

Filtrage d'un signal codé en Jeu de la vie

Filtrage d'un signal codé en Jeu de la vie

I. Présentation du modèle choisi

1. Règles du jeu de la vie de Connway
2. Représentation discrète en base 2
3. Faisceaux de gliders

II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)
2. Portes logiques
3. Additionneur et multiplicateur
4. Un filtre modulable

III. Expériences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée) appliqués à ces différents filtres
2. Nombre de générations nécessaires pour arriver au resultat
3. Implémentation en C

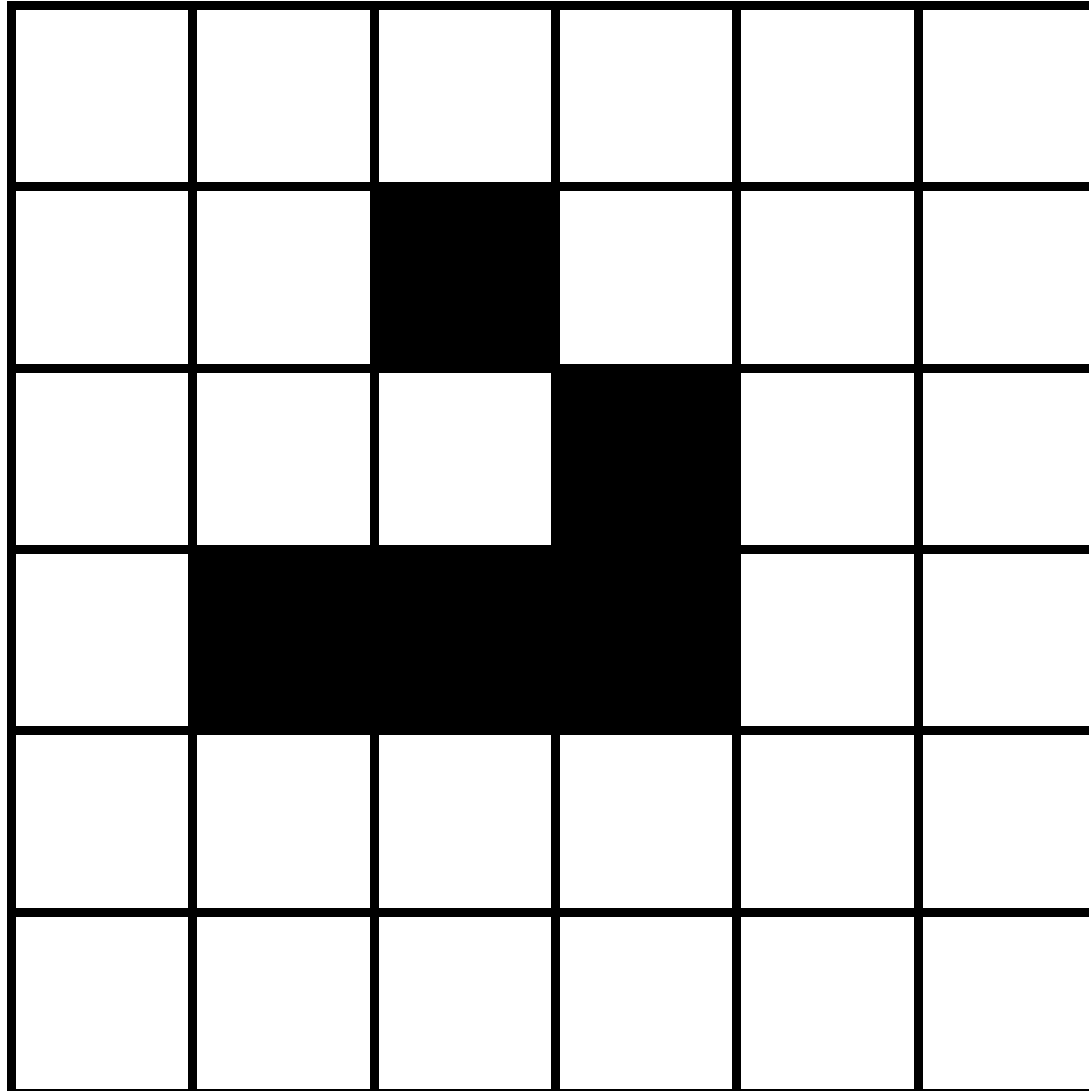
I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway

- Si une cellule a trois voisines vivantes (sans se compter elle même), elle est vivante à l'étape suivante.
- Si une cellule a exactement deux voisines vivantes, elle reste dans son état à l'étape suivante.
- Sinon, elle est morte à l'étape suivante.

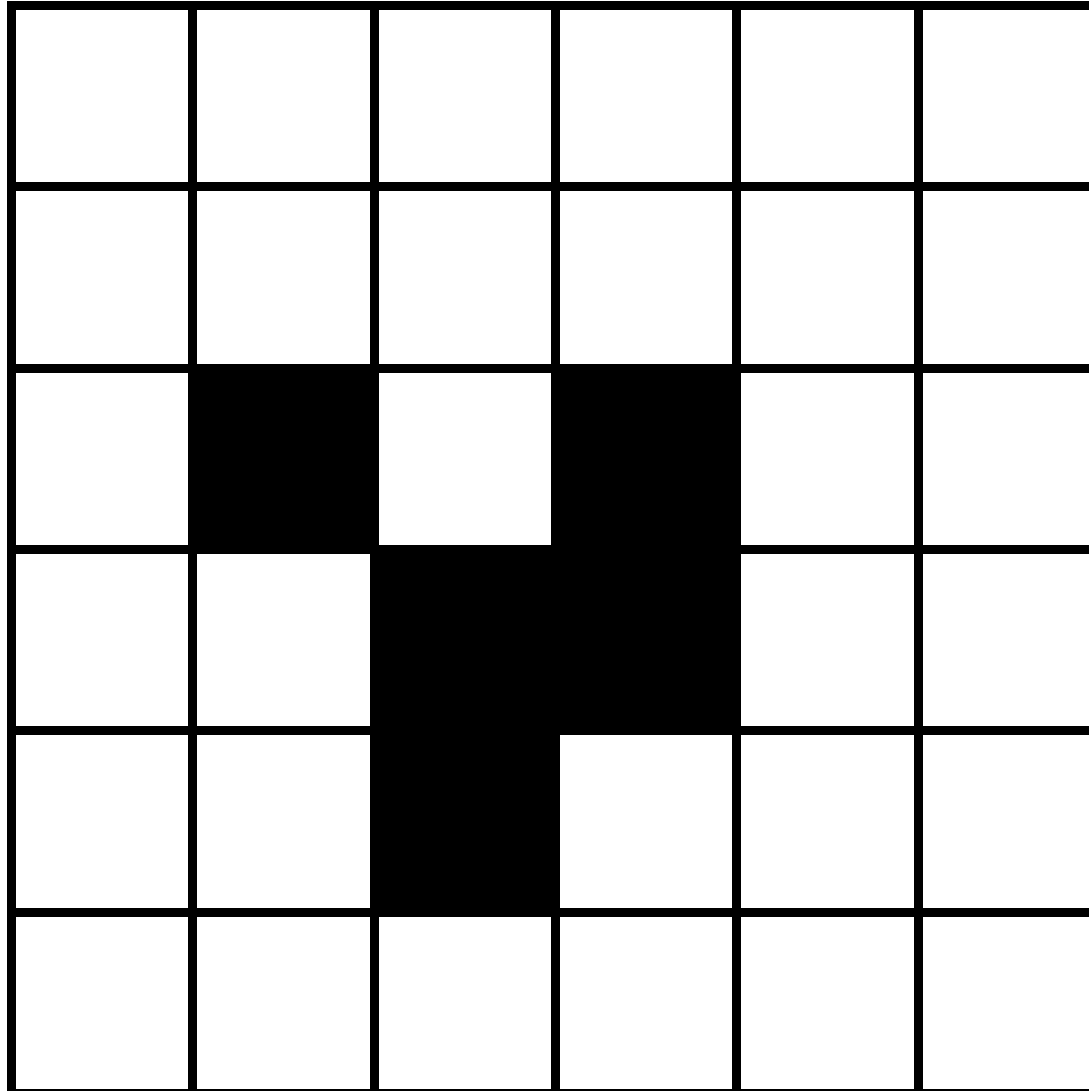
I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway



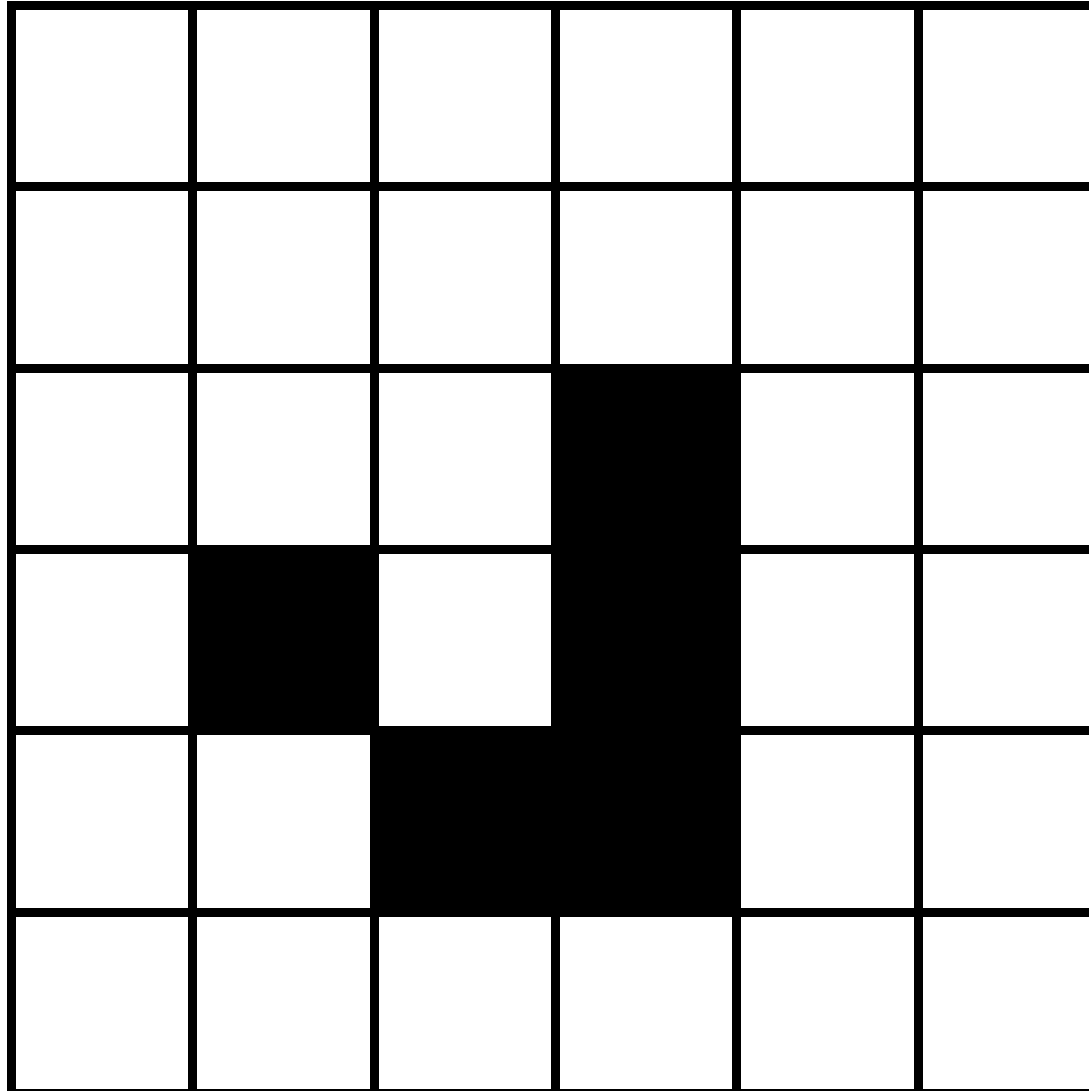
I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway



