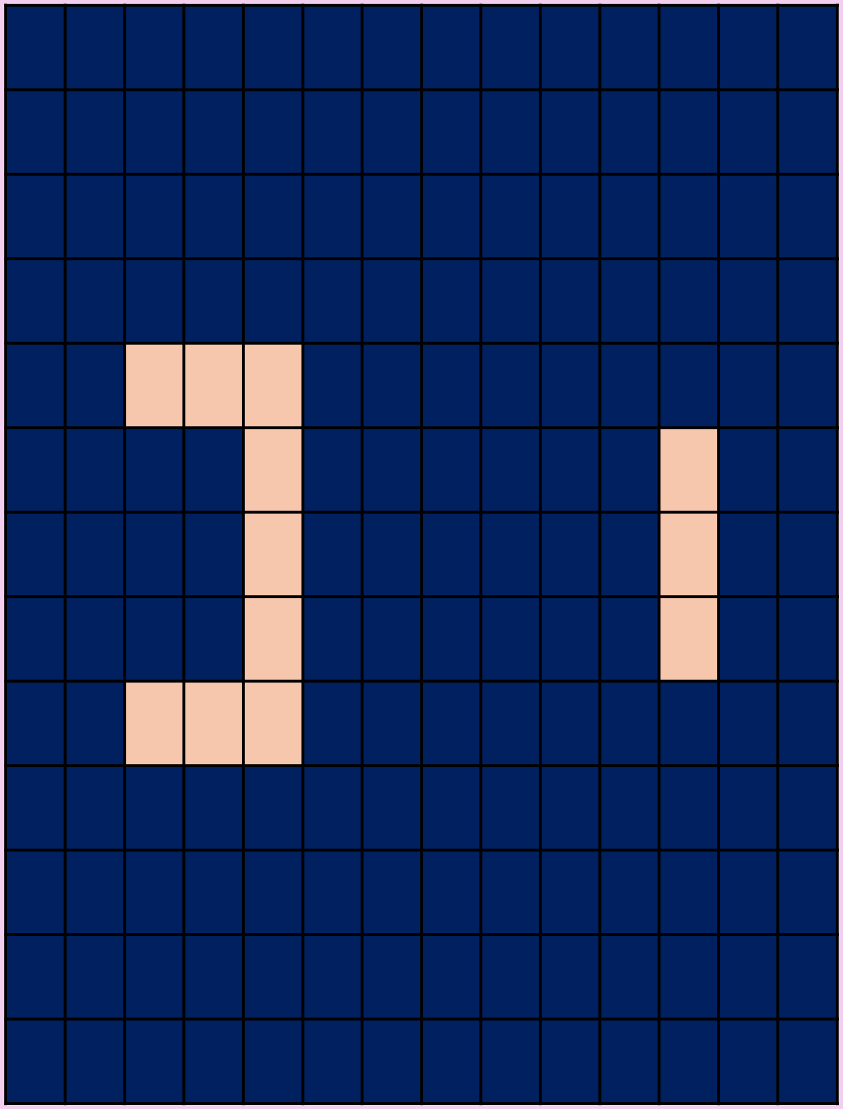
A



B



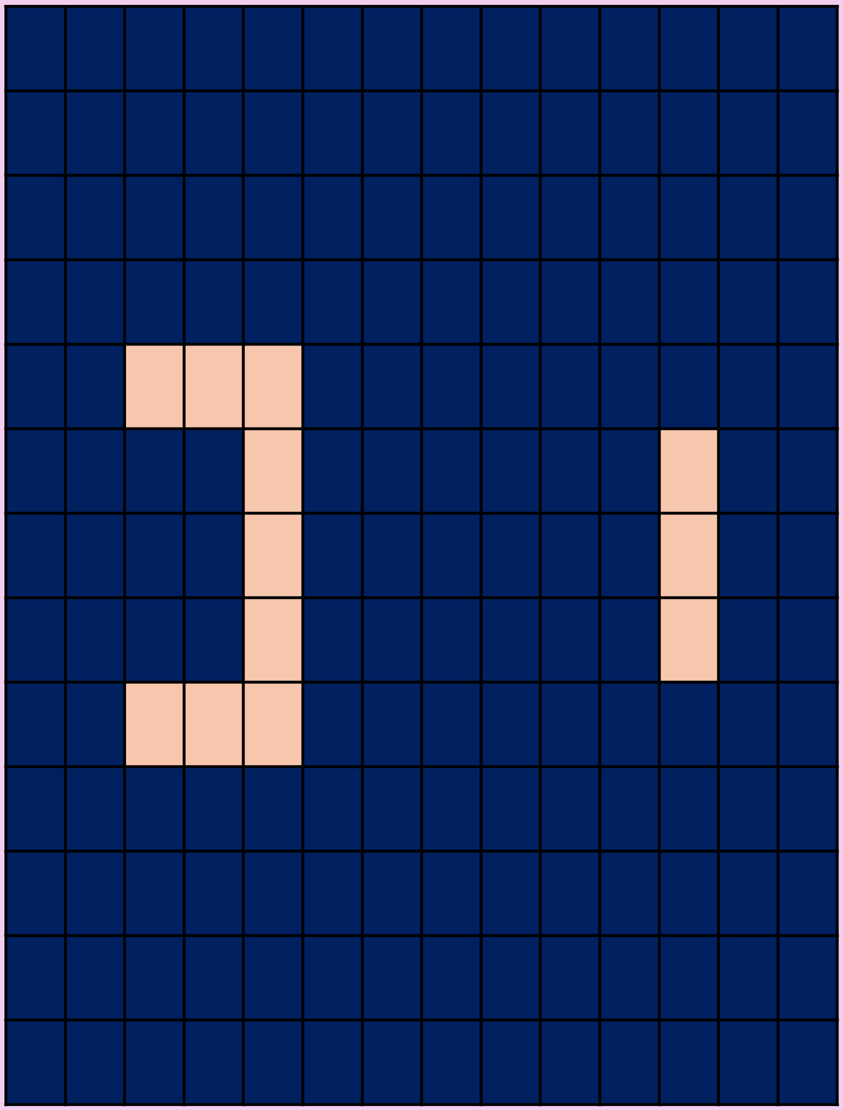
$(A \ominus B)$



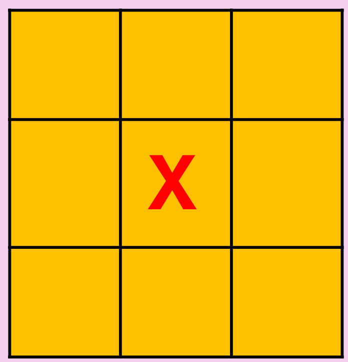
Opening Operation

$$A \circ B = (A \ominus B) \oplus B$$

$(A \ominus B)$

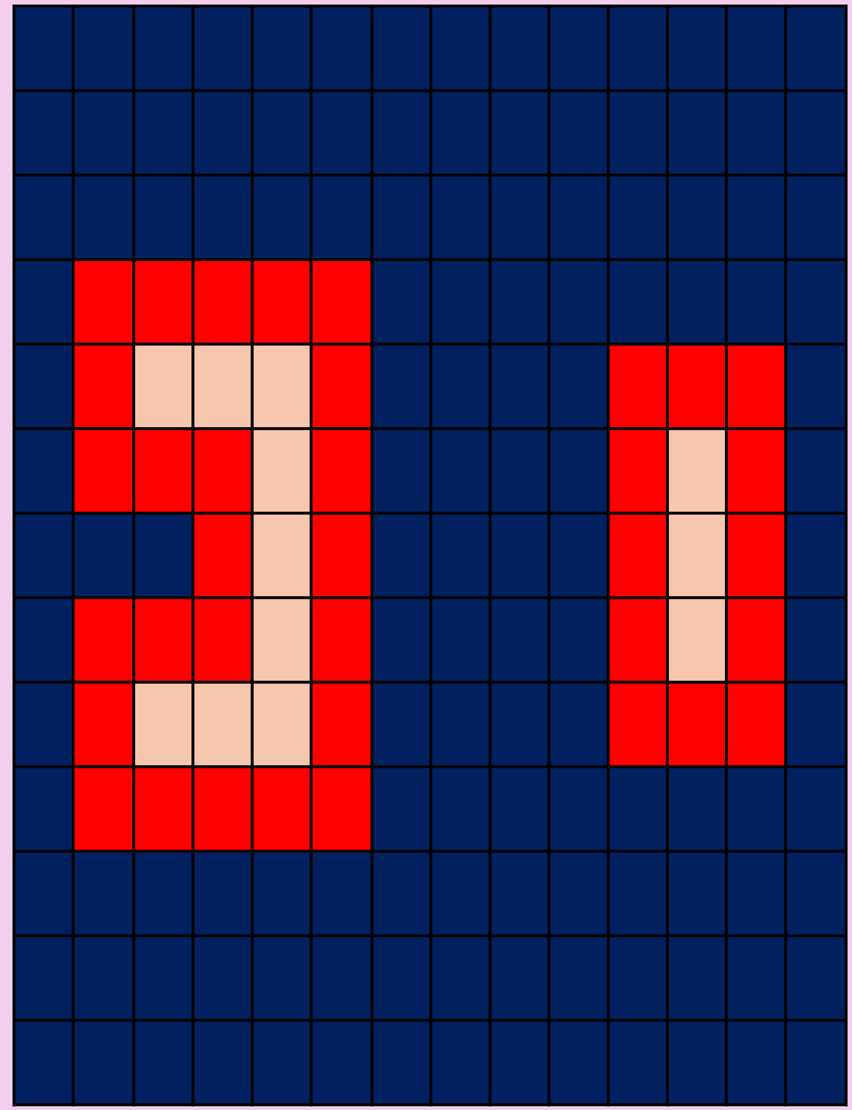


B



Opening

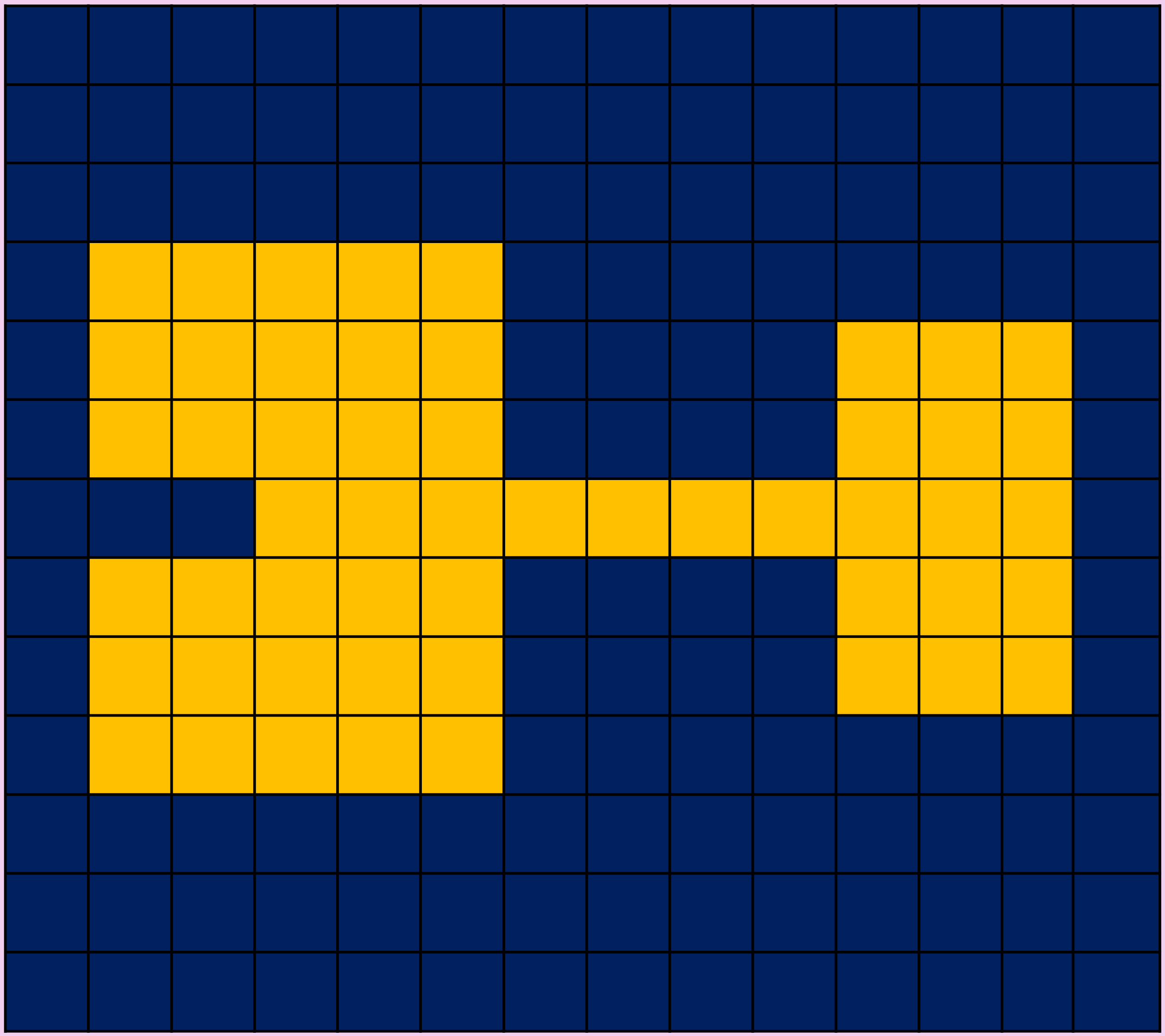
$A \circ B = (A \ominus B) \oplus B$



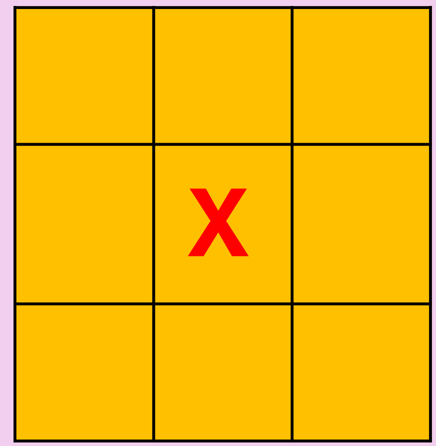
Closing Operation

- Closing Operation: $A \cdot B = (A \oplus B) \ominus B$

A



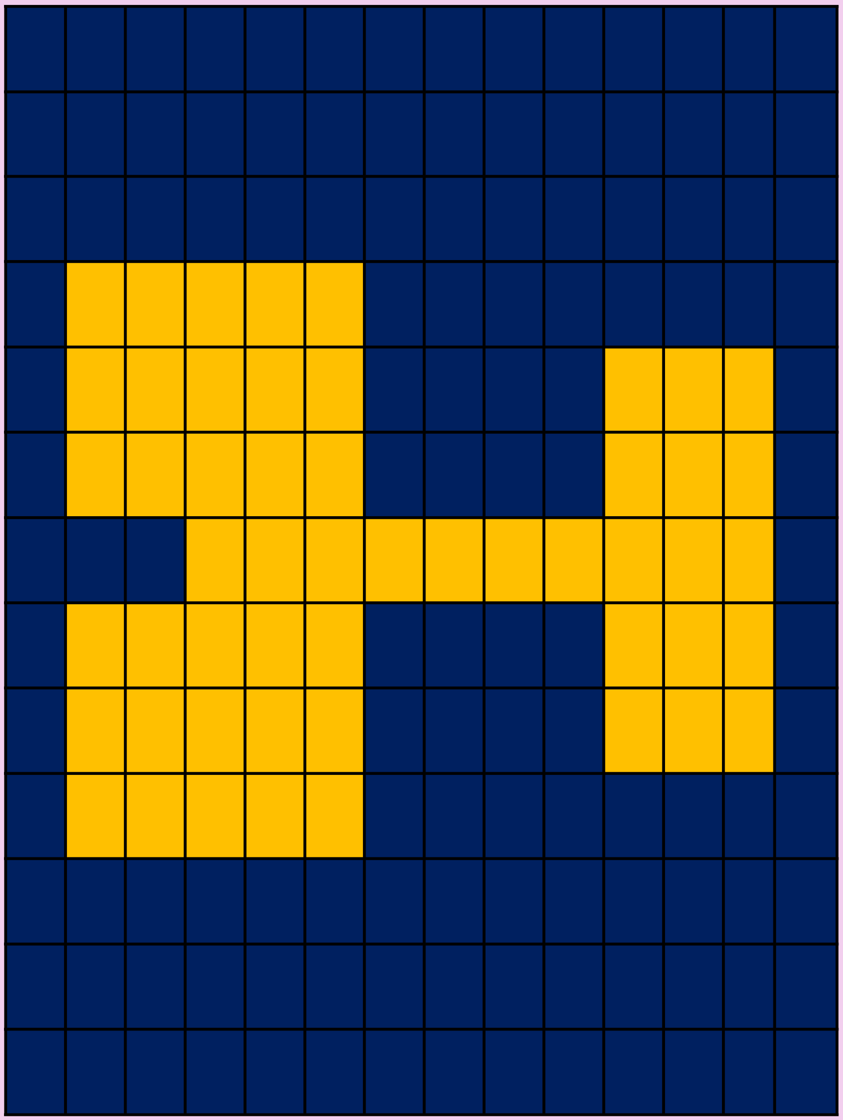
B



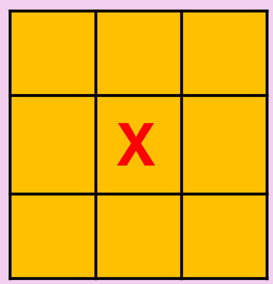
Closing Operation

$$A \cdot B = (A \oplus B) \ominus B$$

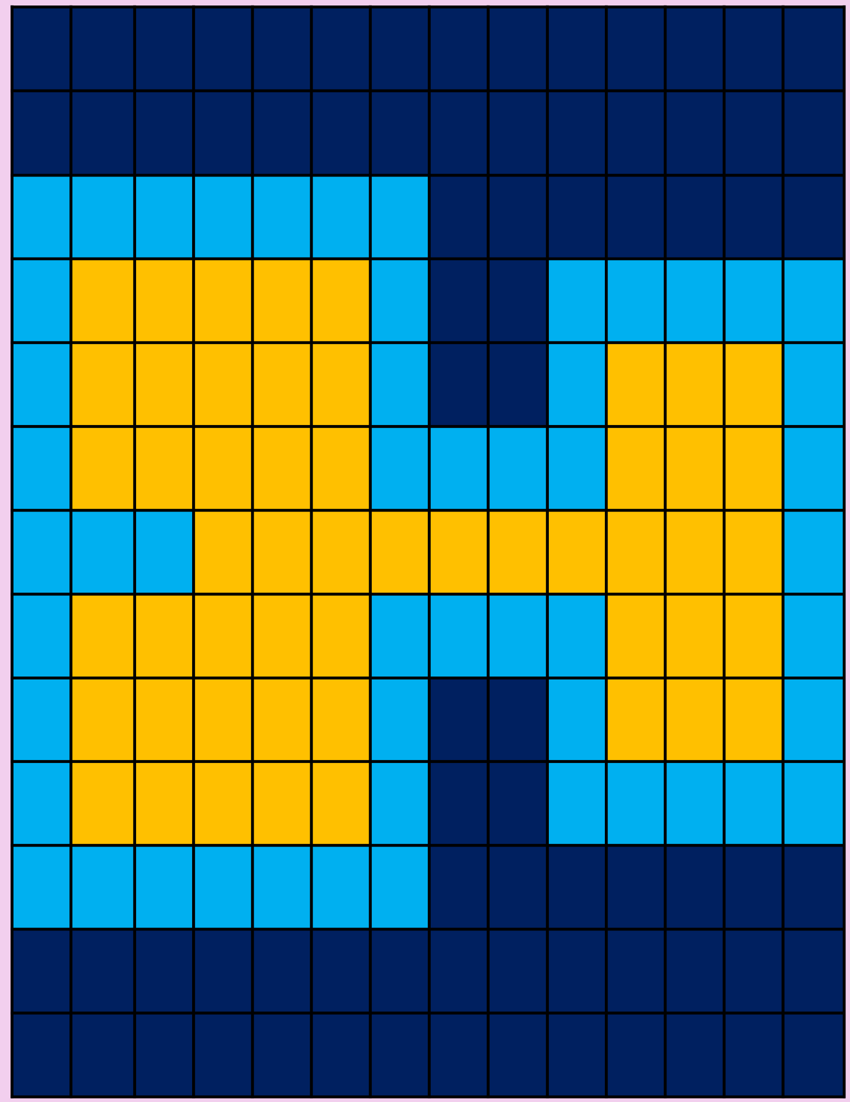
A



B



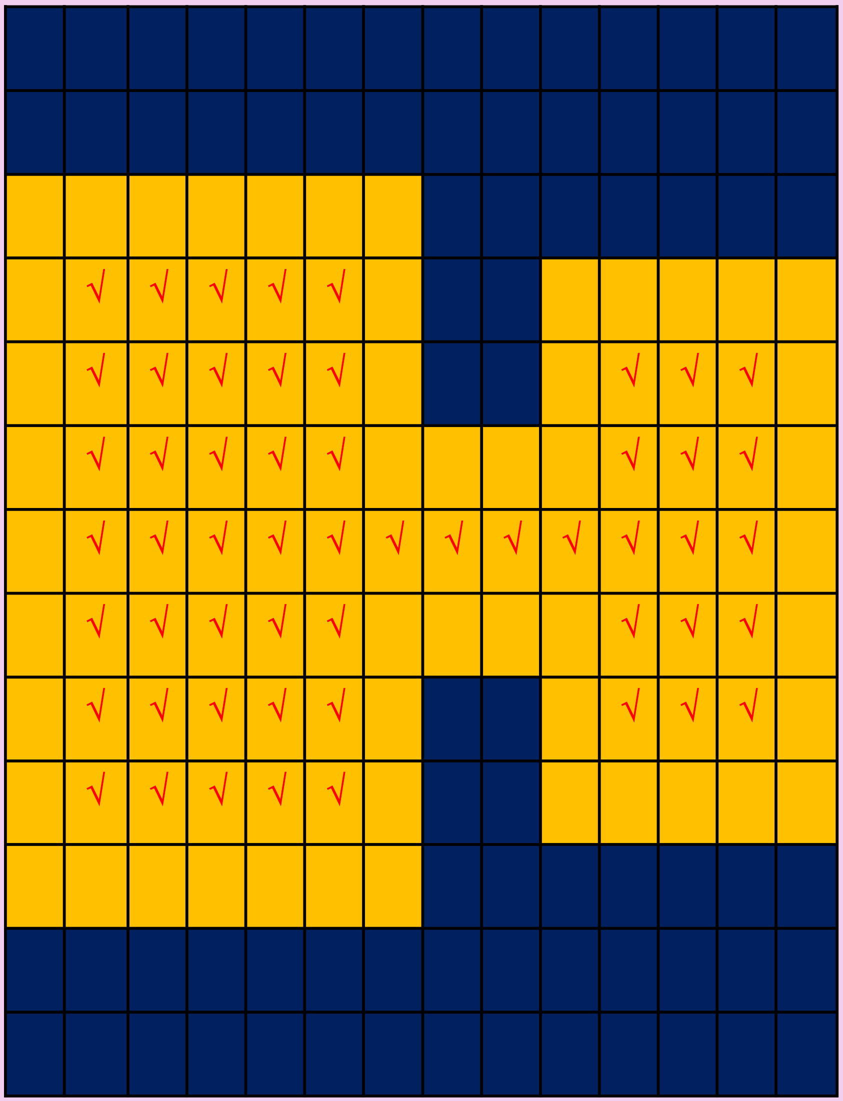
$(A \oplus B)$



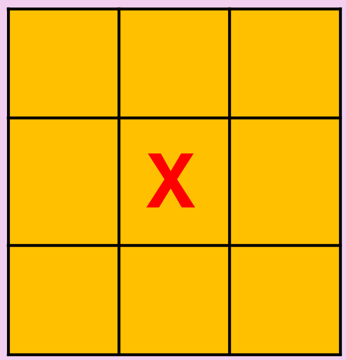
Closing Operation

$$A \cdot B = (A \oplus B) \ominus B$$

$(A \oplus B)$

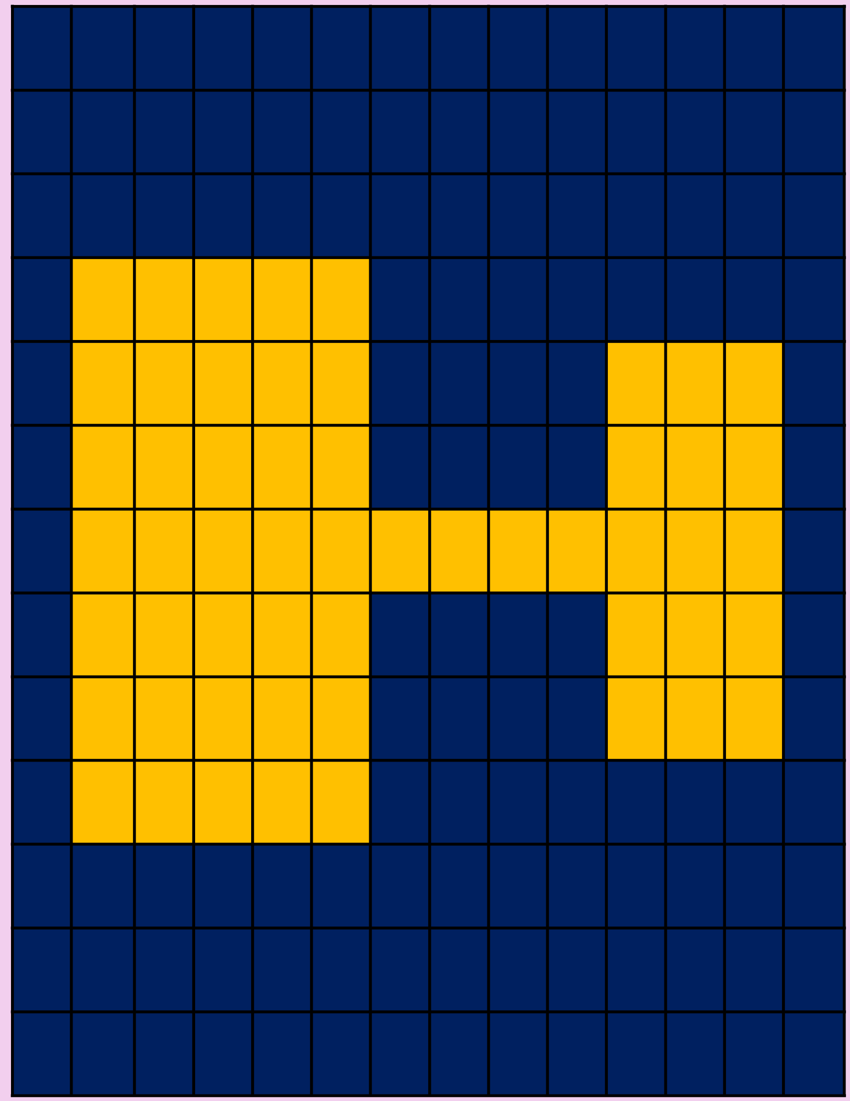


B



Closing

$A \cdot B = (A \oplus B) \ominus B$



Morphological Operations - 2

Hit or Miss Transform

Hit or Miss Transform

- The morphological hit-or-miss transform (HMT) is a basic tool for **Shape Detection.**
- It is useful to be able to match specified configurations of pixels in an image, such as isolated foreground pixels or pixels that are endpoints of line segment.
- It is a morphological operator for finding local patterns of pixels.
- The hit-or-miss transform is a general binary morphological operation that can be used to look for patterns of foreground and background pixels in an image