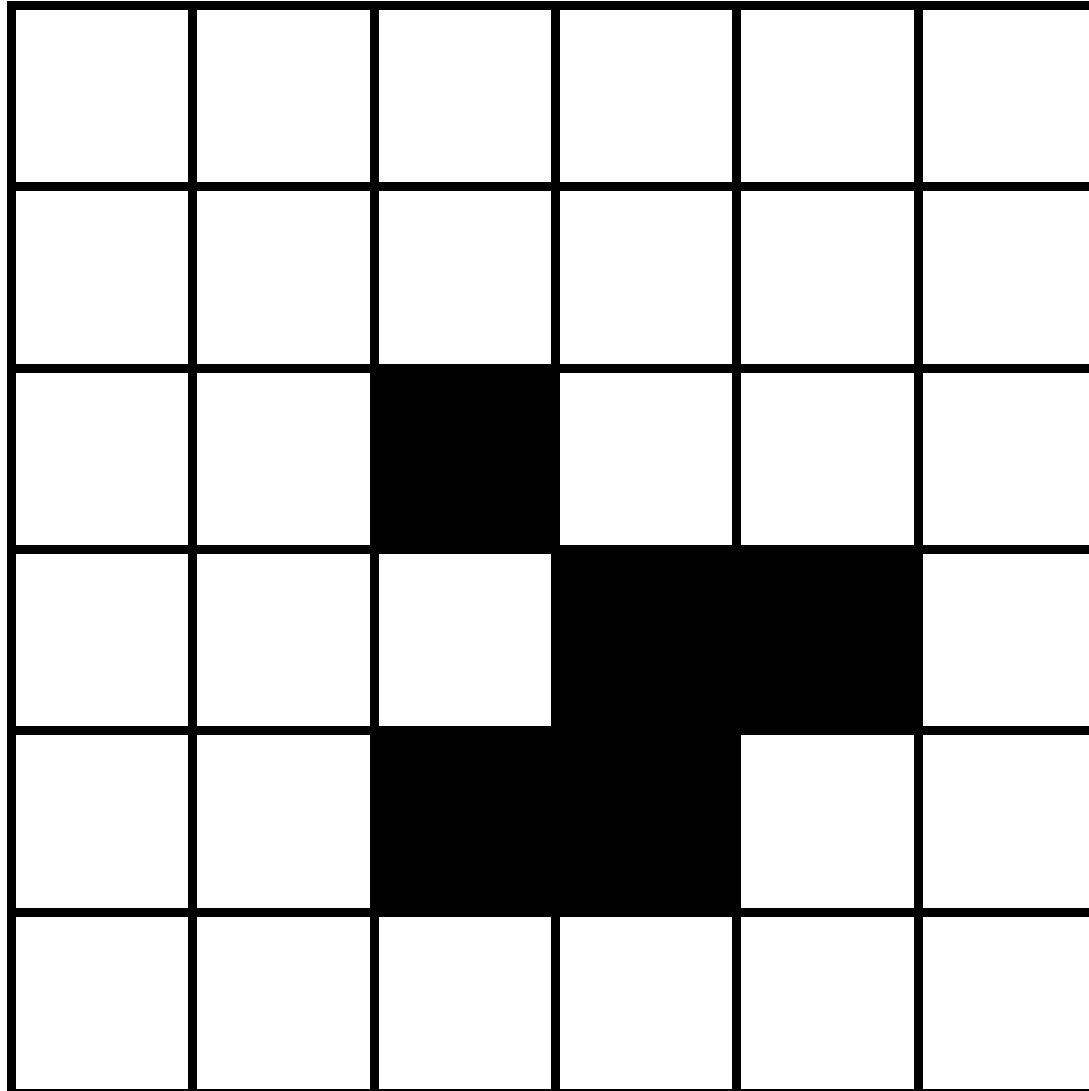
I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway



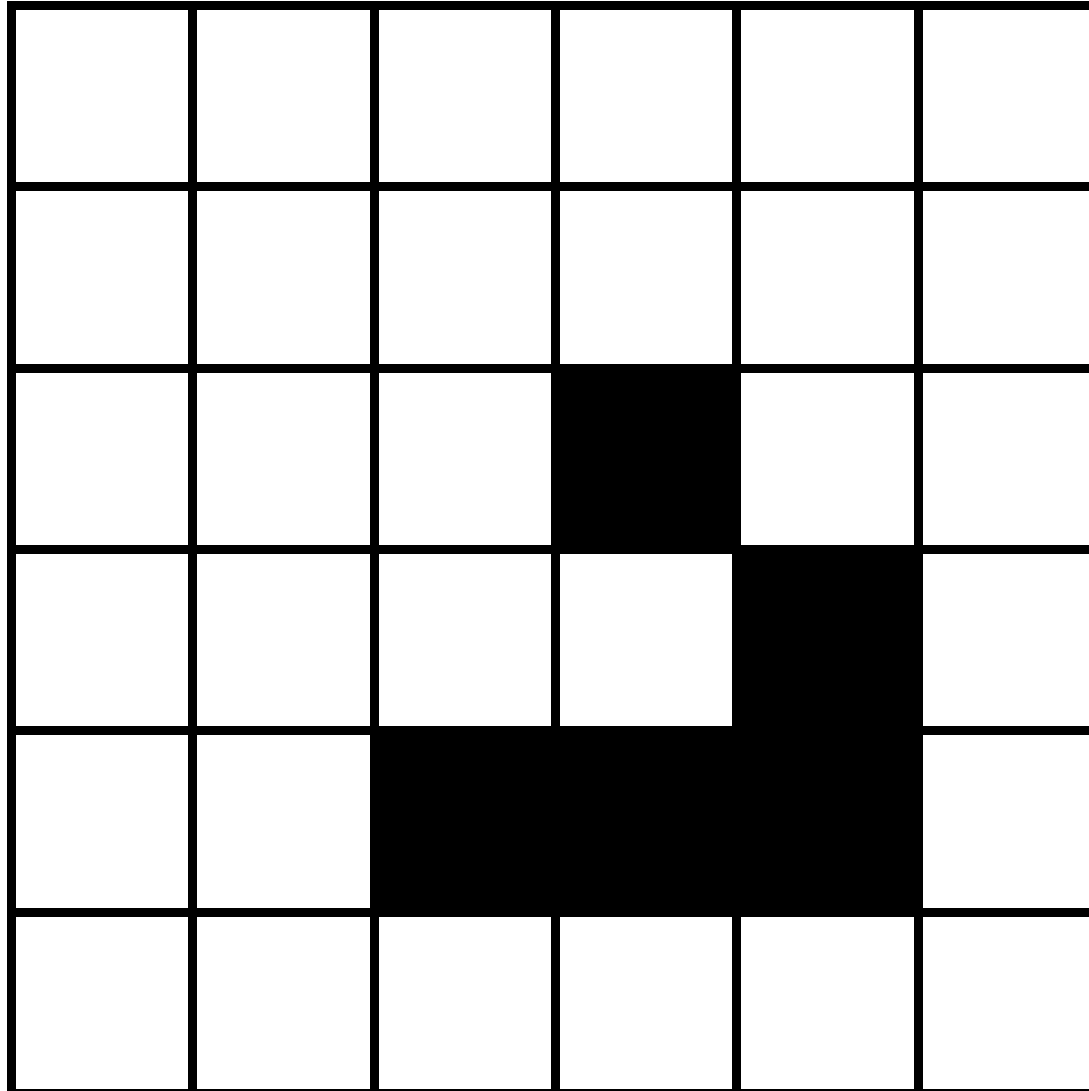
I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway



I. Présentation du modèle choisi

1. Règles du jeu de la vie de Conway



I. Présentation du modèle choisi

2. Représentation discrète en base 2

Chaque couple (Amplitude, fréquence)

$$e(t) = \sum_{n=1}^{+\infty} A_{n-1} \sin(2\pi n f_0 t + \varphi_{n-1})$$



$$s(t) = \sum_{n=1}^{+\infty} G(A_{n-1}) \sin(2\pi n f_0 t + \varphi_{n-1})$$

I. Présentation du modèle choisi

2. Représentation discrète en base 2

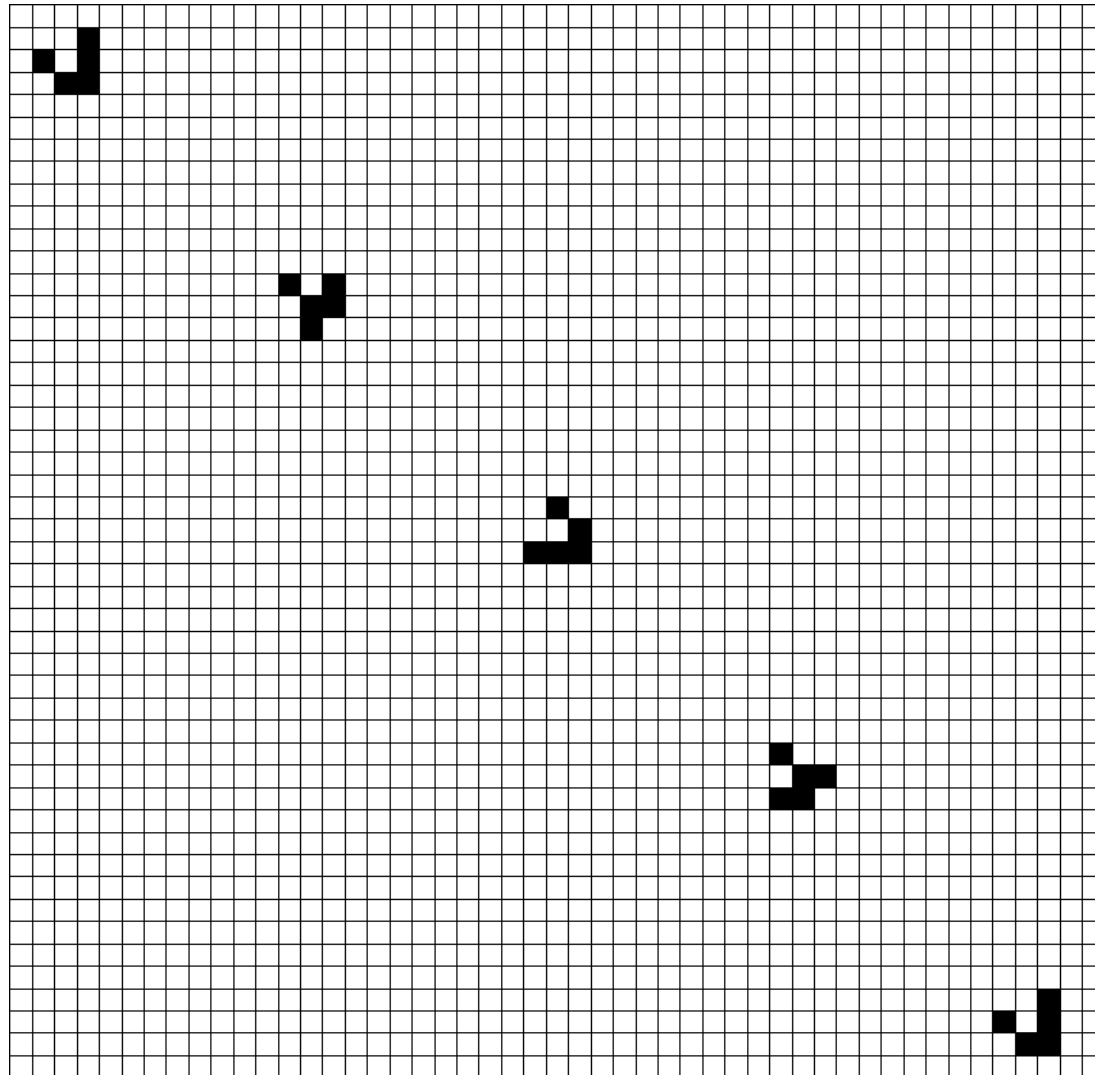
$$G(A_n) = \frac{A_n}{2^{f(n)}}$$

Filtre passe bas : $G(A_n) = \frac{A_n}{2^n}$

I. Présentation du modèle choisi

3. Faisceaux de gliders

p43



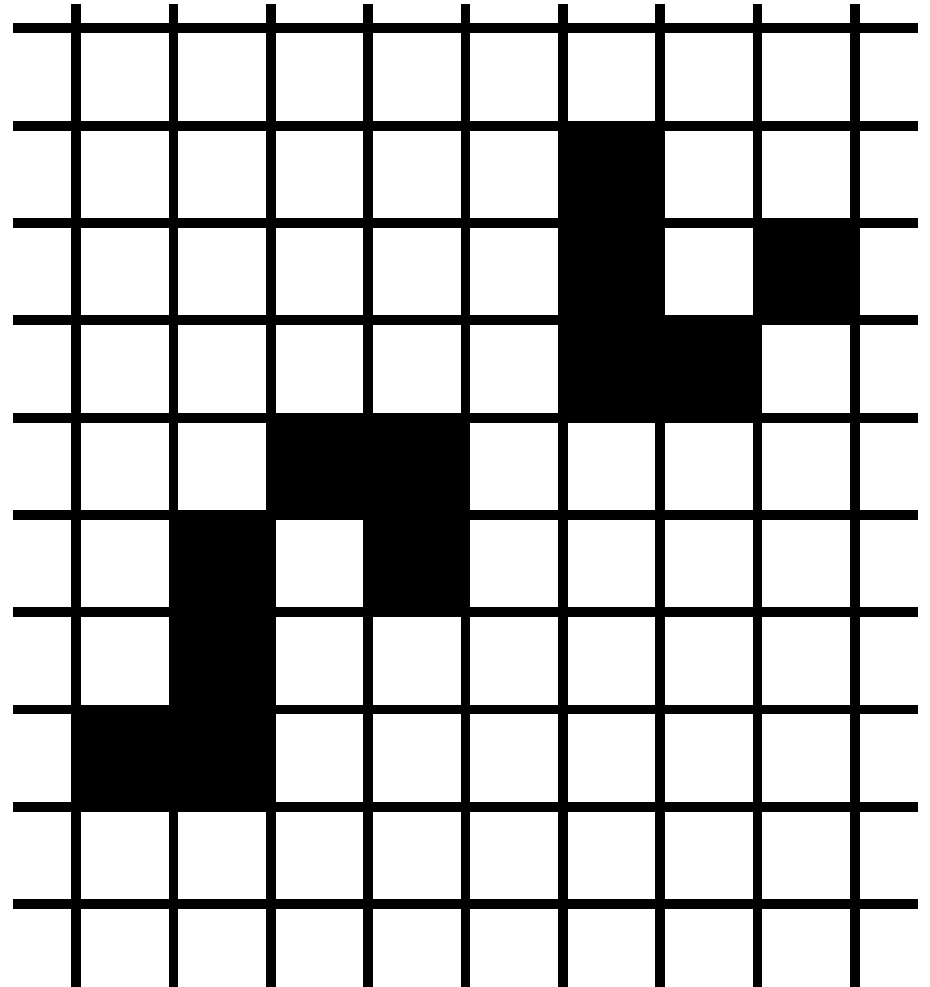
II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