❑ **Concepts:**

  - Hit object
  - Miss background

Hit or Miss Transform

- Mathematical expression for Hit or Miss Transform is given as:

$$I \circledast S = (I \ominus S) \cap \left(I^c \ominus (W - S)\right)$$

- It can be written as

$$I \circledast S = (I \ominus S_1) \cap (I^c \ominus S_2)$$

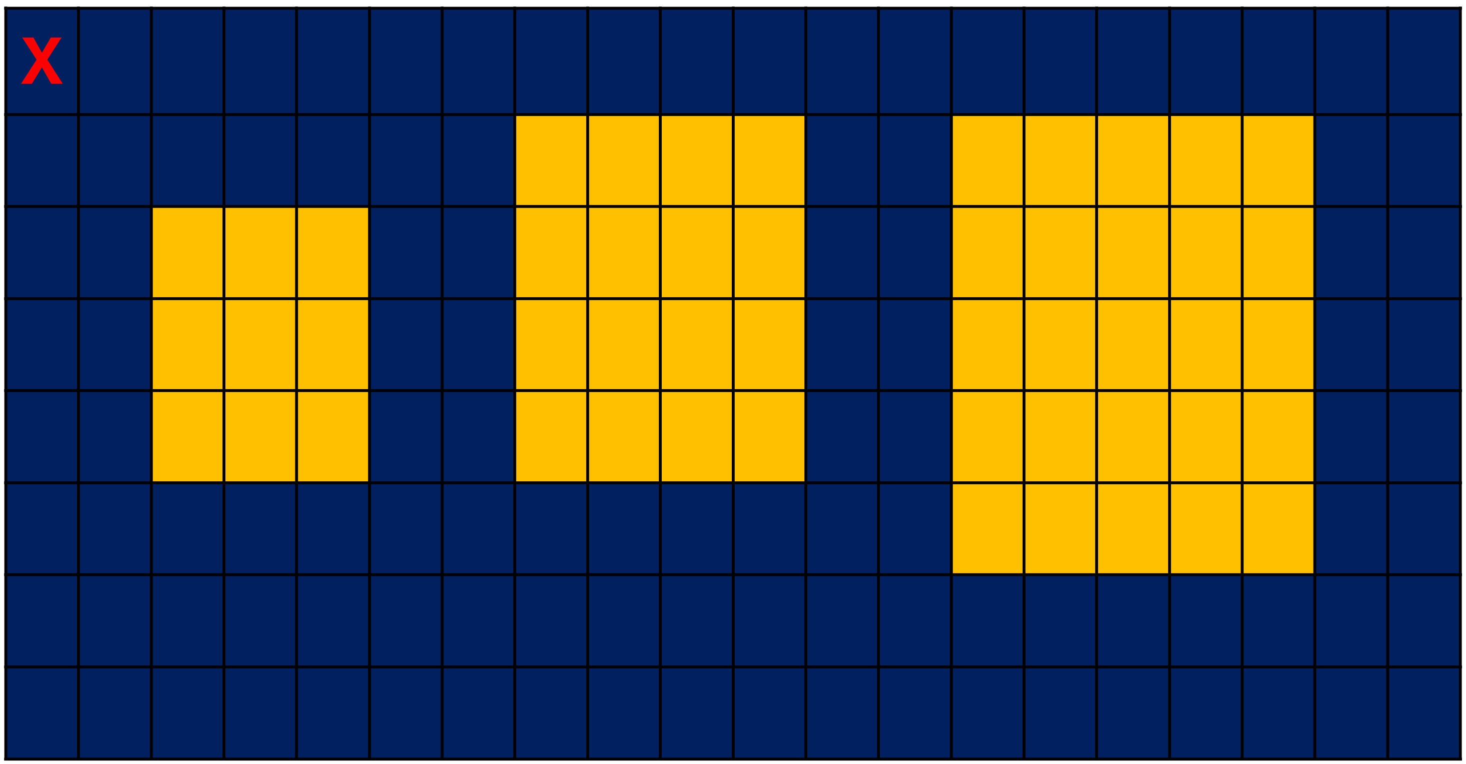
Where,

- S1 is the set formed from elements of S associated with an object (S in this case)
- S2 is the set of elements of S associated with the corresponding background (W – S)

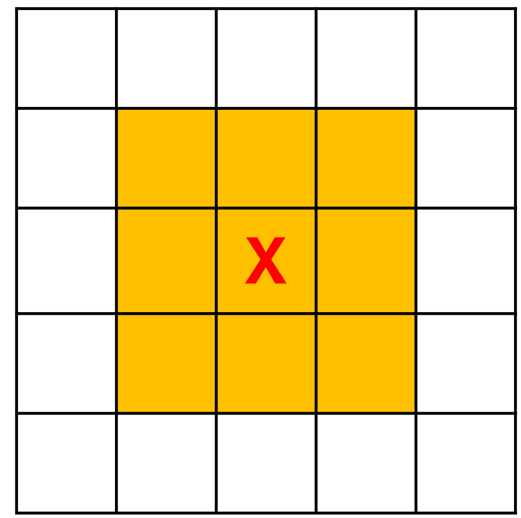
- **HIT Condition:** The set contains all the points at which, S1 found a match (HIT) in I and S2 found a match in I^C

Hit or Miss Transform

Example 1:



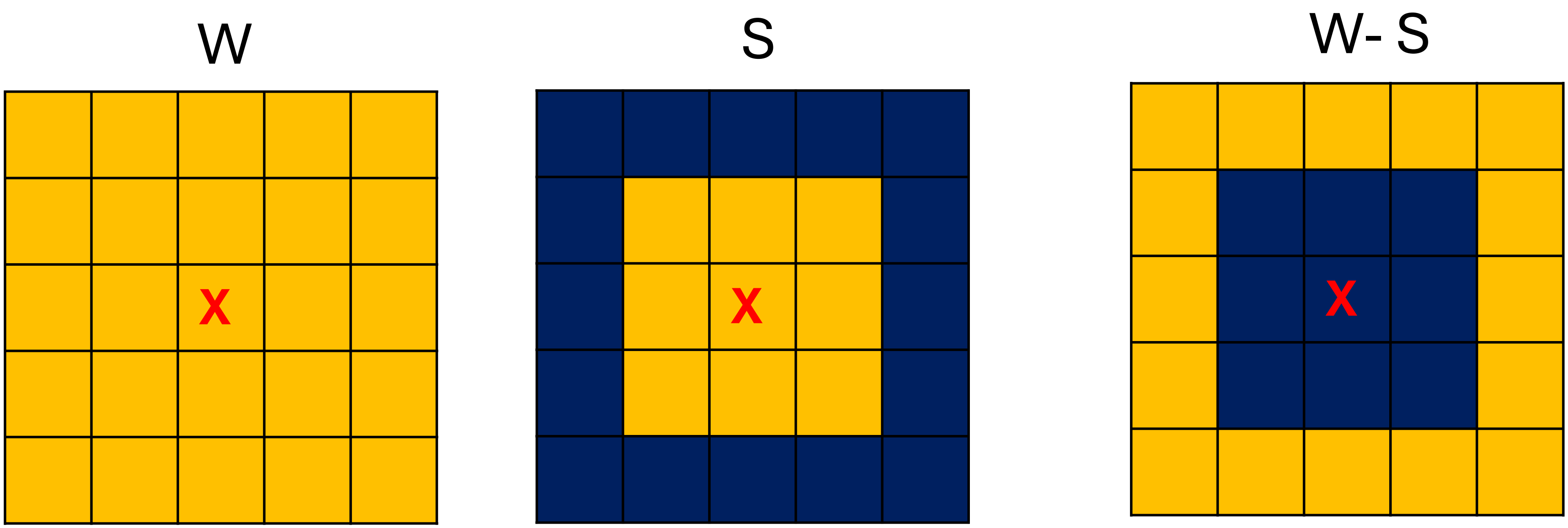
Input Image



Structuring Element

Hit or Miss Transform

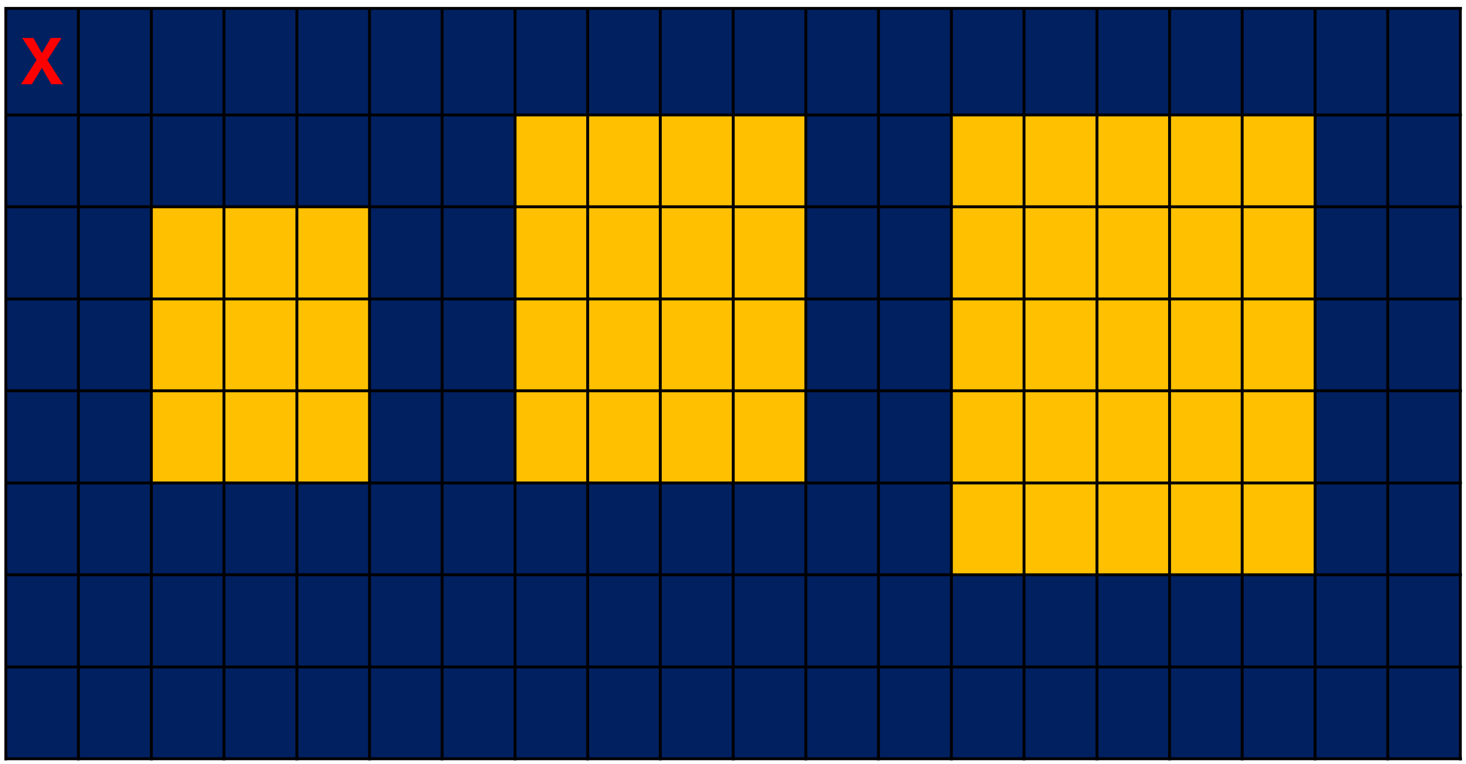
Step 1:



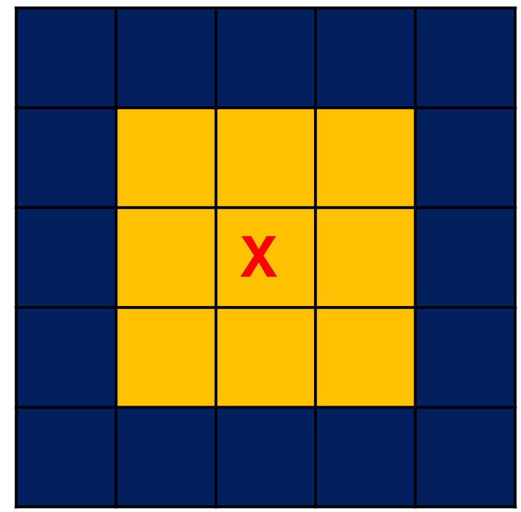
Hit or Miss Transform

Step 2: Perform the $(I \ominus S)$

I

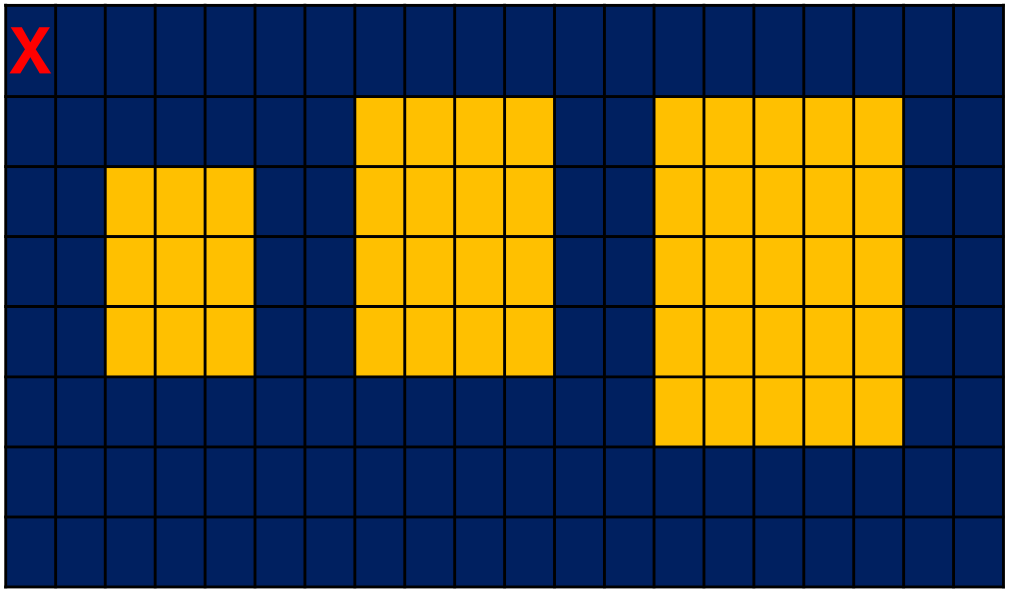


S

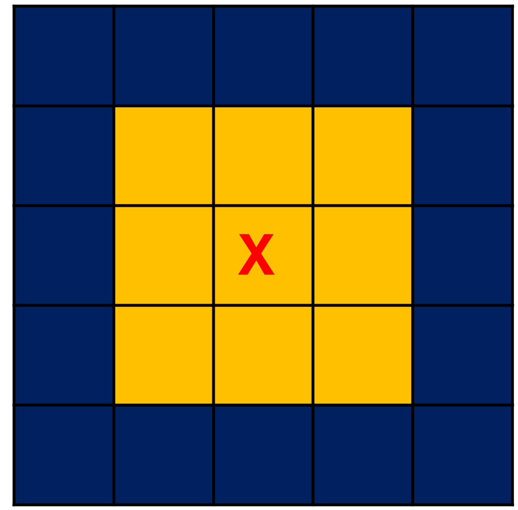


Hit or Miss Transform

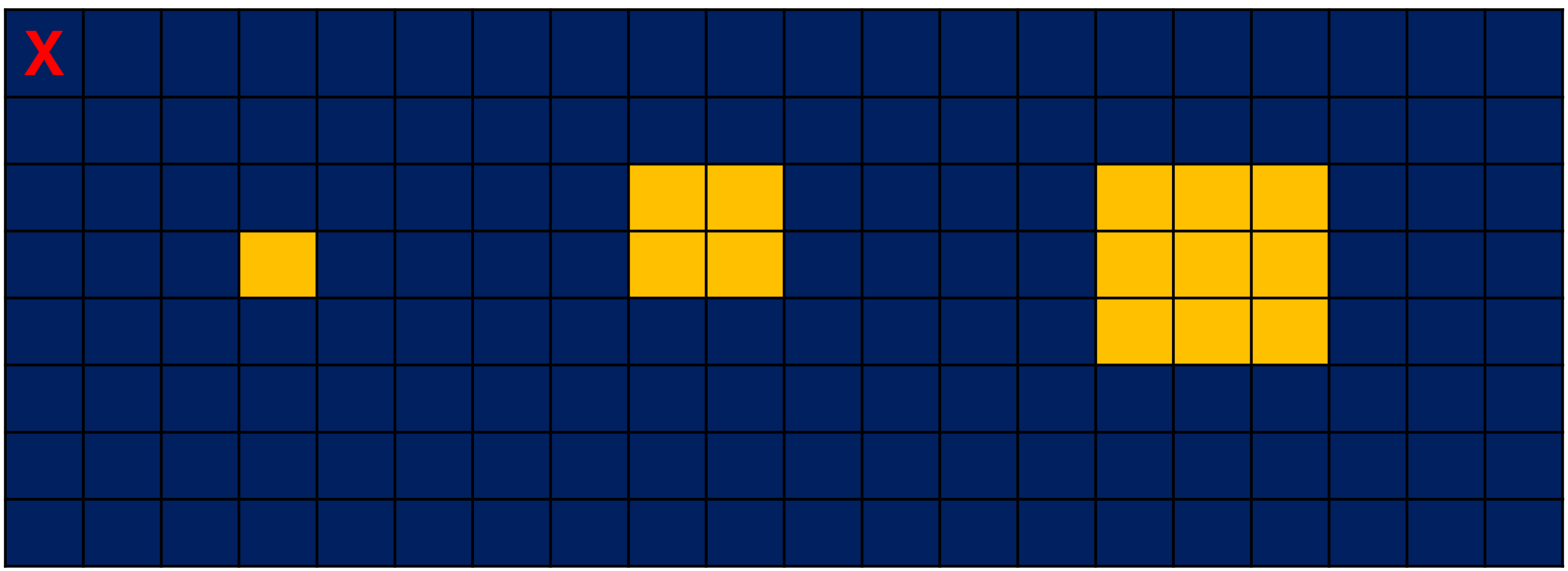
I



S



$I \ominus S$

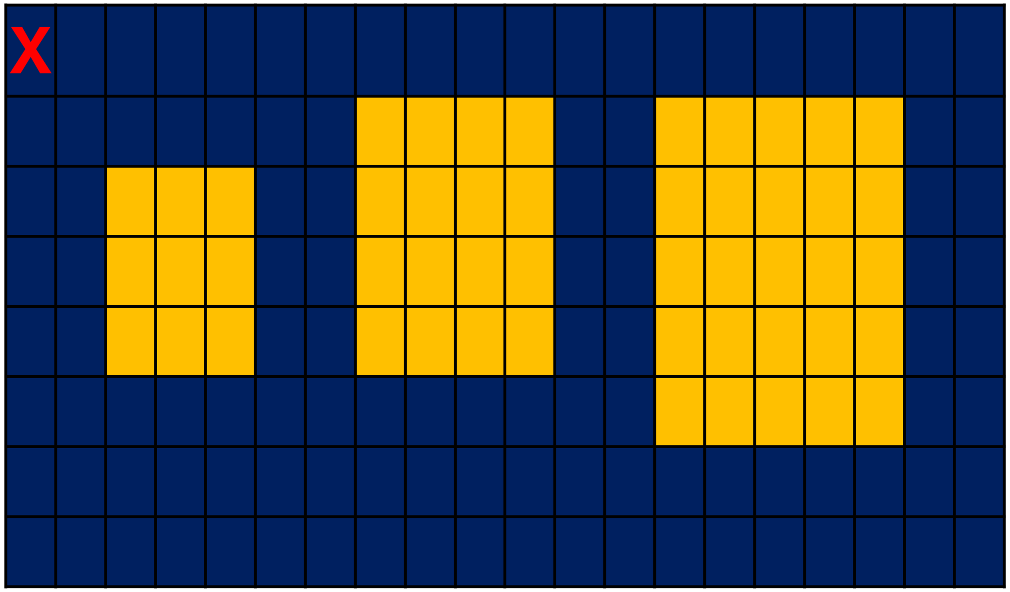


Hit or Miss Transform

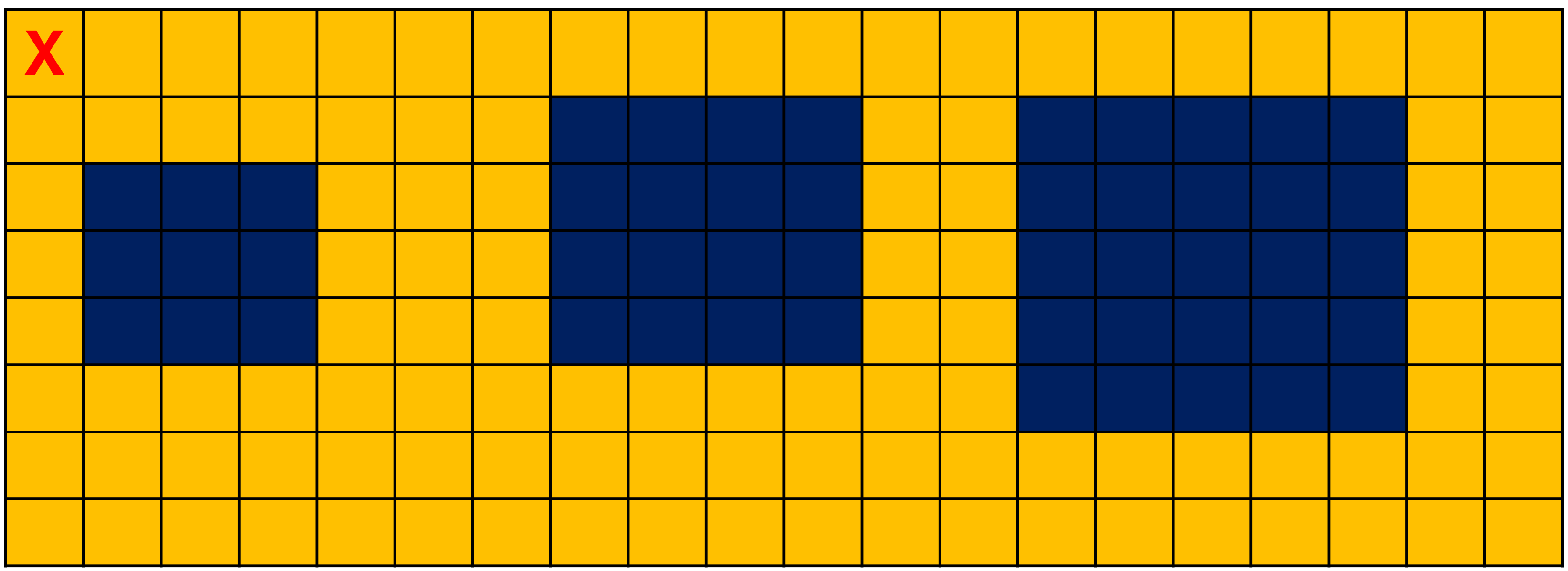
Step 3: Perform the I^c

$$I^c \ominus (W - S)$$

I



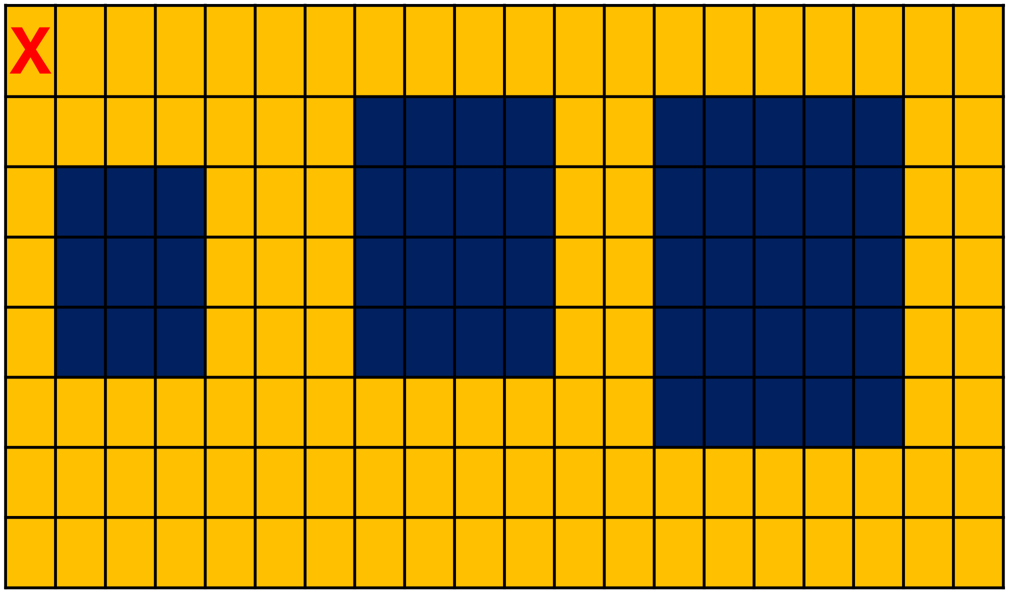
I^c



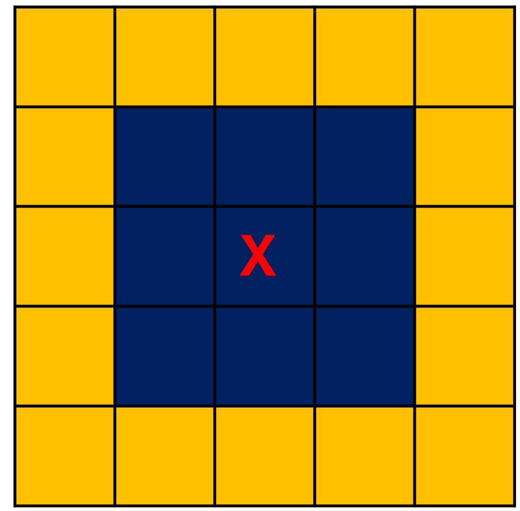
Hit or Miss Transform

Step 4: Determine $(I^C \ominus (W\text{-}S))$

I^c

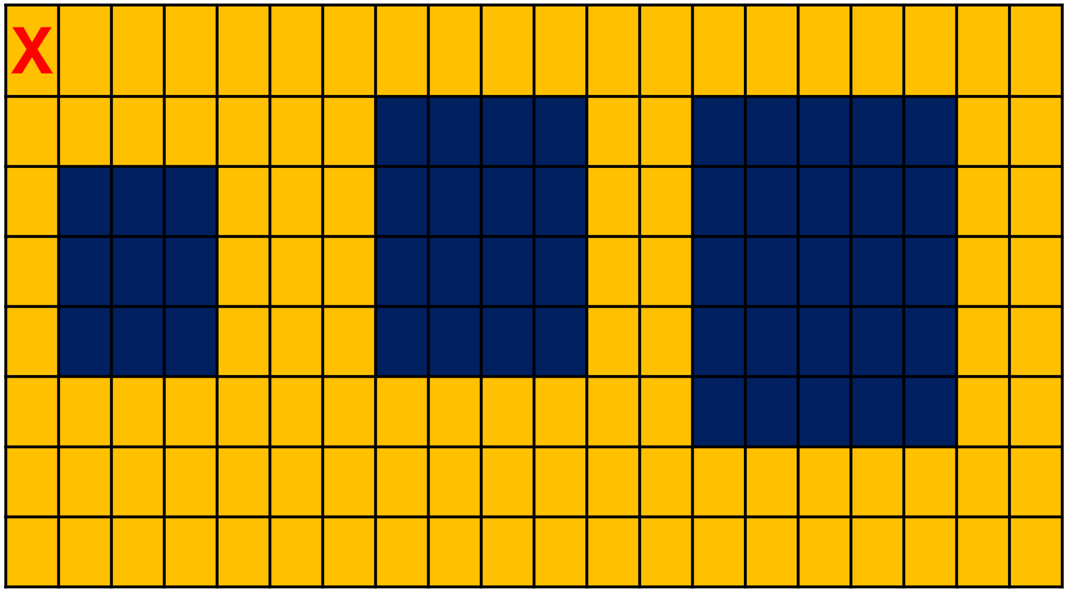


W- S

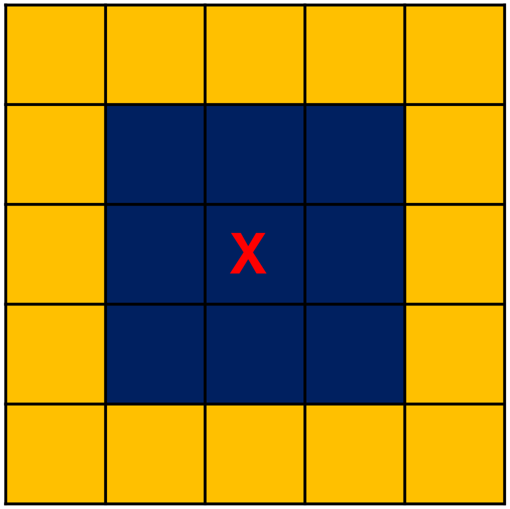


Hit or Miss Transform

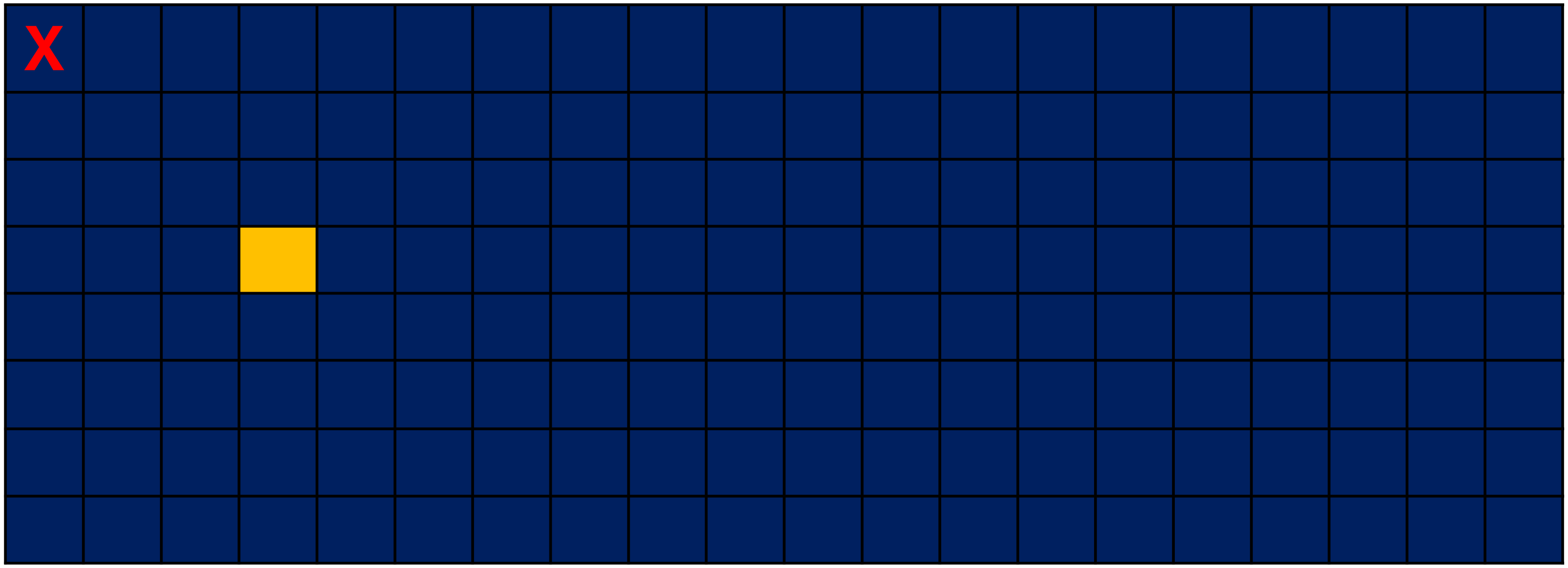
I^c



W- S



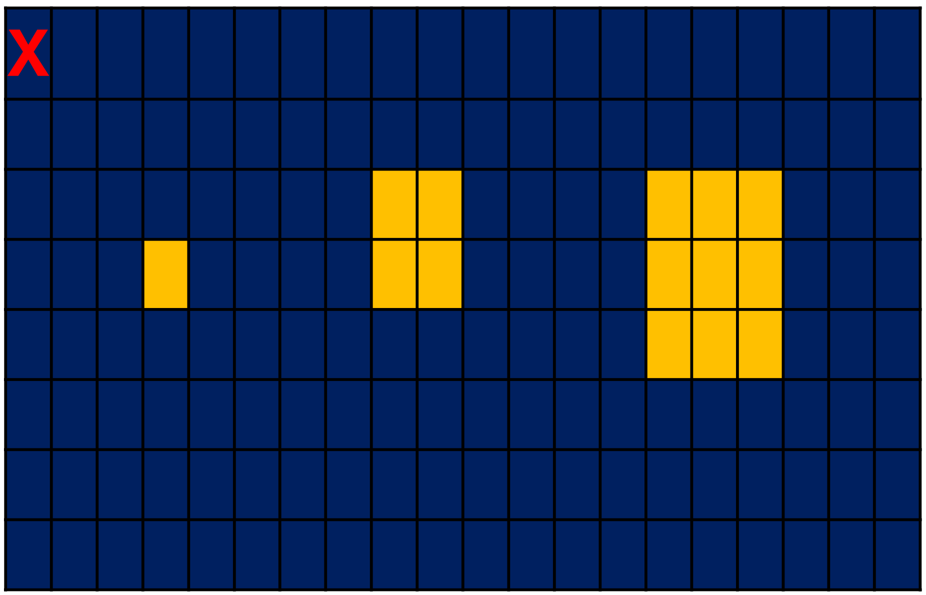
$I^C \ominus (W\text{-}S)$



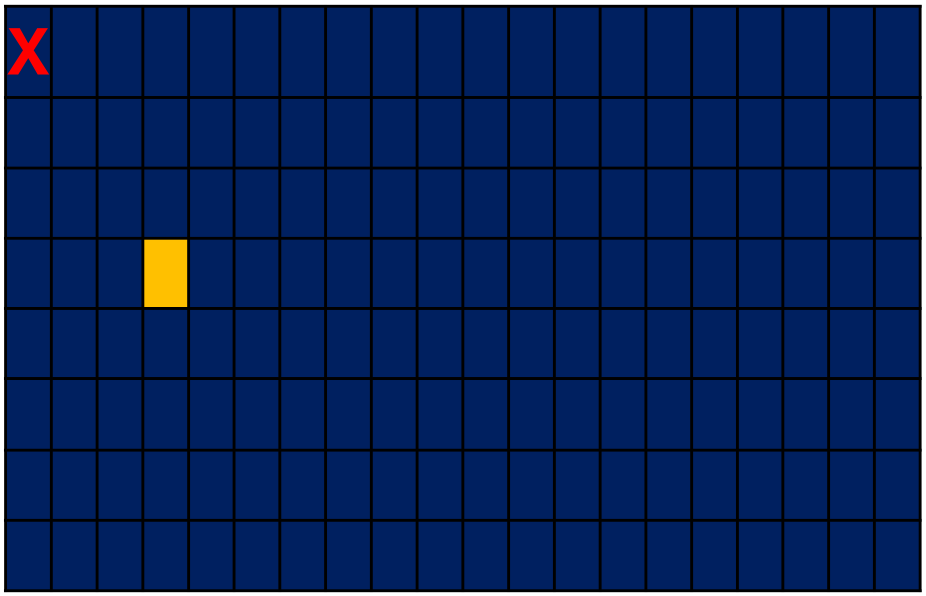
Hit or Miss Transform

Step 4: Perform the operations $(I \ominus S) \cap \left(I^c \ominus (W - S)\right)$