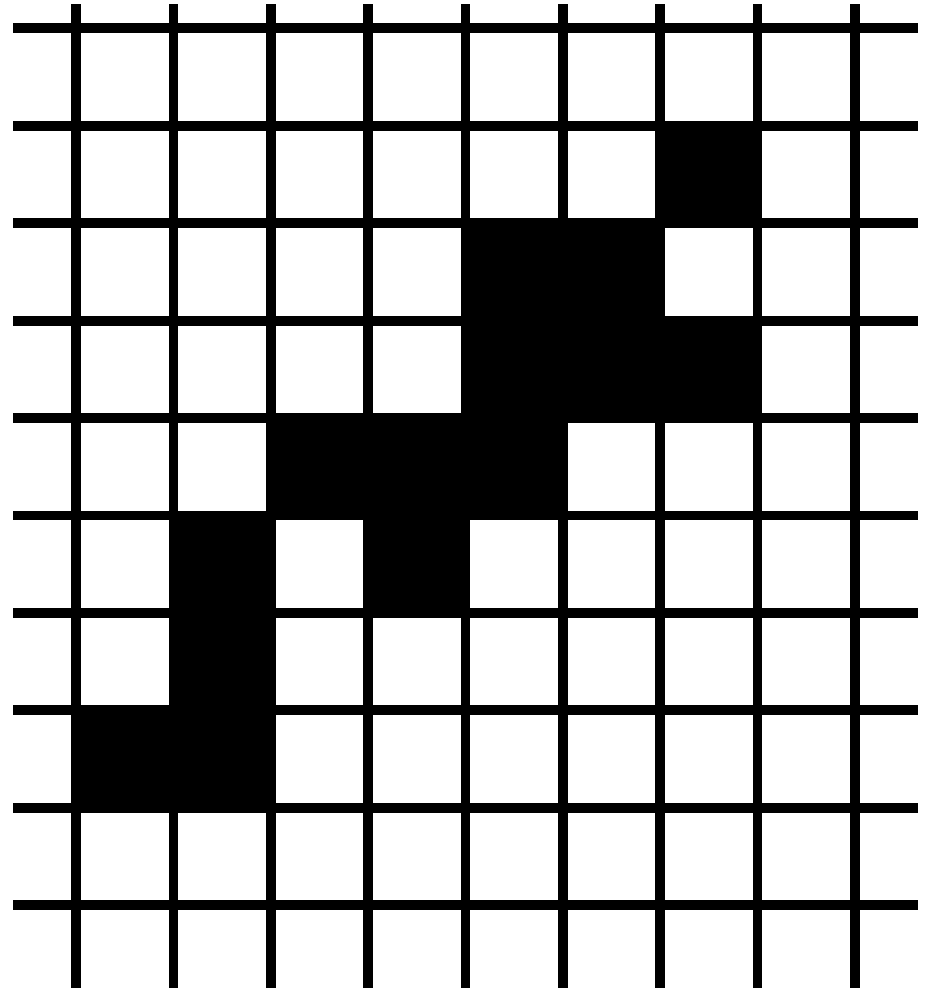
Glider Stopper :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

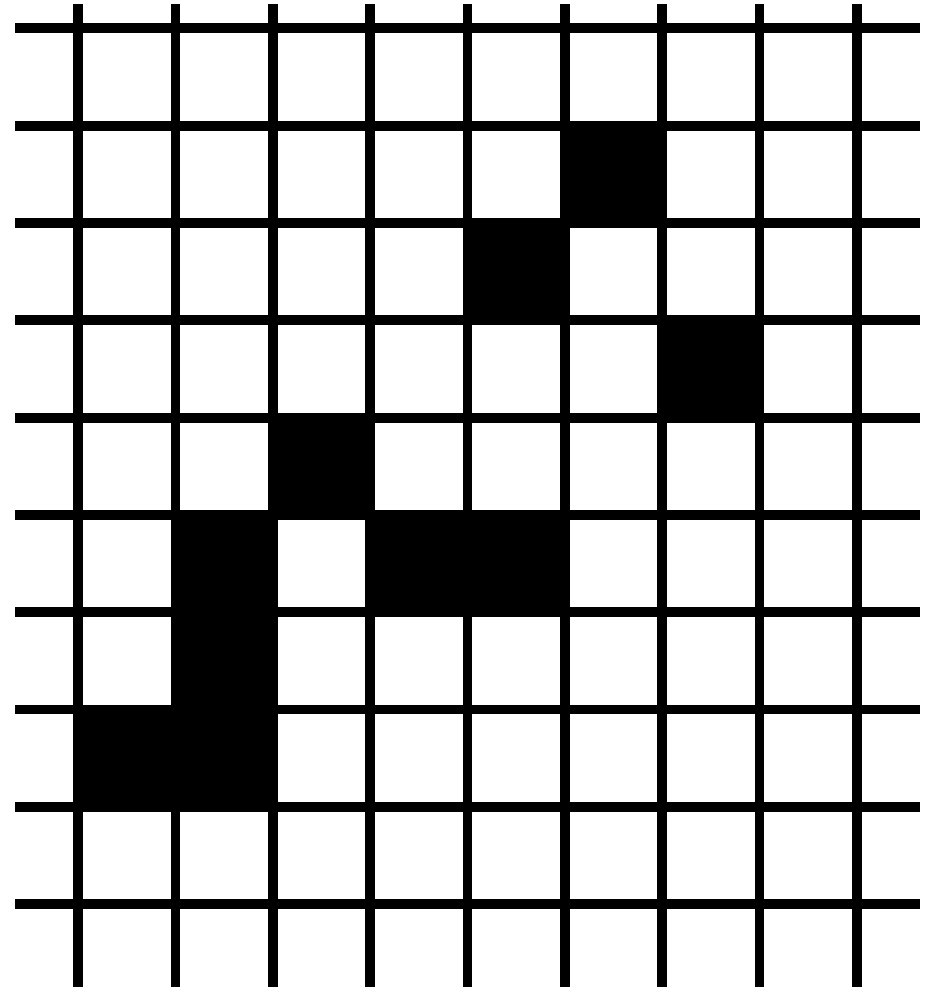
Glider Stopper :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

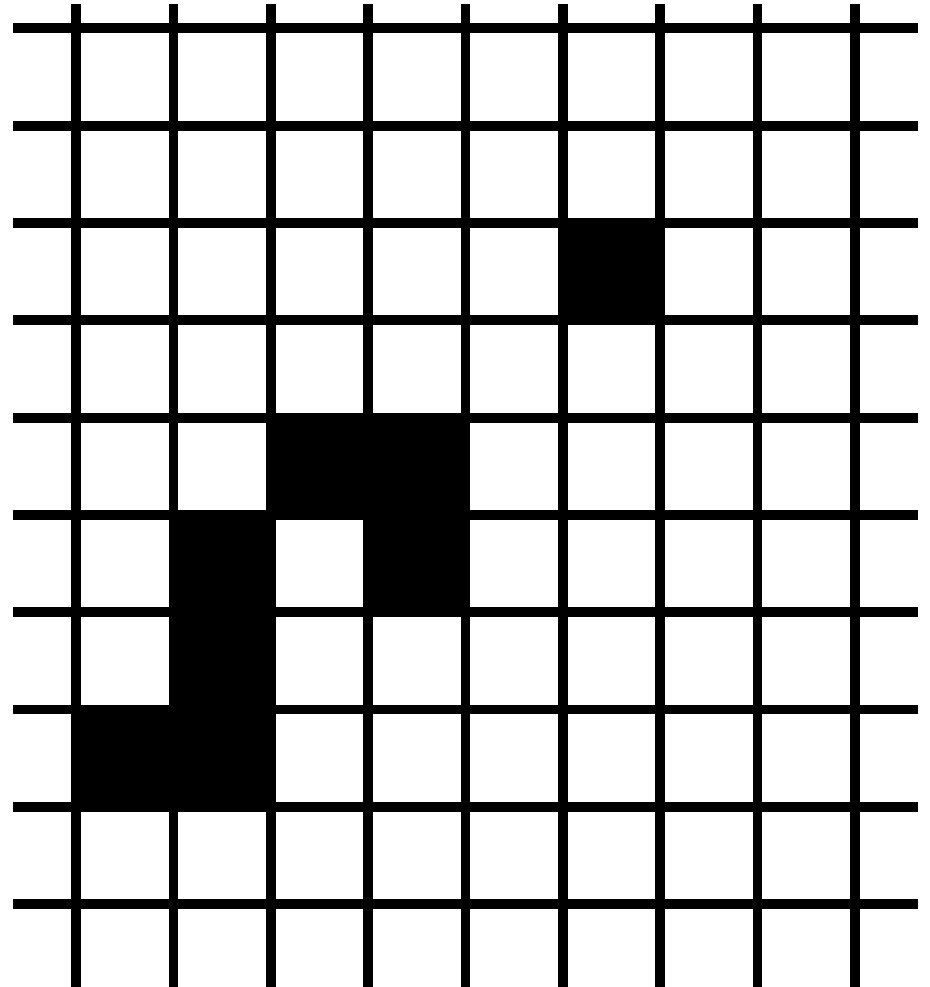
Glider Stopper :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

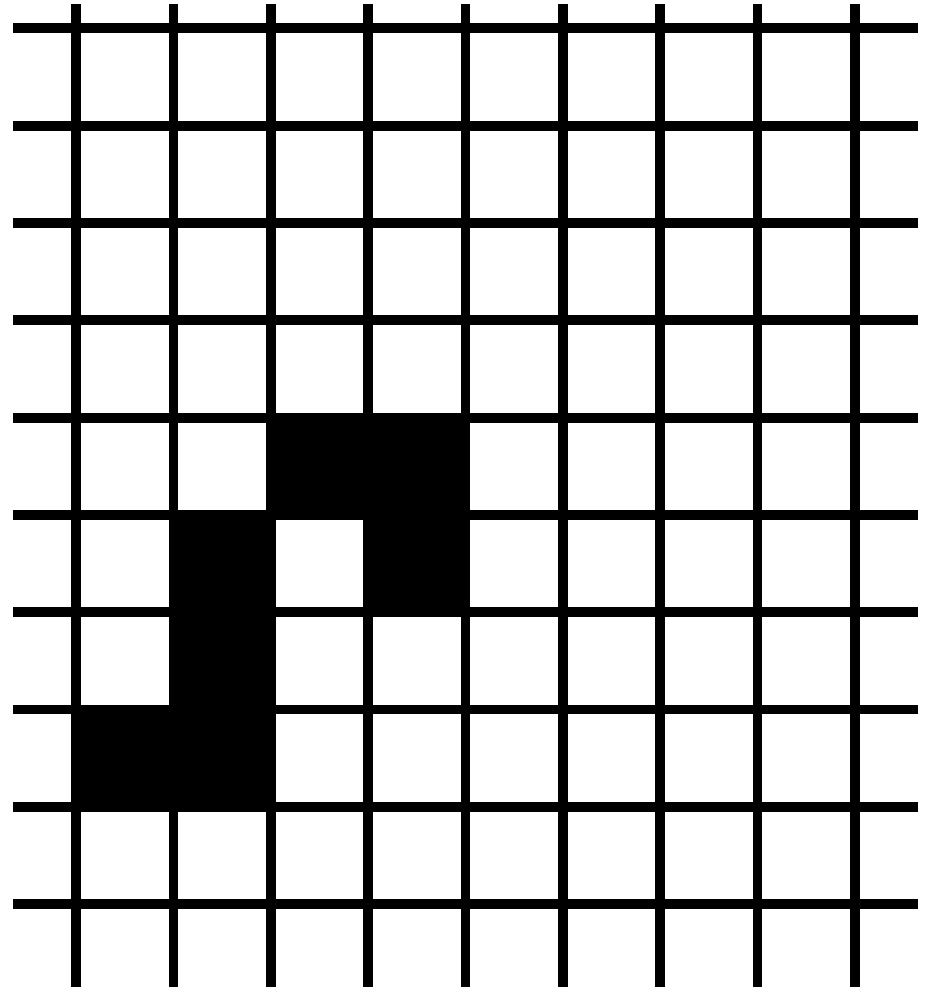
Glider Stopper :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