$I \ominus S$

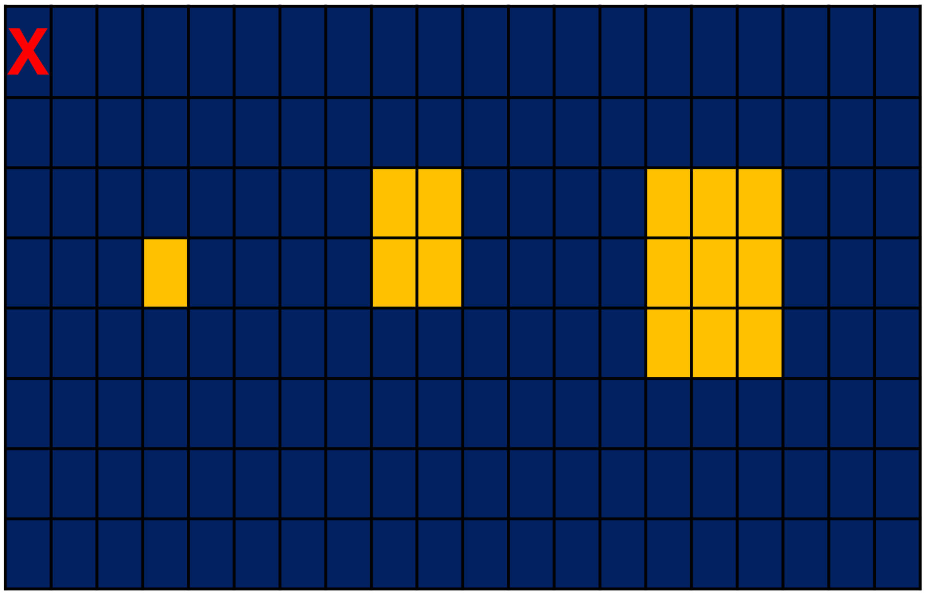


$I^c \ominus (W - S)$

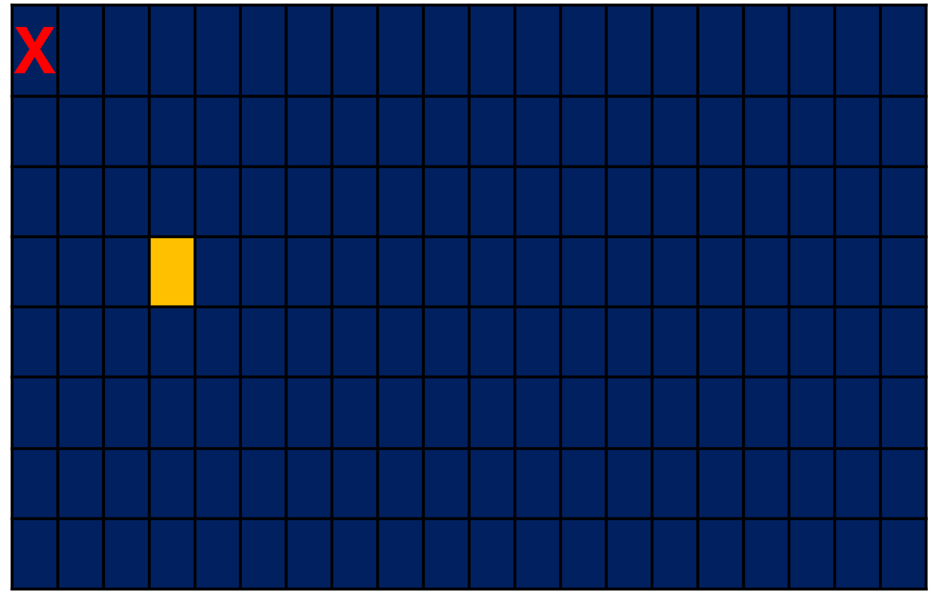


Hit or Miss Transform

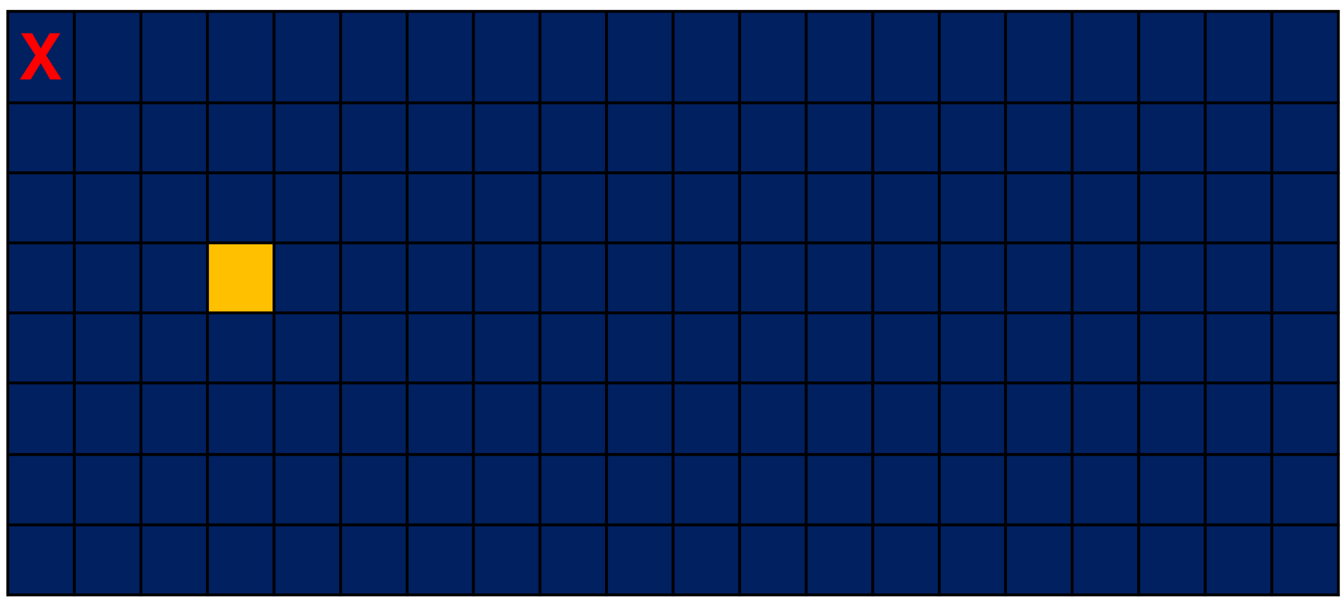
$I \ominus S$



$I^c \ominus (W - S)$

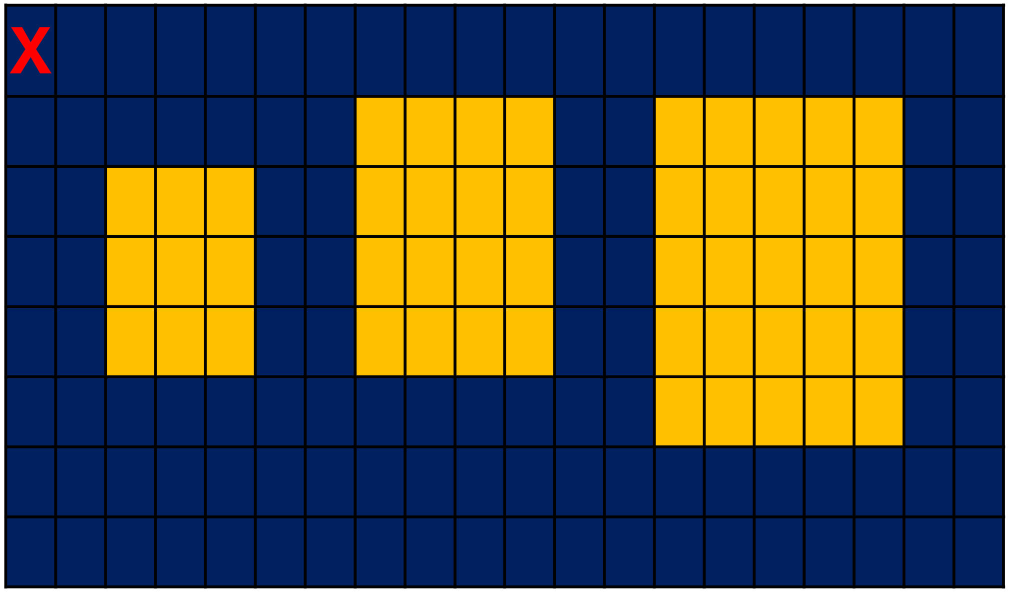


$(I \ominus S) \cap \left(I^c \ominus (W - S)\right)$

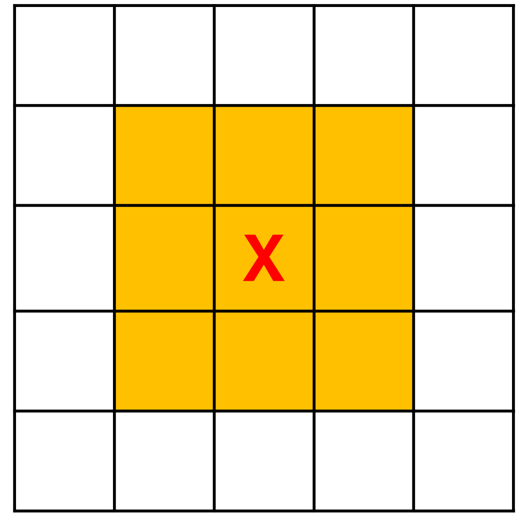


Hit or Miss Transform

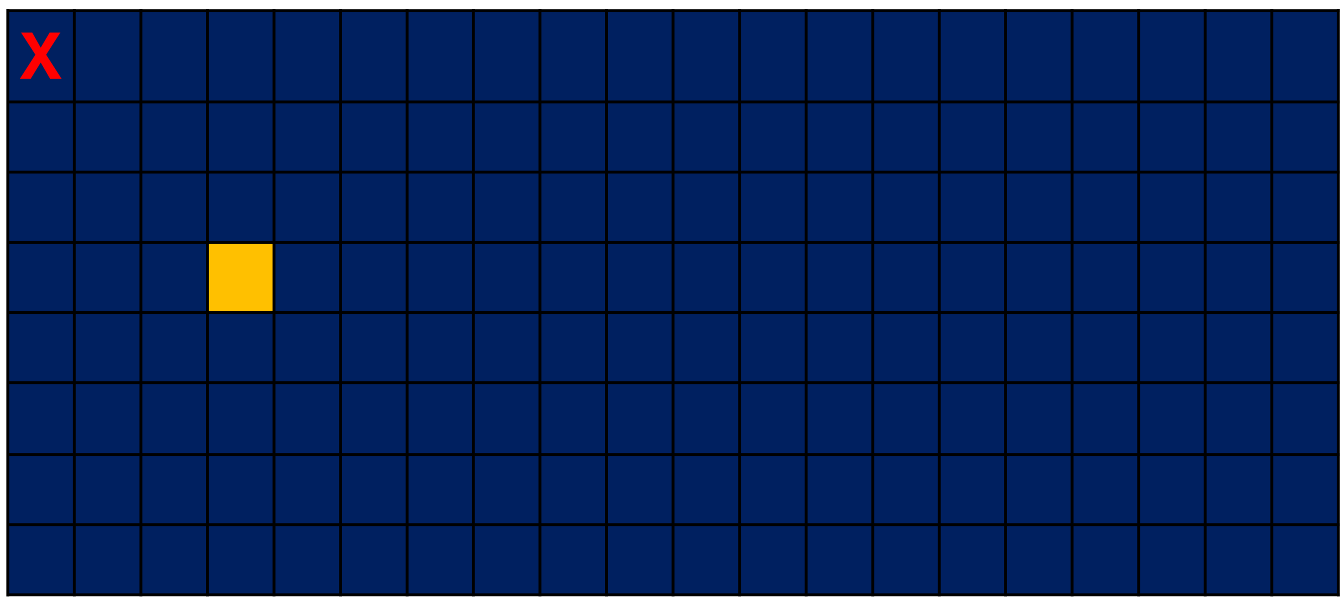
I



S



$(I \ominus S) \cap \left(I^c \ominus (W - S)\right)$



Boundary Extraction

Boundary Extraction

- The boundary of an object in an image is the set of pixels that have one or more neighbors
- A boundary is a ***contour in the image that represents a change in pixel ownership from one object to another***
- There are two types of boundary
  - **Internal boundary:** It contains boundary pixels that are inside the object

$$\boldsymbol{\beta}(\boldsymbol{I}) = \boldsymbol{I} - (\boldsymbol{I} \ominus \boldsymbol{S})$$

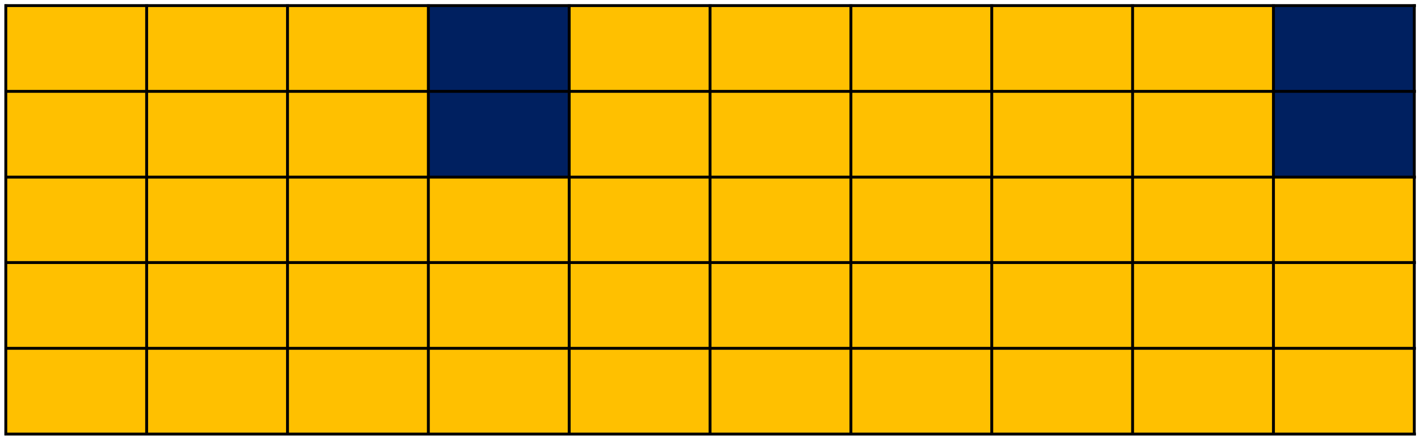
  - **External boundary:** It contains boundary pixels that are outside the object

$$\boldsymbol{\beta}(\boldsymbol{I}) = (\boldsymbol{I} \oplus \boldsymbol{S}) - \boldsymbol{I}$$

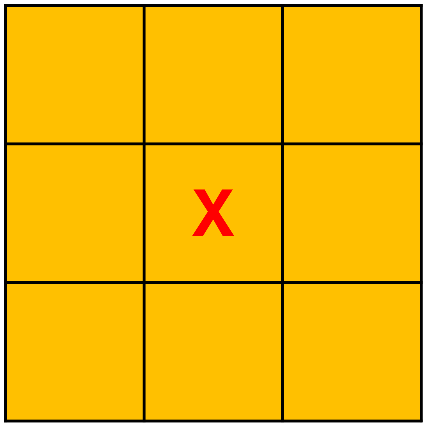
Internal Boundary Extraction

Example 1: Internal Boundary extraction $\boldsymbol{\beta(I) = I - (I \ominus S)}$

$\boldsymbol{I}$



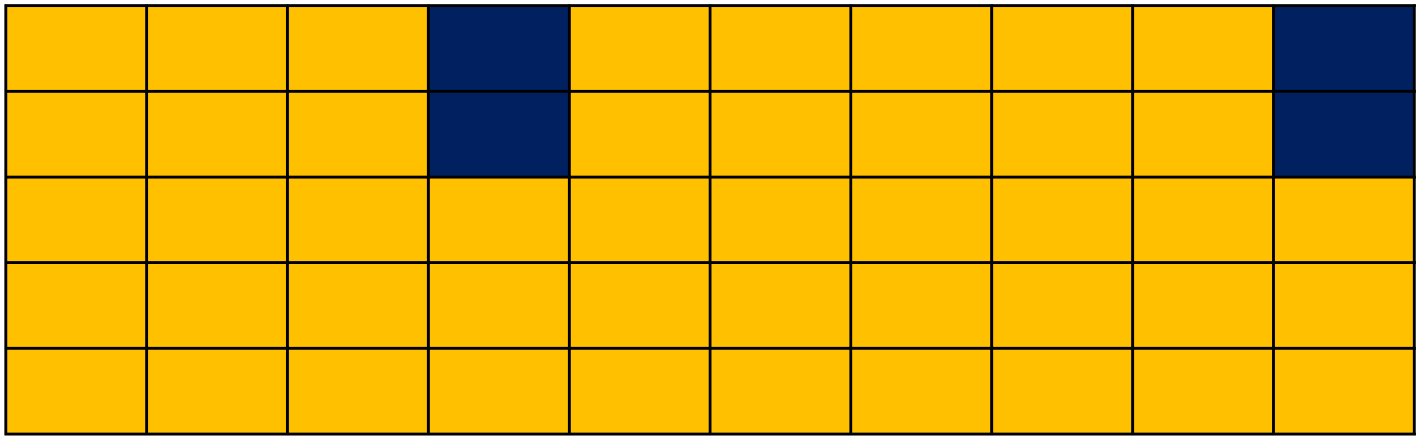
$\boldsymbol{S}$



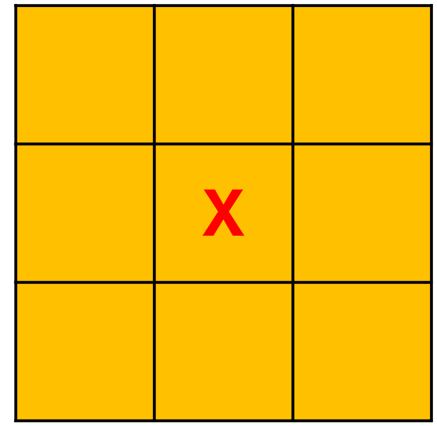
Internal Boundary Extraction

Example 1: Internal Boundary extraction $\boldsymbol{\beta(I) = I - (I \ominus S)}$

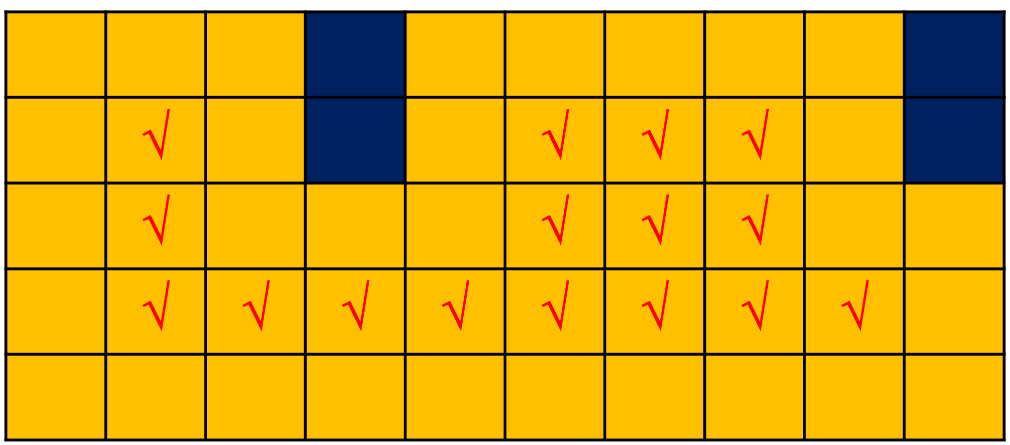
$\boldsymbol{I}$



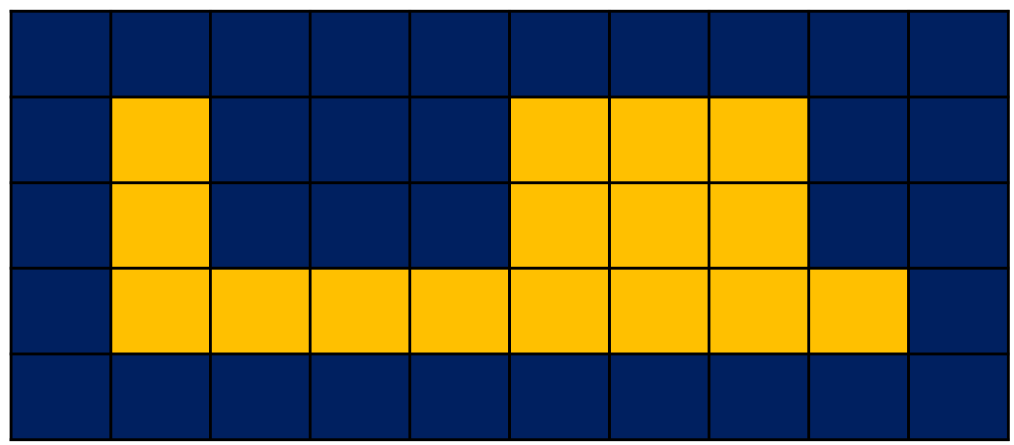
$\boldsymbol{S}$



Step 1: Determine $(\boldsymbol{I} \ominus \boldsymbol{S})$



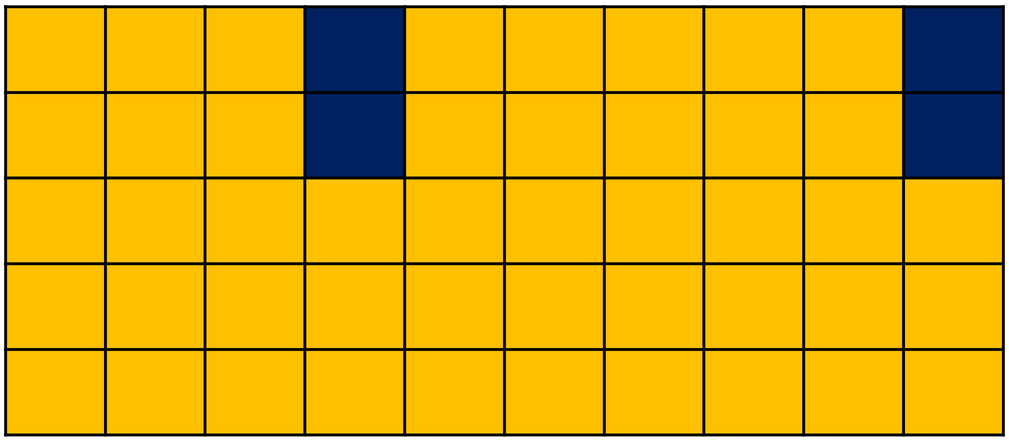
$(\boldsymbol{I} \ominus \boldsymbol{S})$