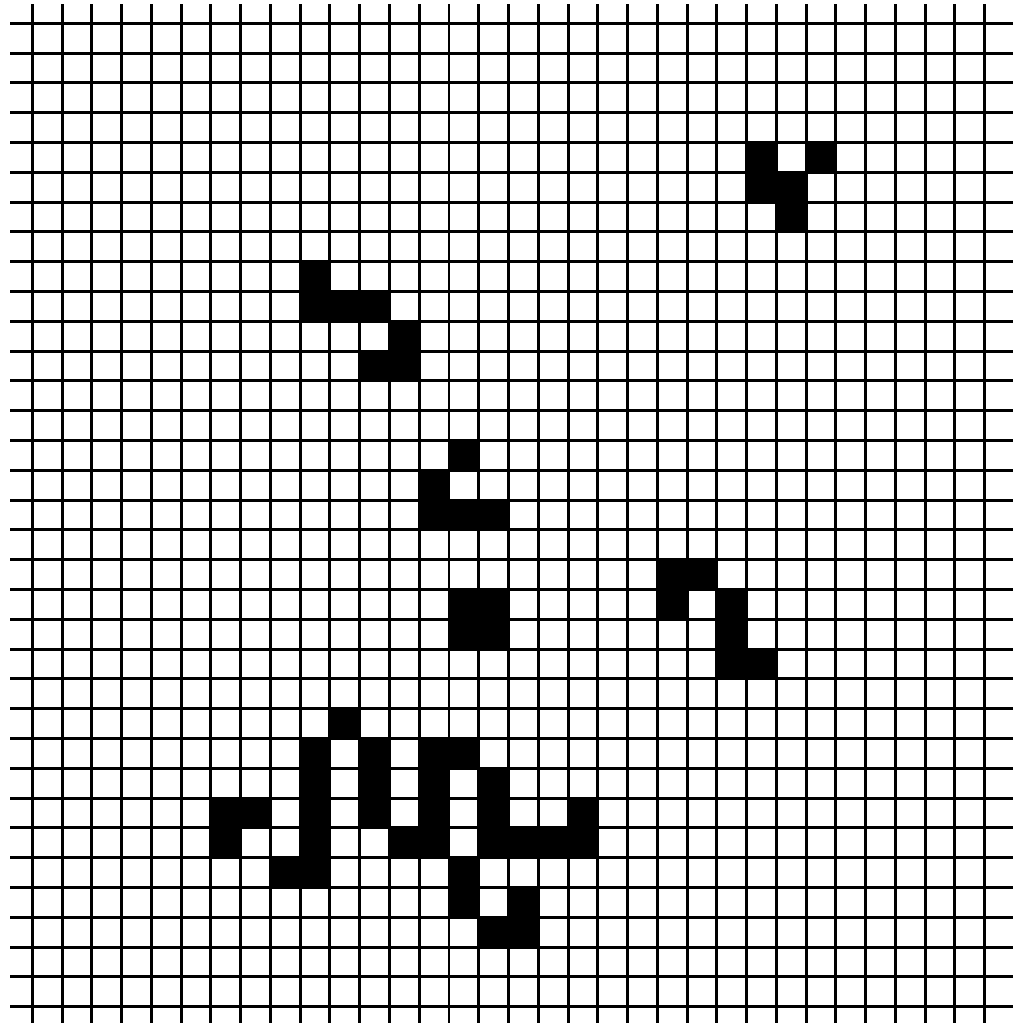
Glider Stopper :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

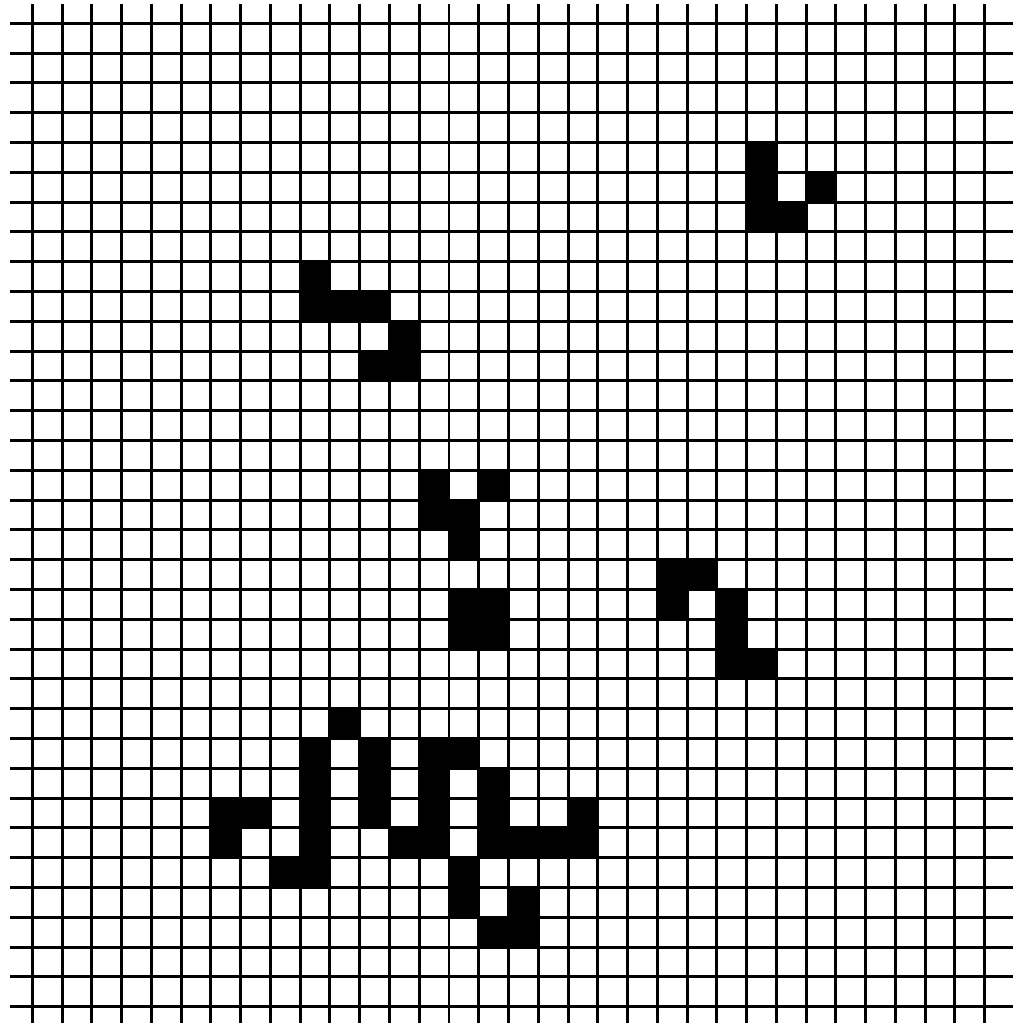
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

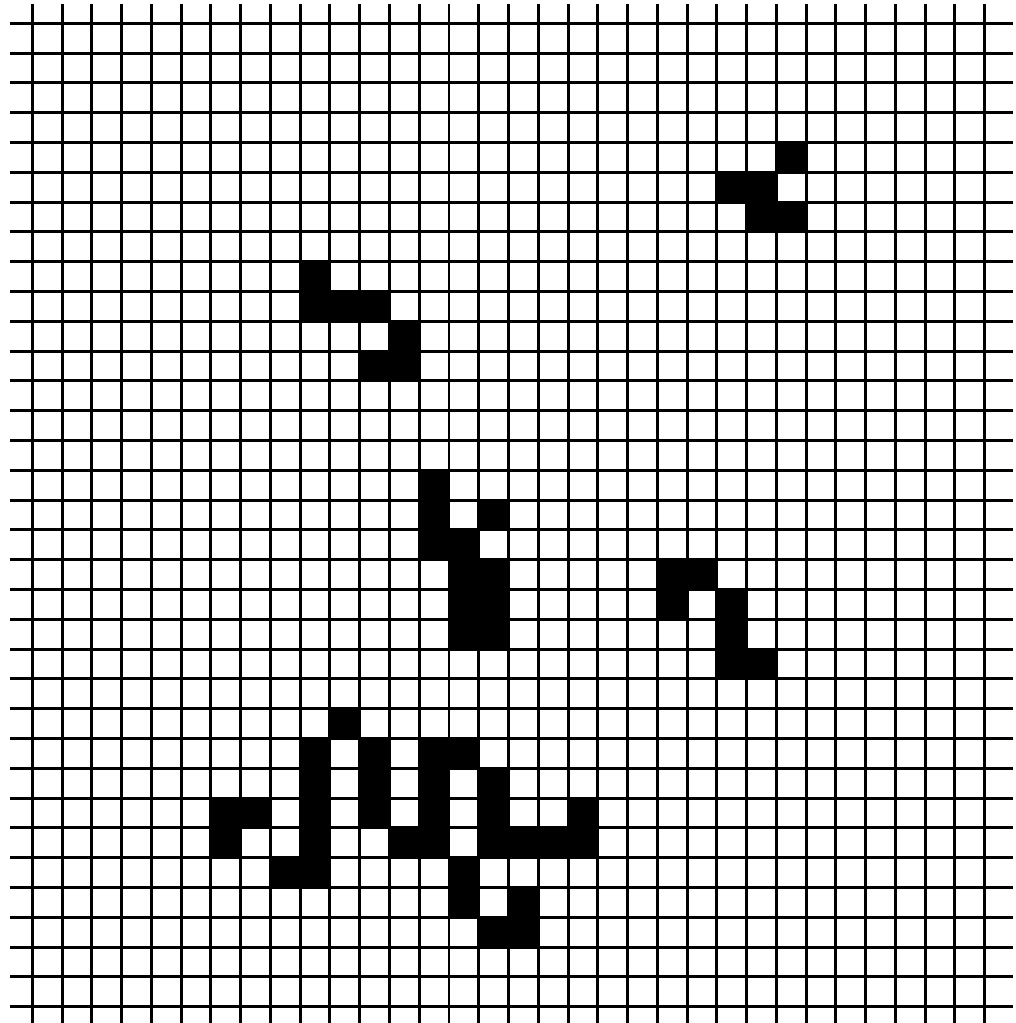
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

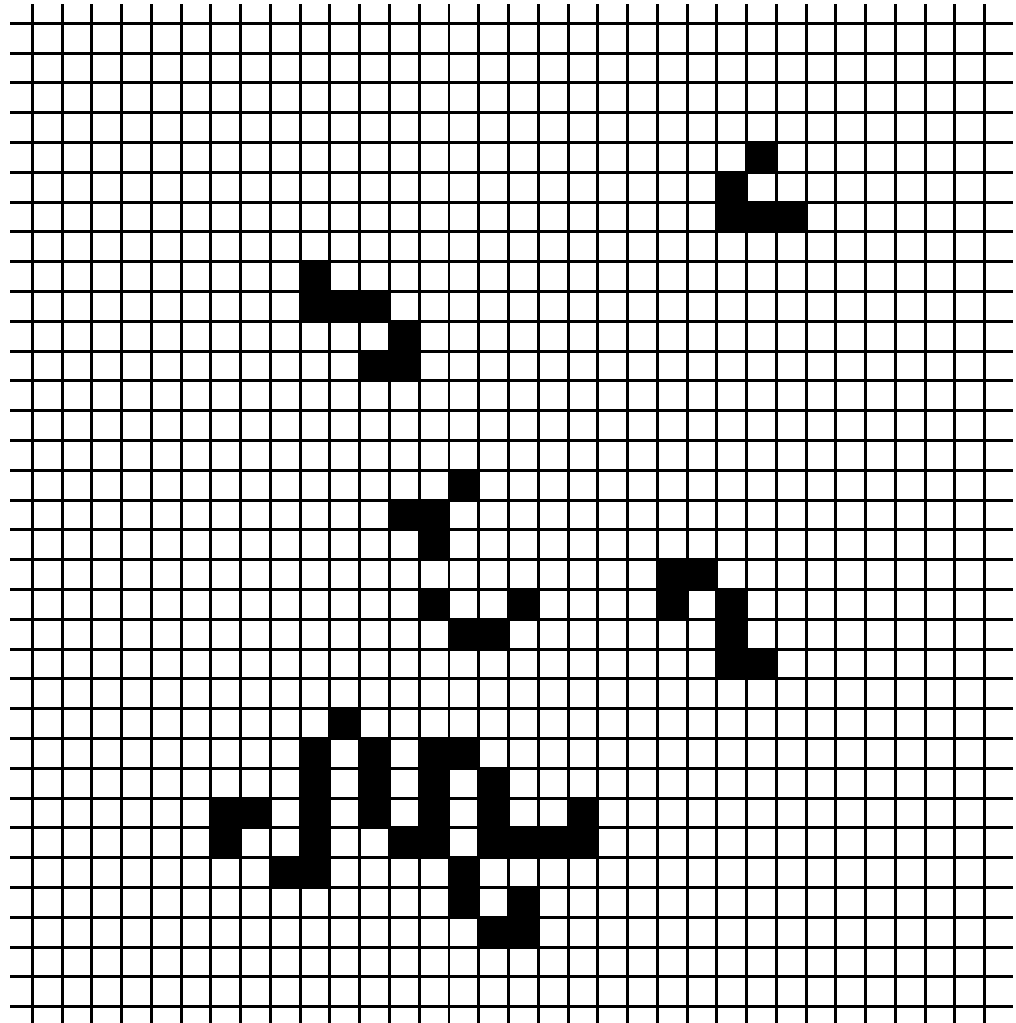
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