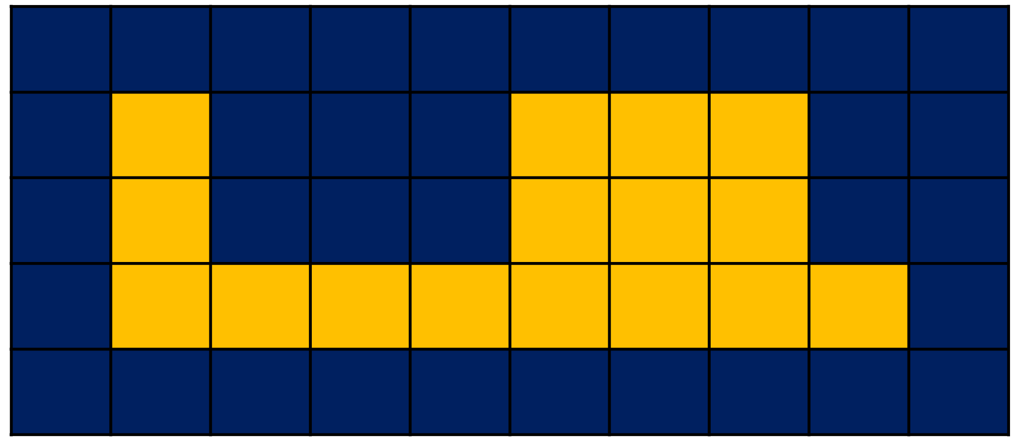
Internal Boundary Extraction

Step 2: Determine $I - (I \ominus S)$

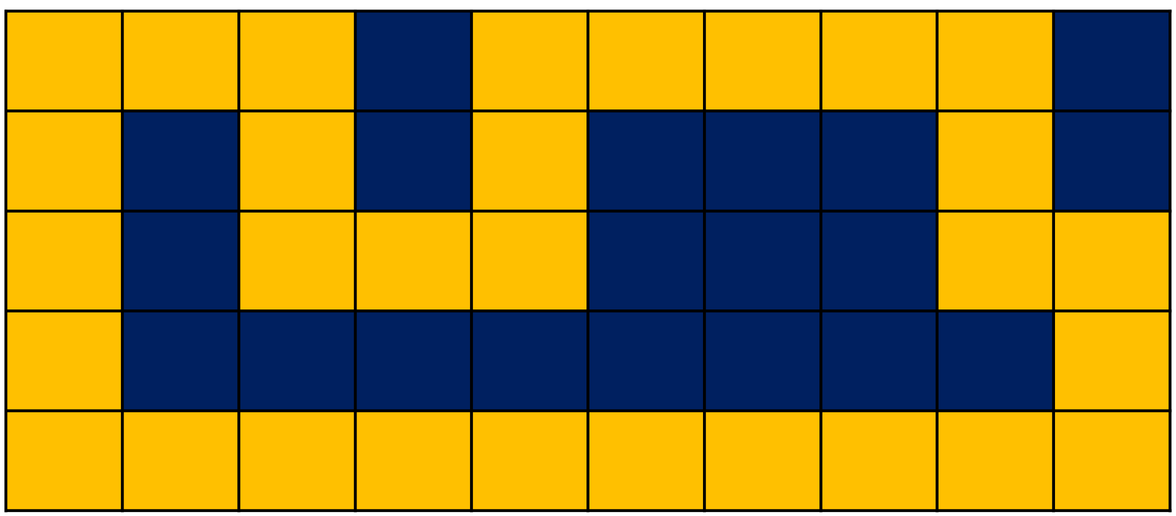
I



$(I \ominus S)$



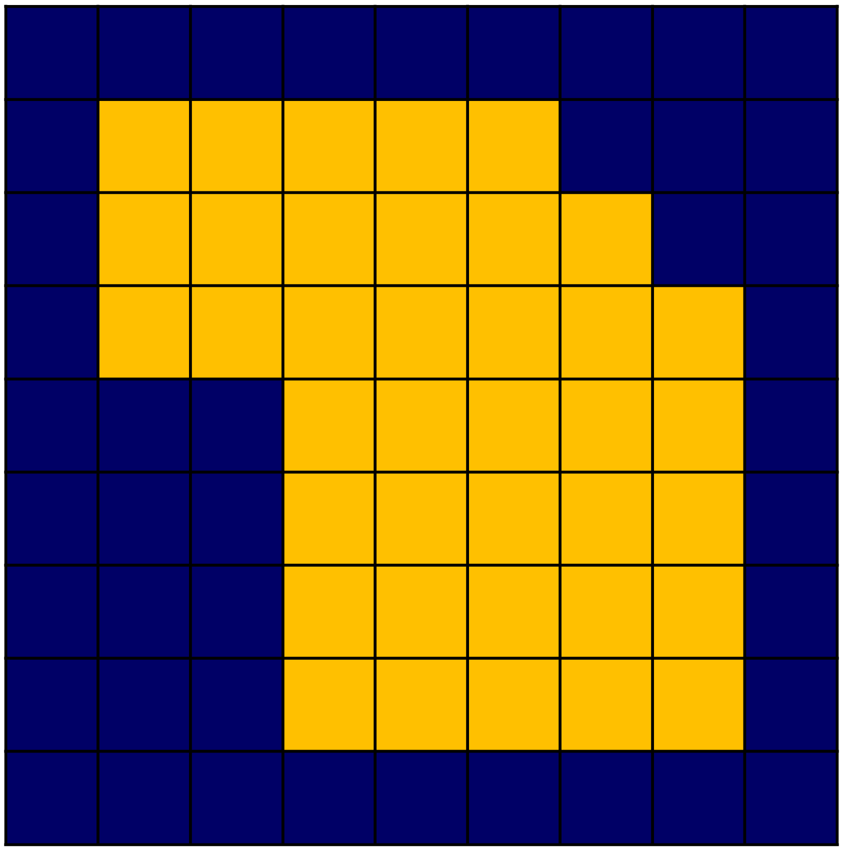
$$Internal\ Boundary = I - (I \ominus S)$$



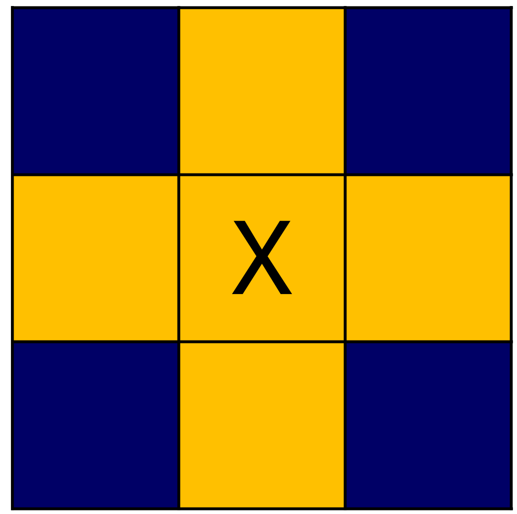
External Boundary Extraction

Example 2: Determine the external boundary $\boldsymbol{\beta(I) = (I \oplus S) - I}$

I

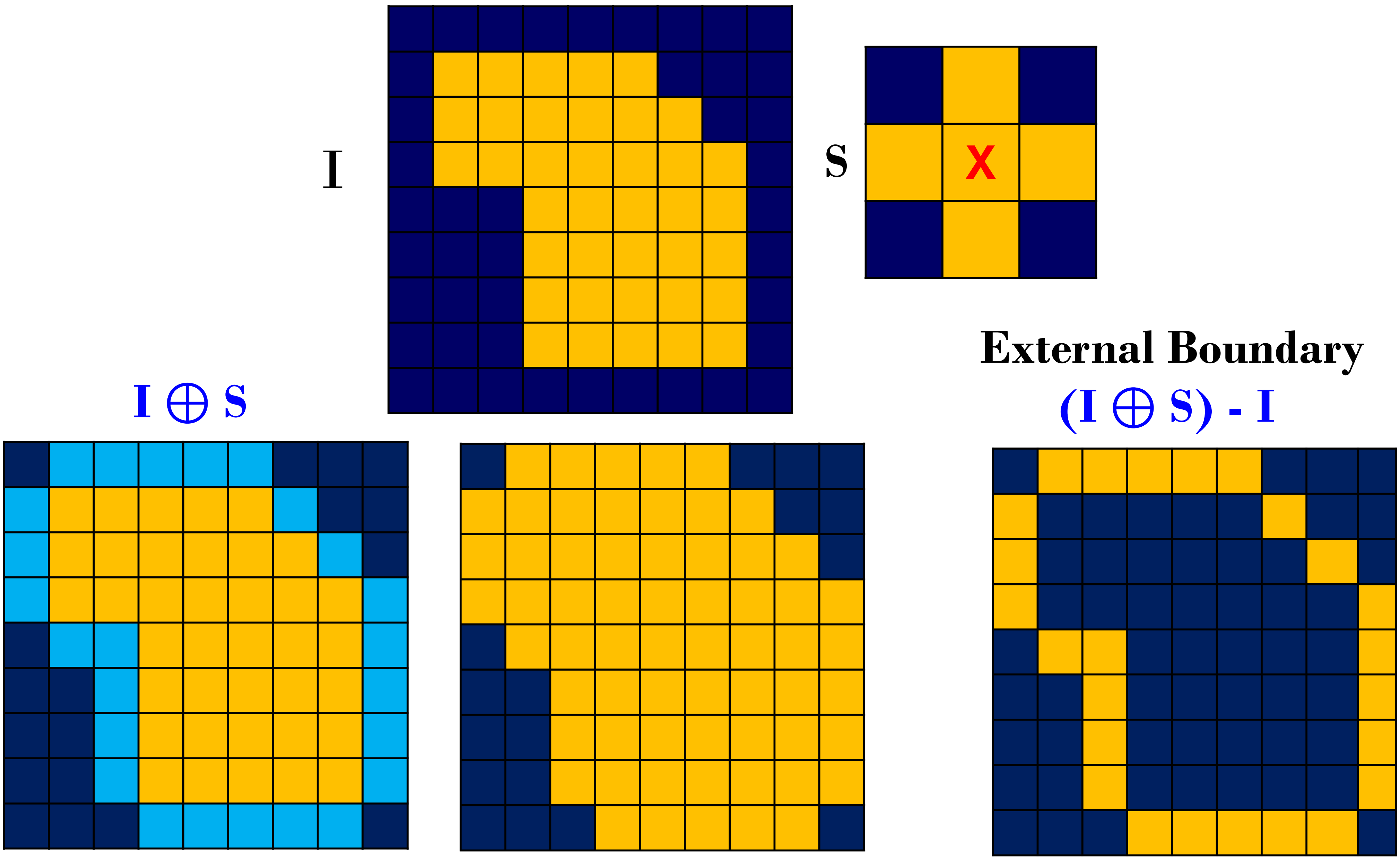


S



External Boundary Extraction

$$\beta(I) = (I \oplus S) - I$$



Region Filling

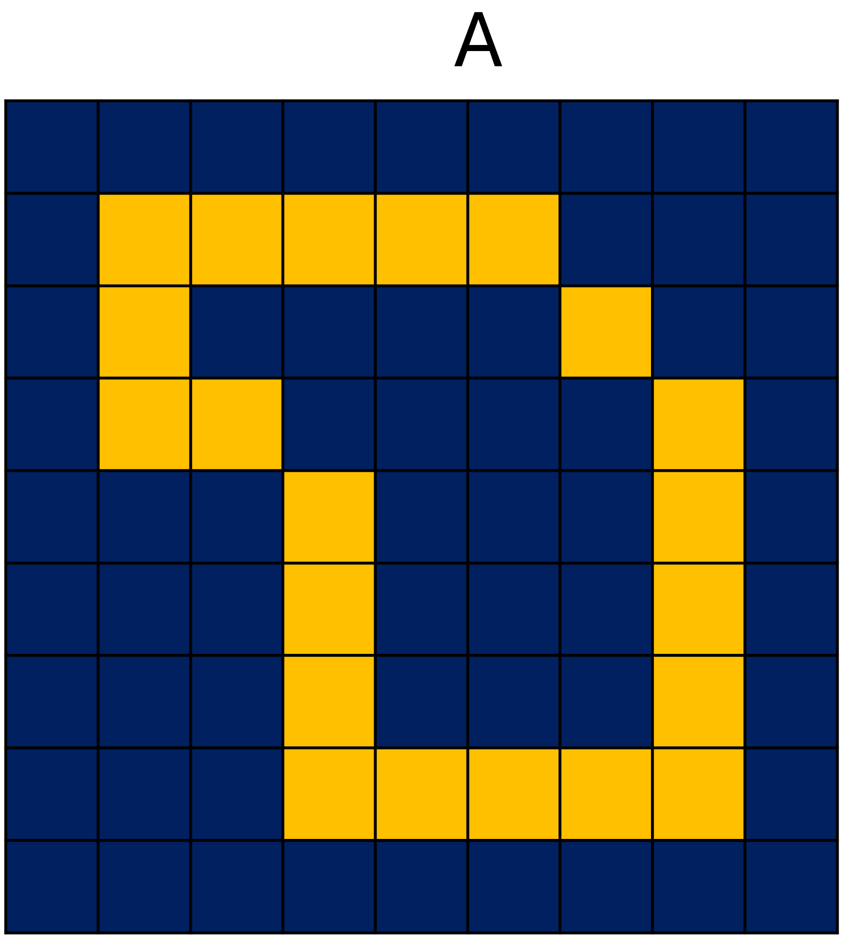
Region Filling

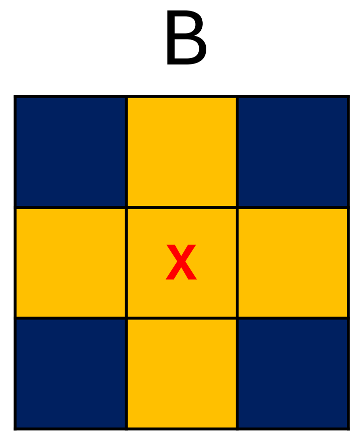
- Region filling is used to fill the selected region of the object.
- The process that achieves this is ass follows:

1. Let P be the point of the region to be filled. Initially P = X_0. Let A be the subset of element that represents the region. Let K denote the iteration.
2. Let B be the structuring element.
3. Repeat steps 4 to 6
4. Set K = K +1
5. $X_K = (X_{K-1} \oplus B) \cap A^c$ and store the result
6. If $X_K = X_{K-1}$ (point of convergence), then STOP
7. Exit

Region Filling

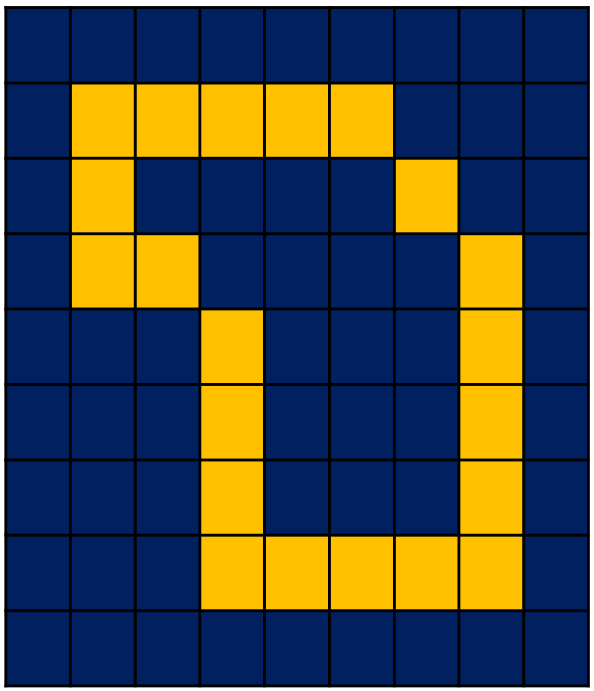
Example 1:



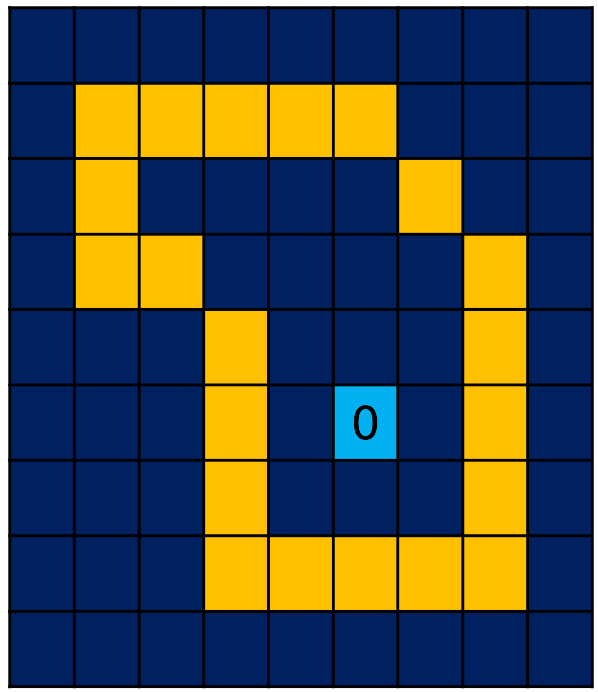


Region Filling

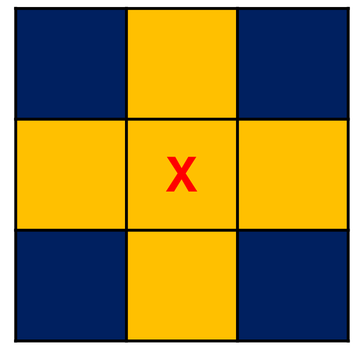
A



X_0



B



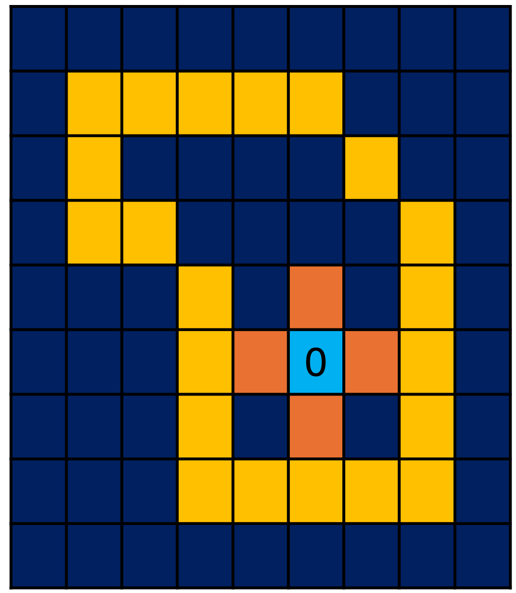
Region Filling

$$X_K = (X_{K-1} \oplus B) \cap A^c$$

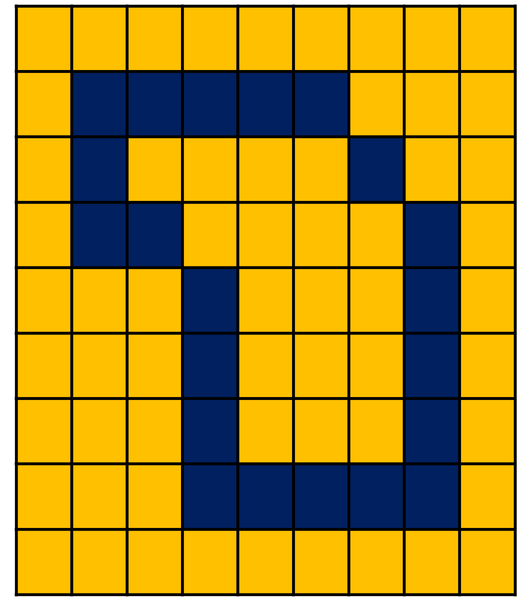
For K=1

$$X_1 = (X_0 \oplus B) \cap A^c$$

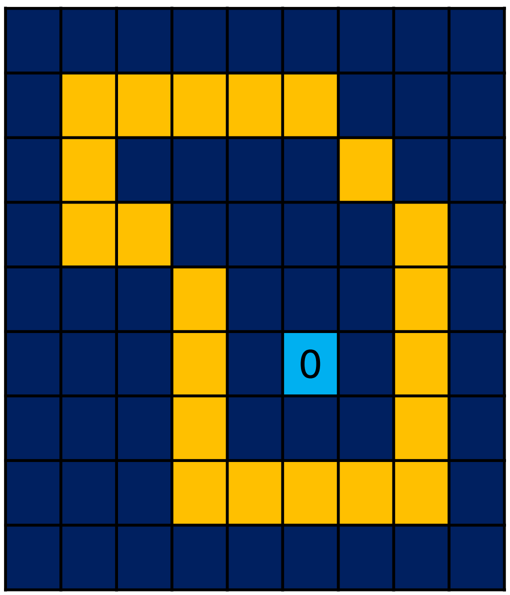
$(\boldsymbol{X_0} \oplus \boldsymbol{B})$



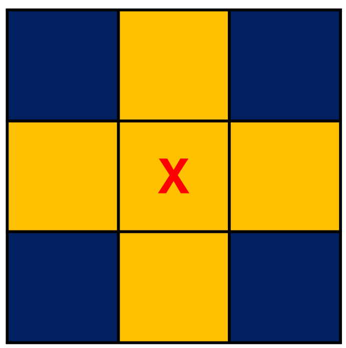
A^C



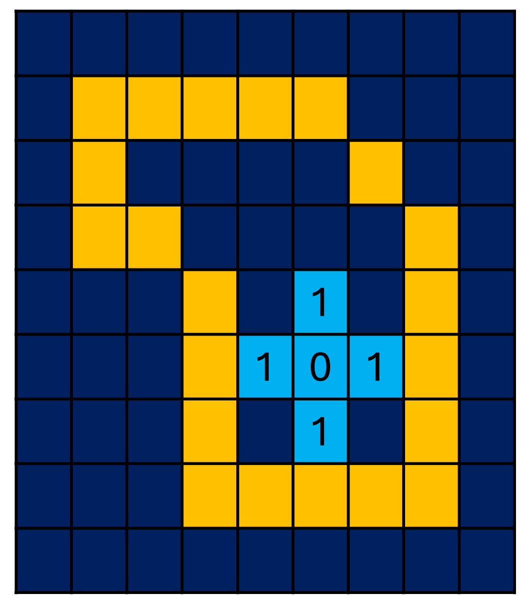
$\boldsymbol{X_0}$



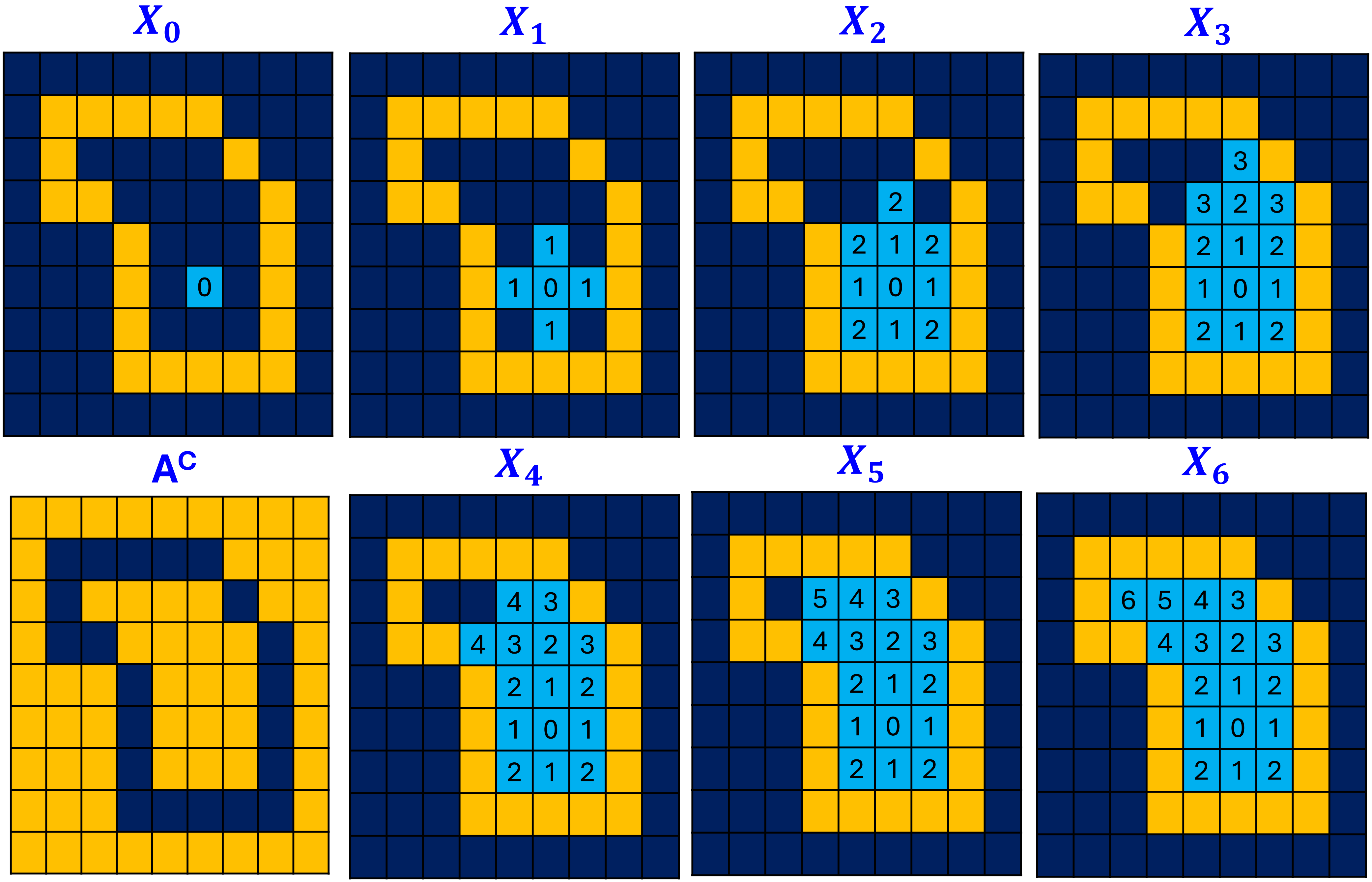
B



$$\boldsymbol{X_1 = (X_0 \oplus B) \cap A^c}$$

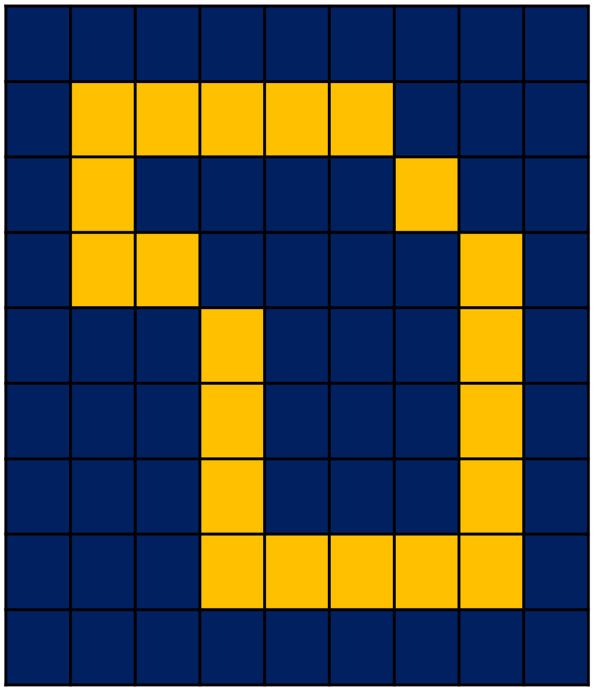


Region Filling

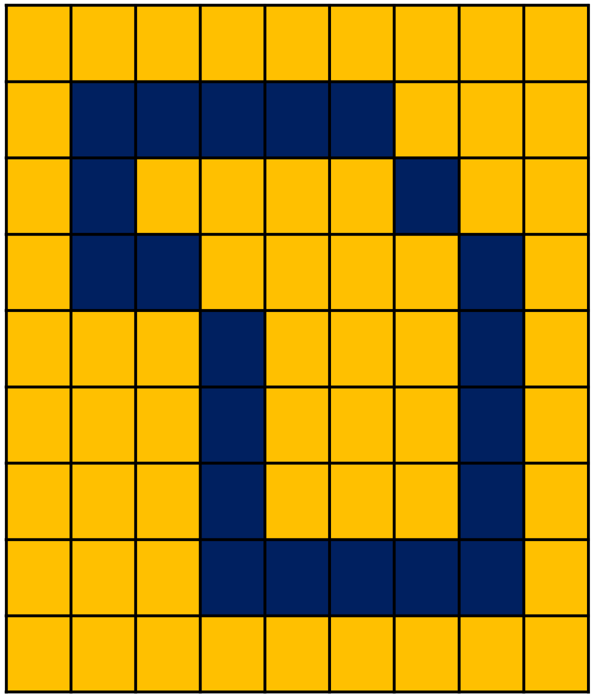


Region Filling

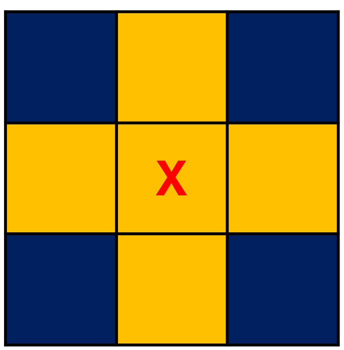
A



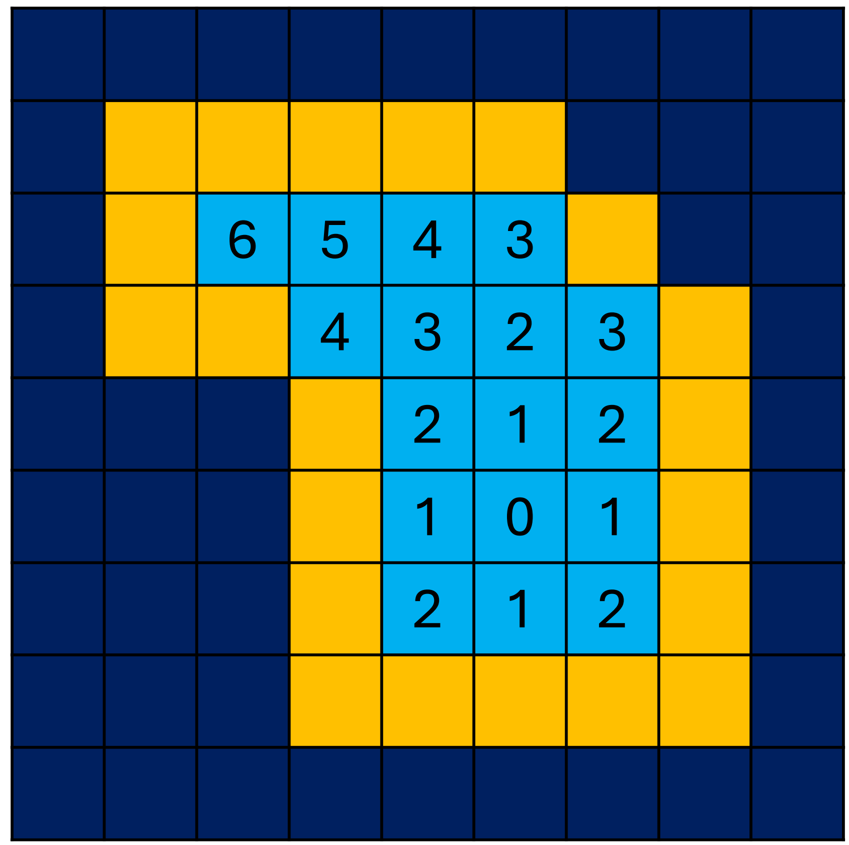
A^C



B



Region Filling Operation



Thank You