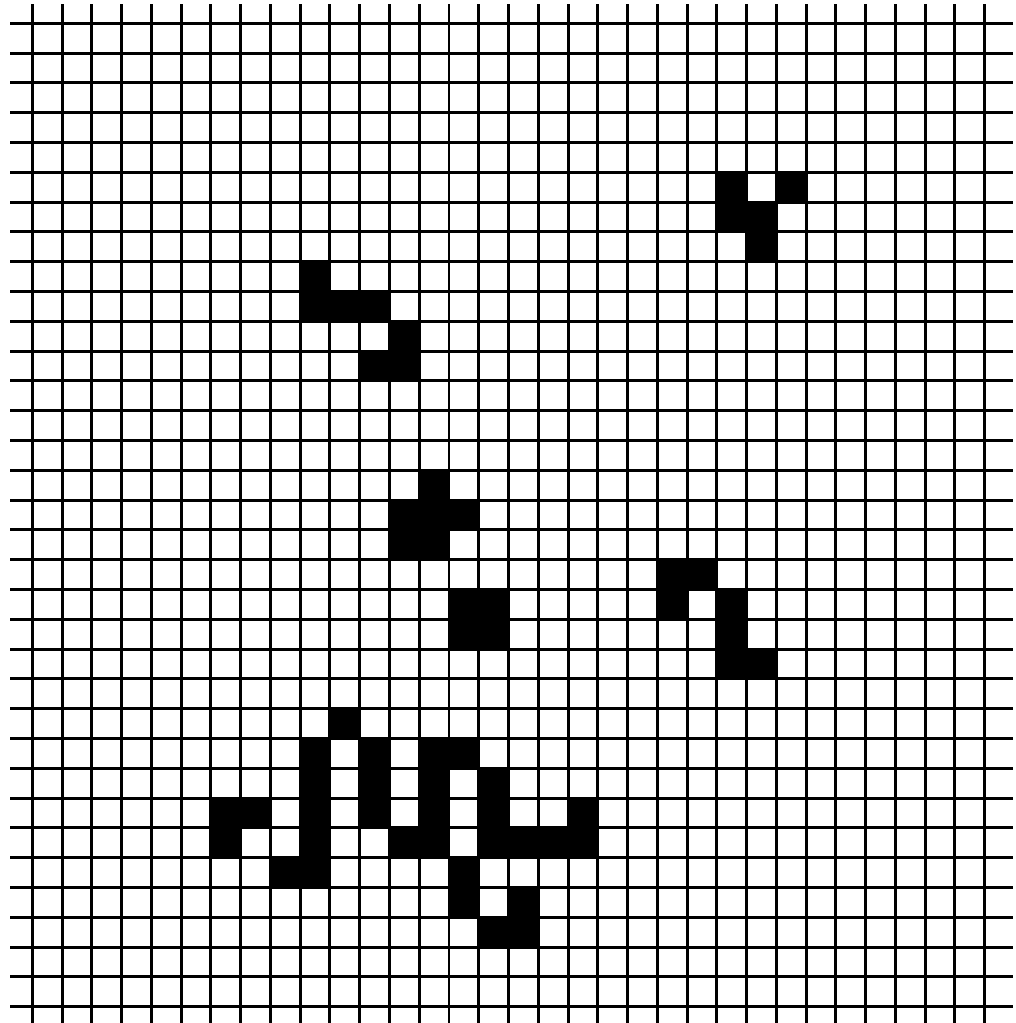
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

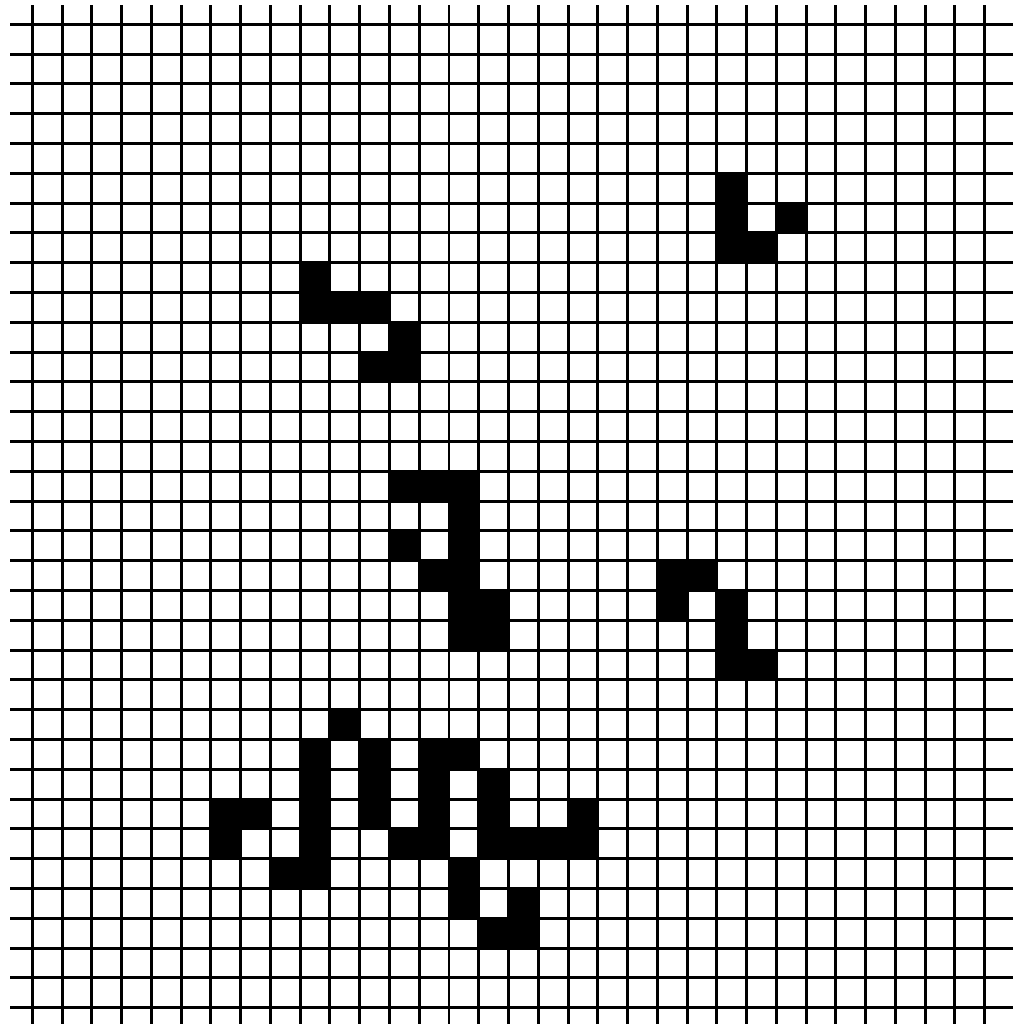
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

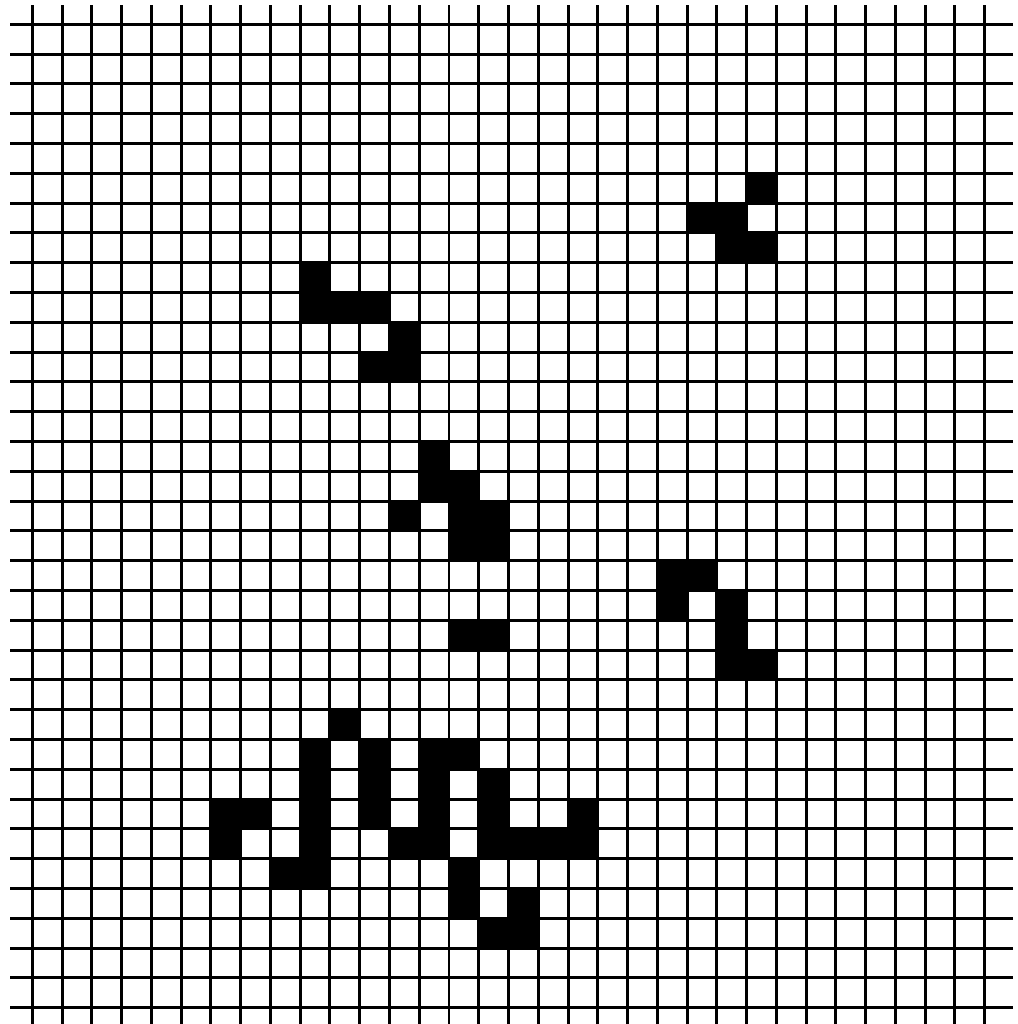
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

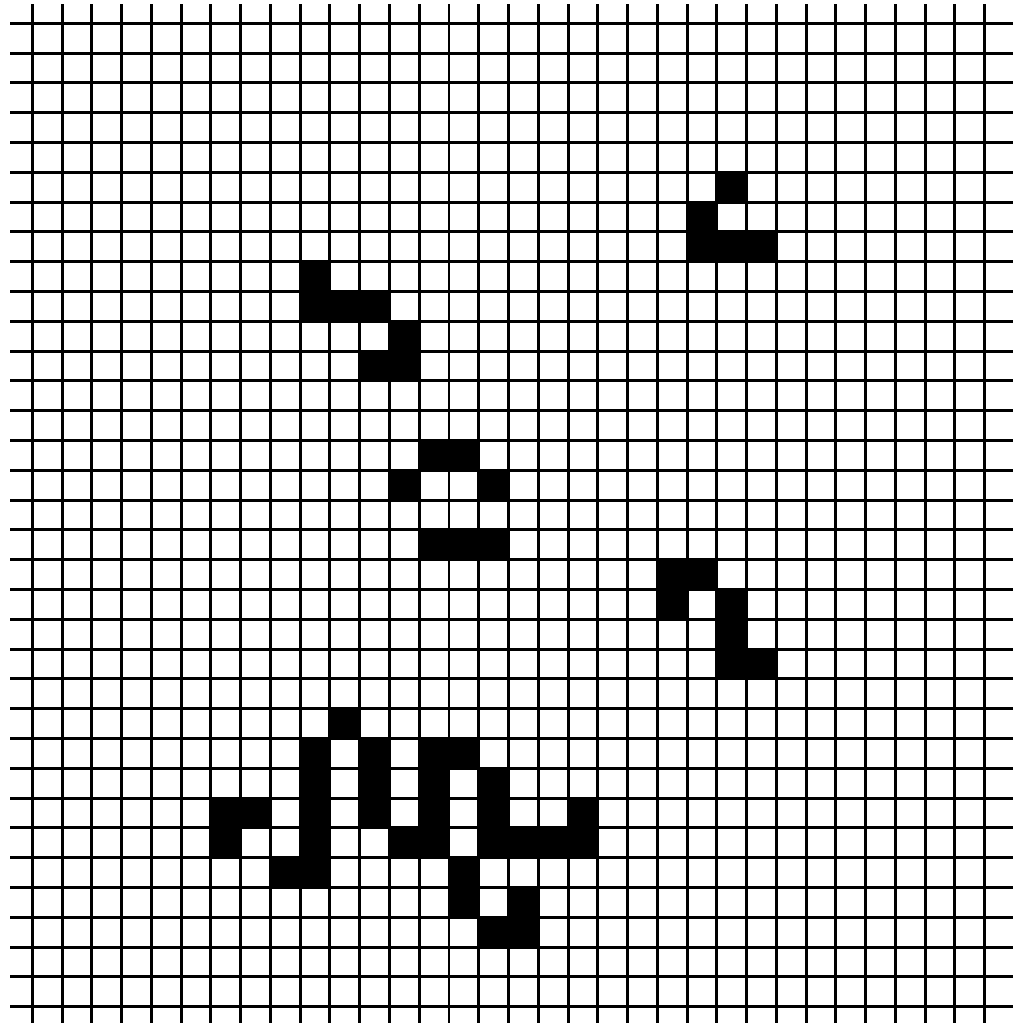
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

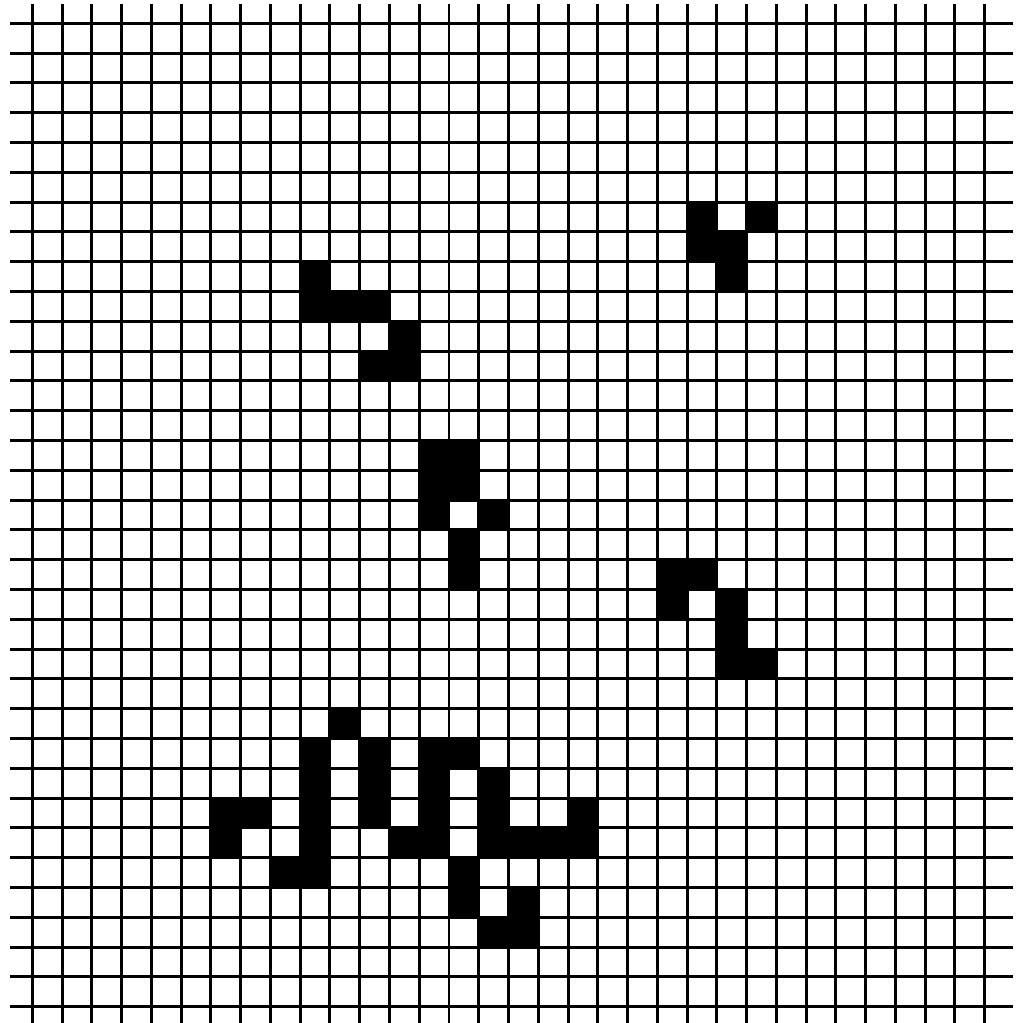
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

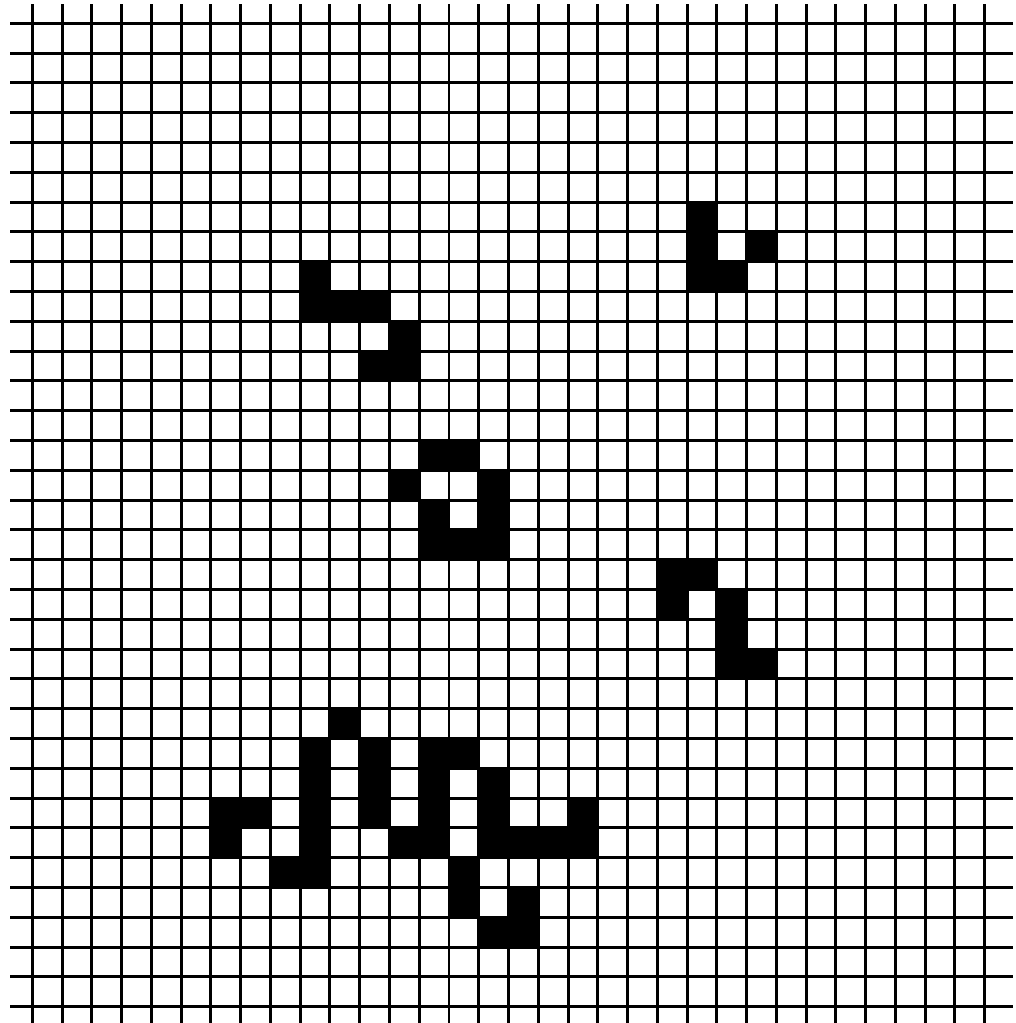
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

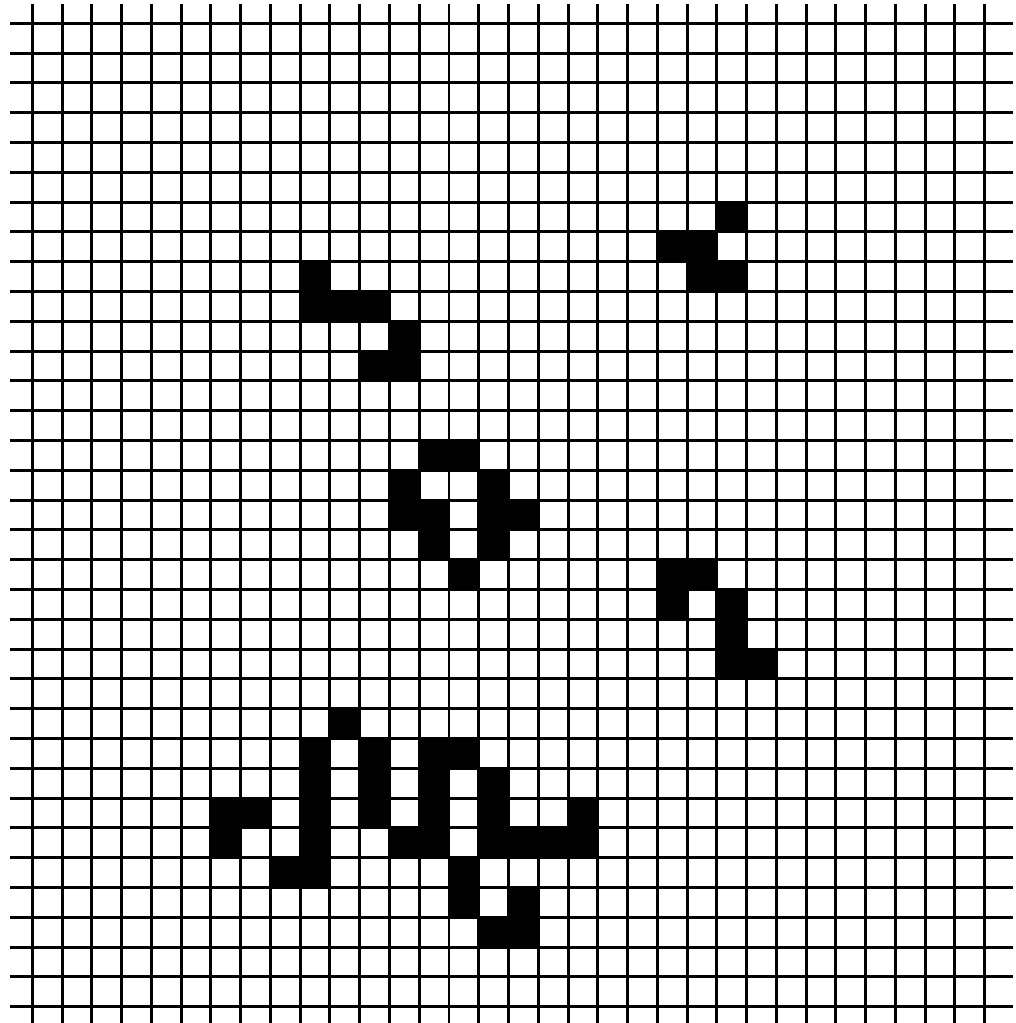
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

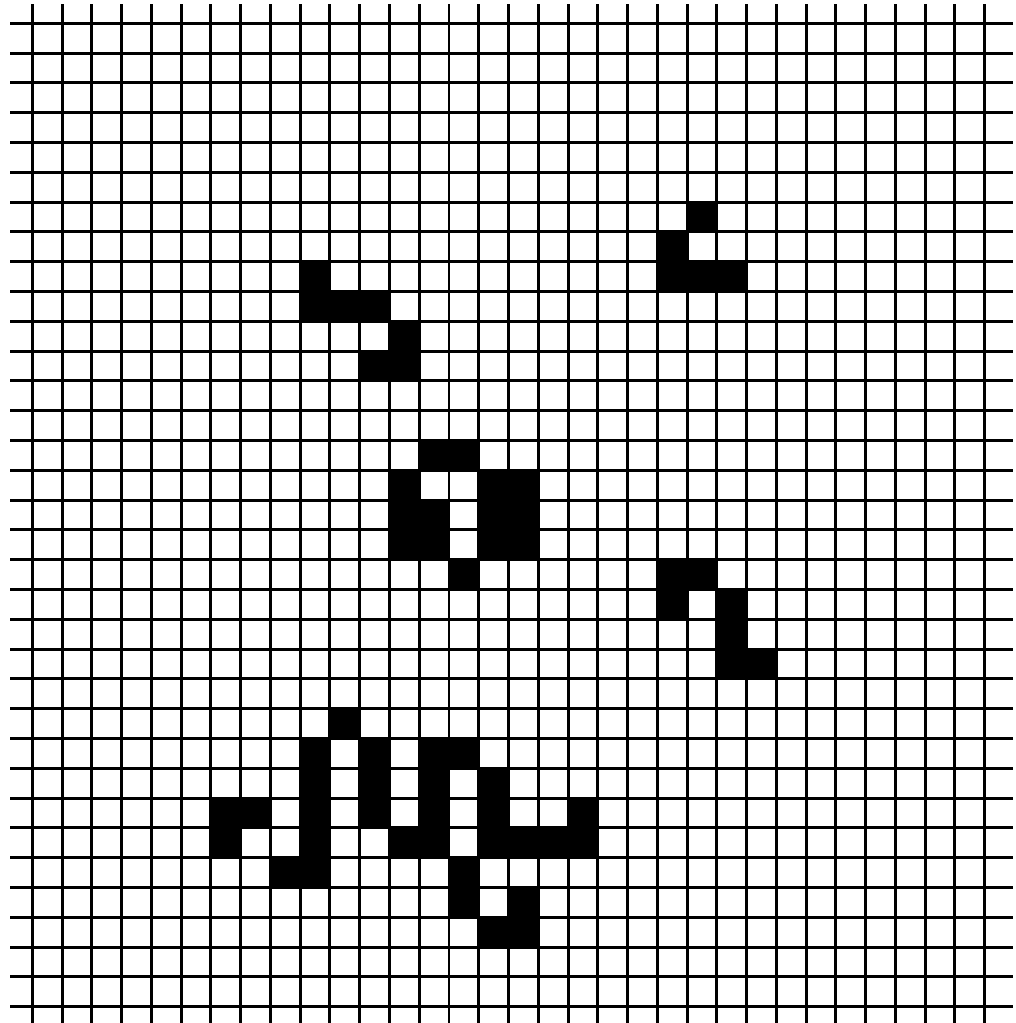
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

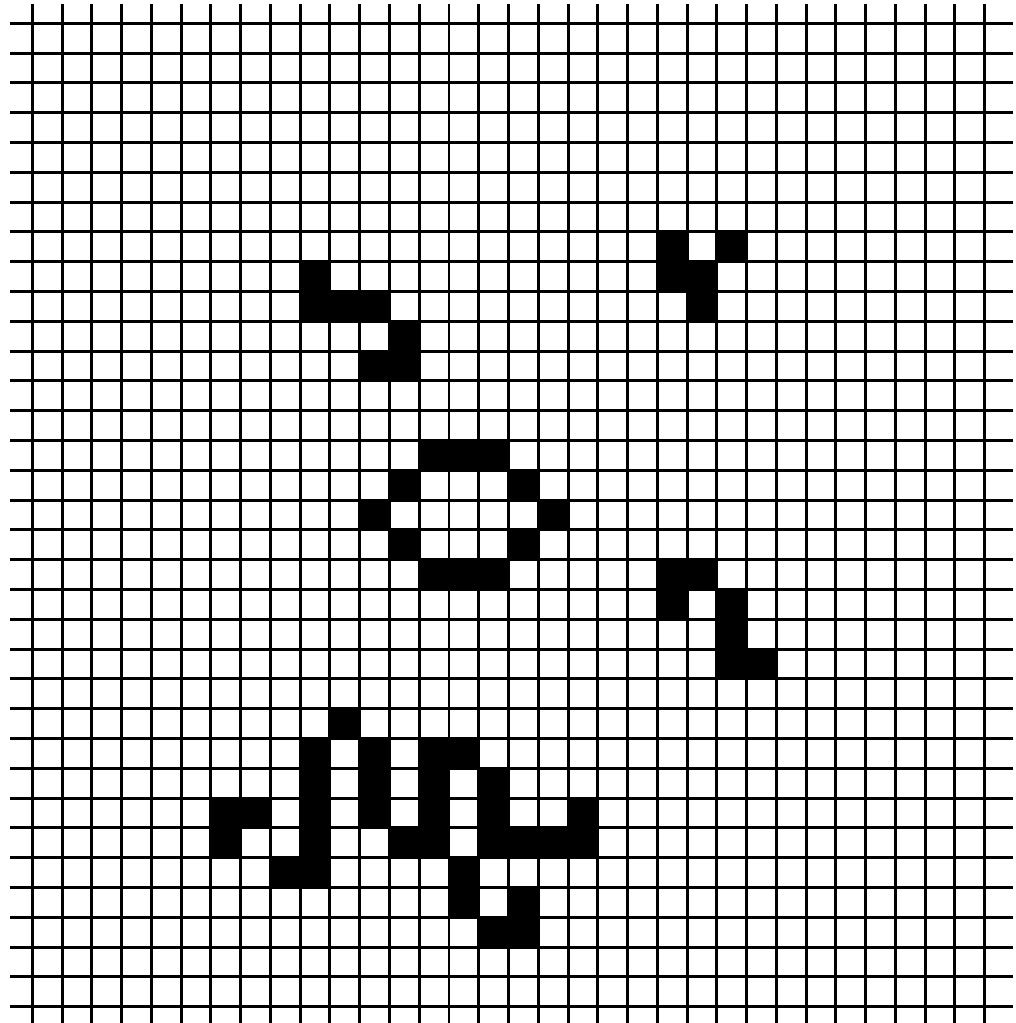
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

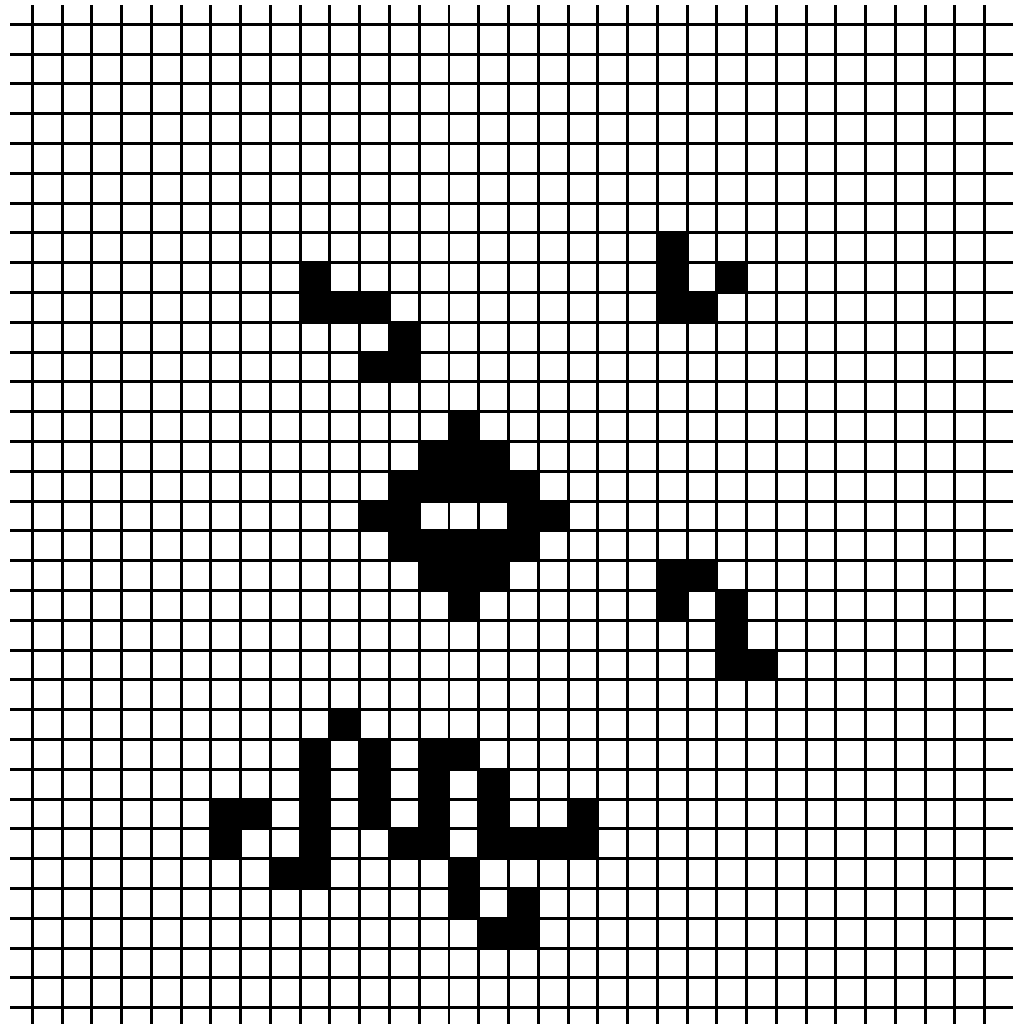
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

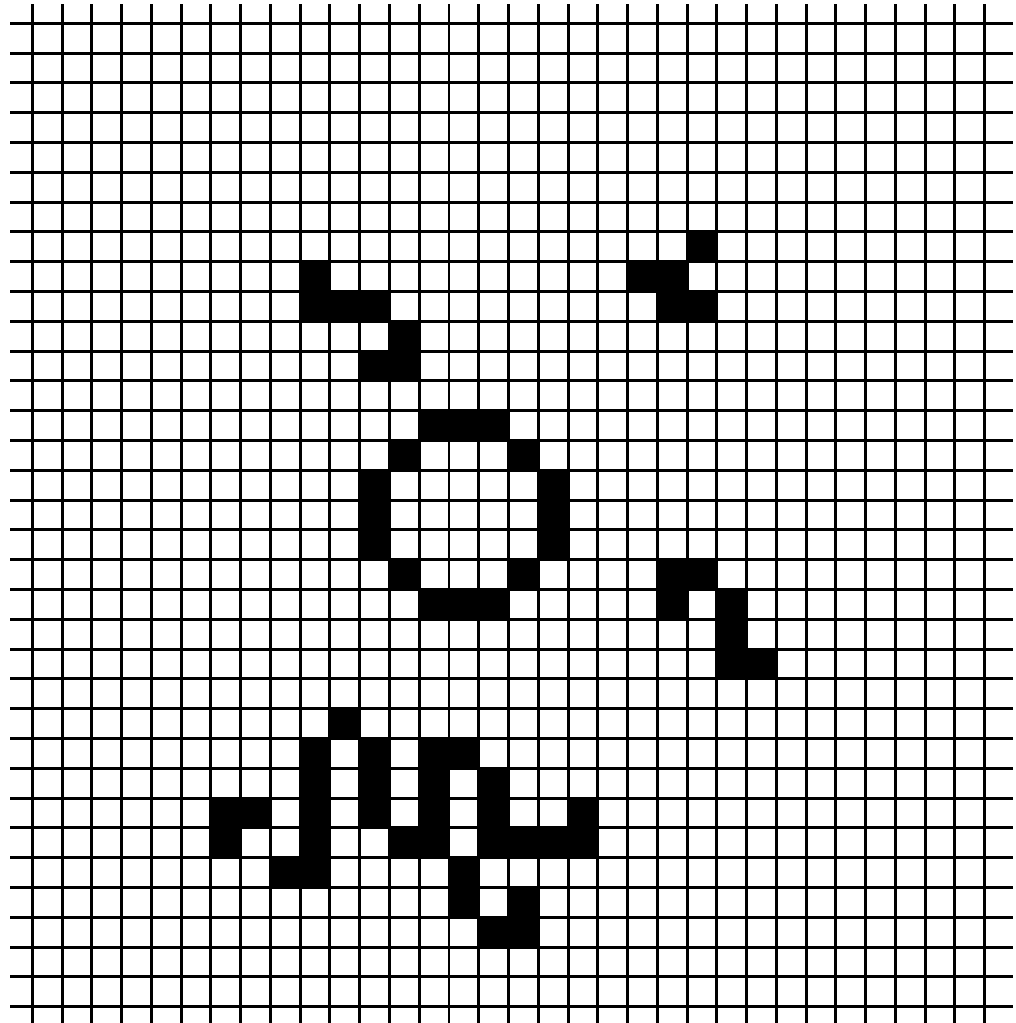
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

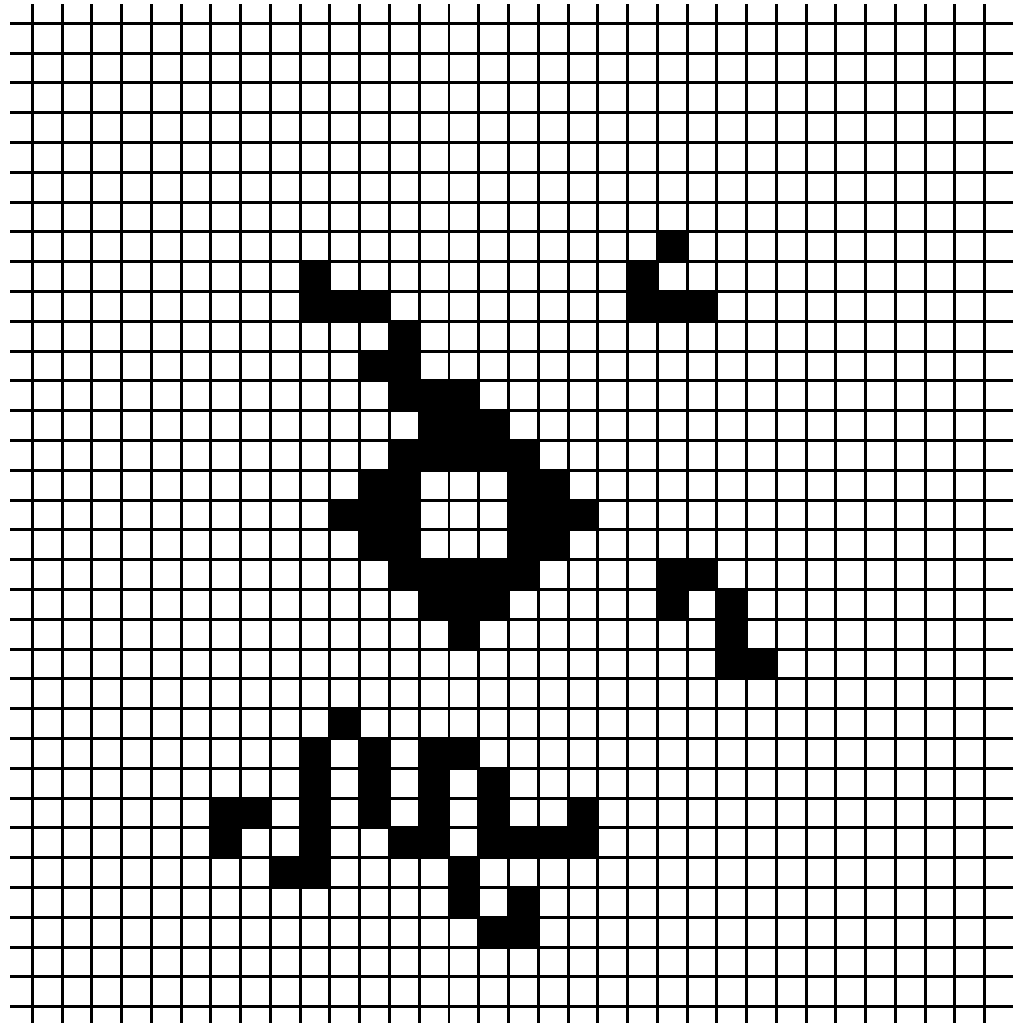
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

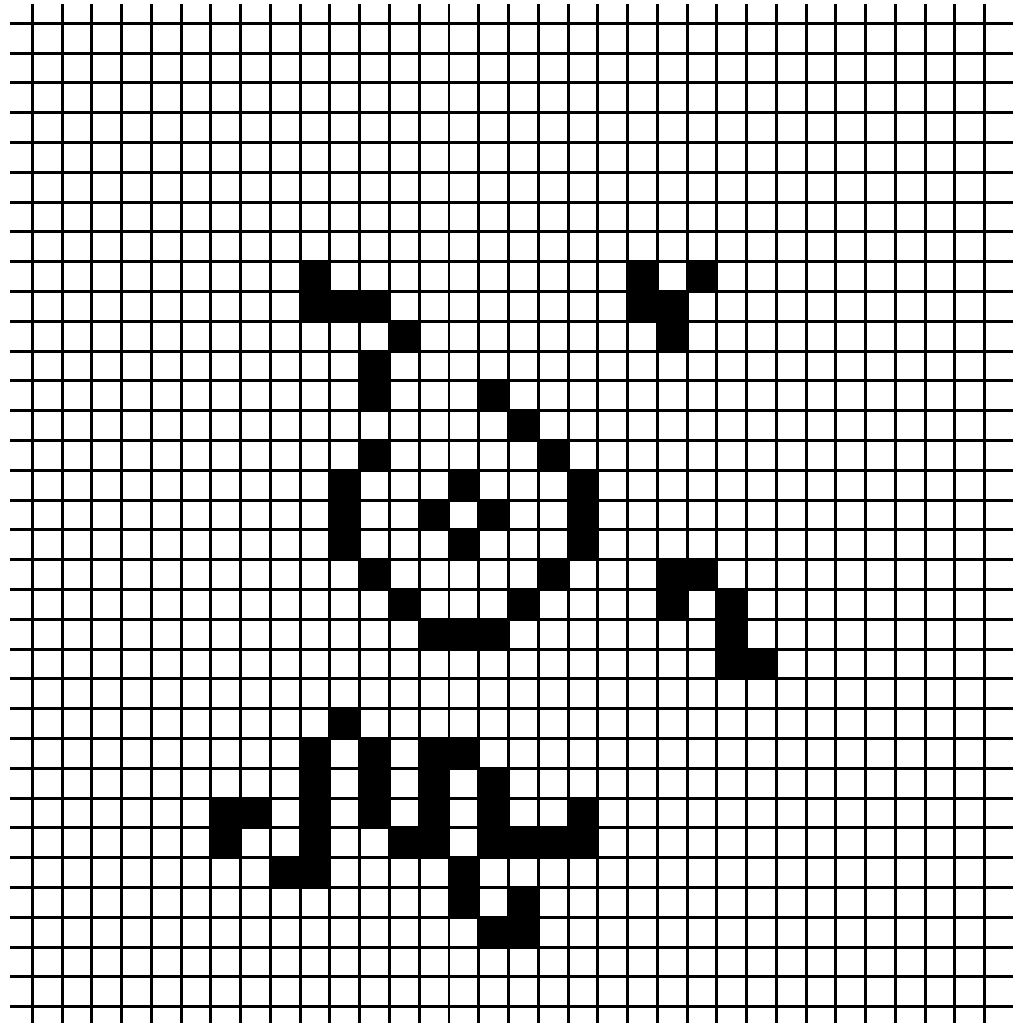
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

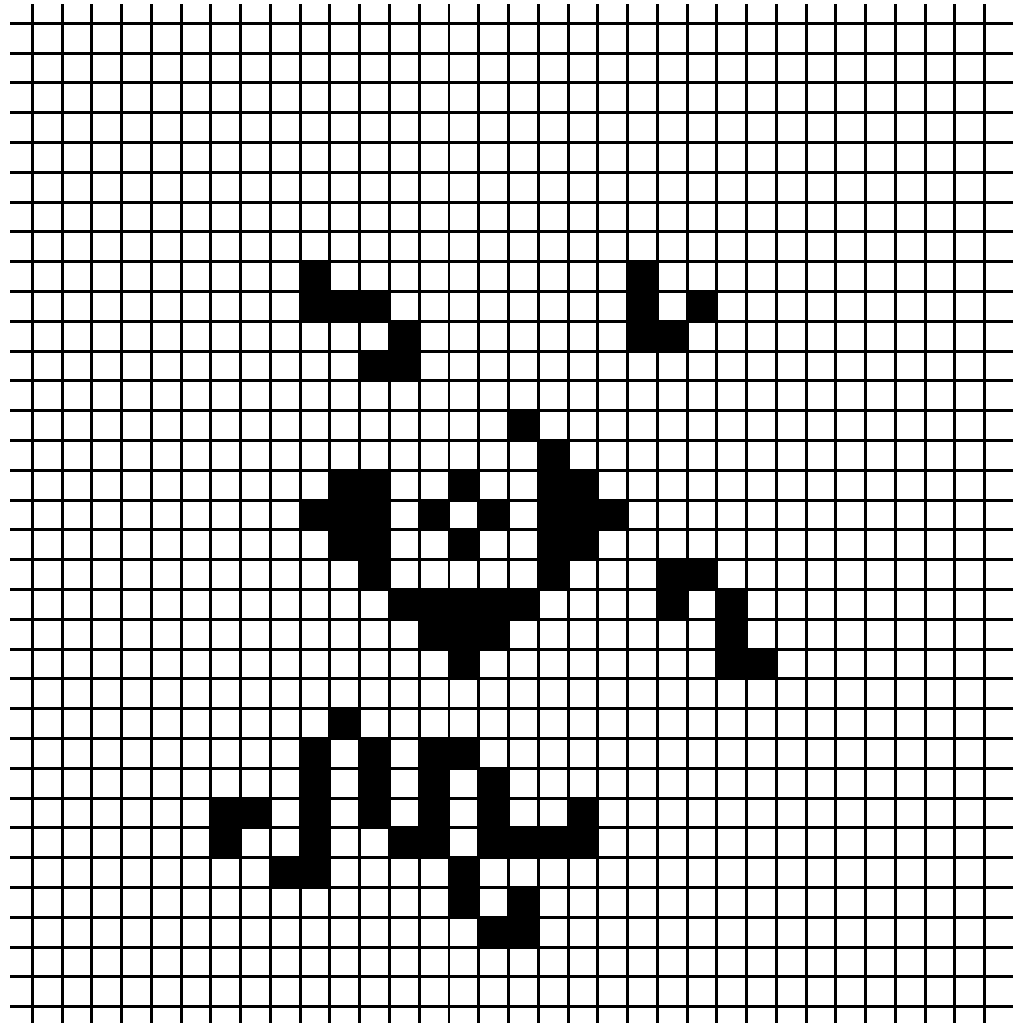
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

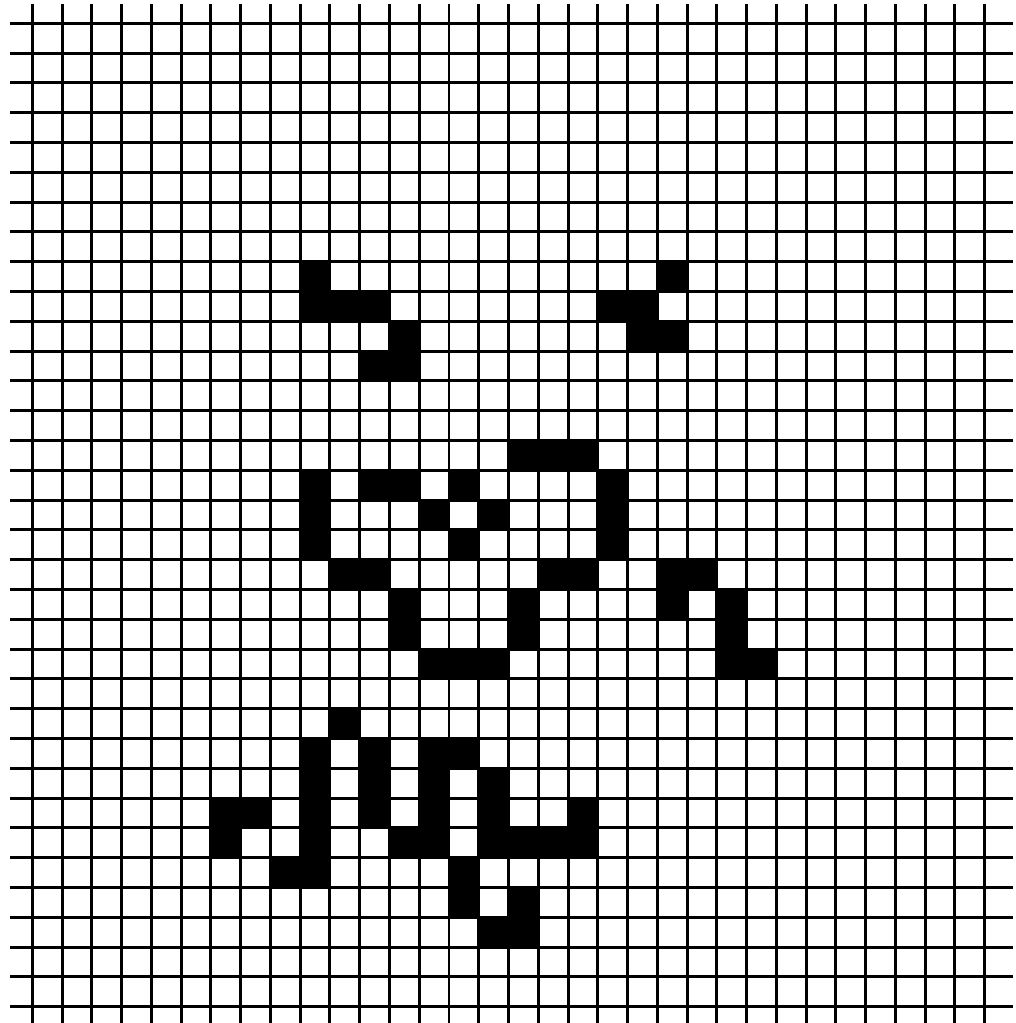
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

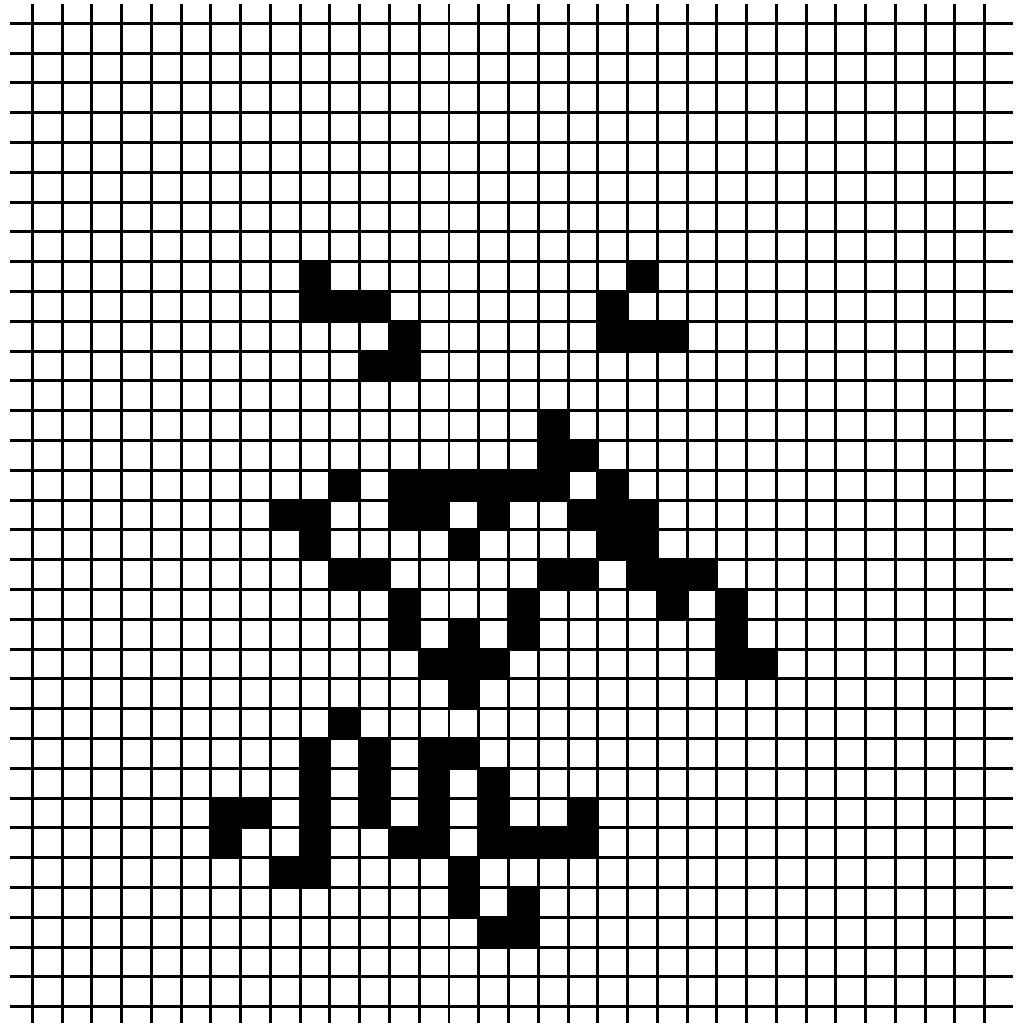
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

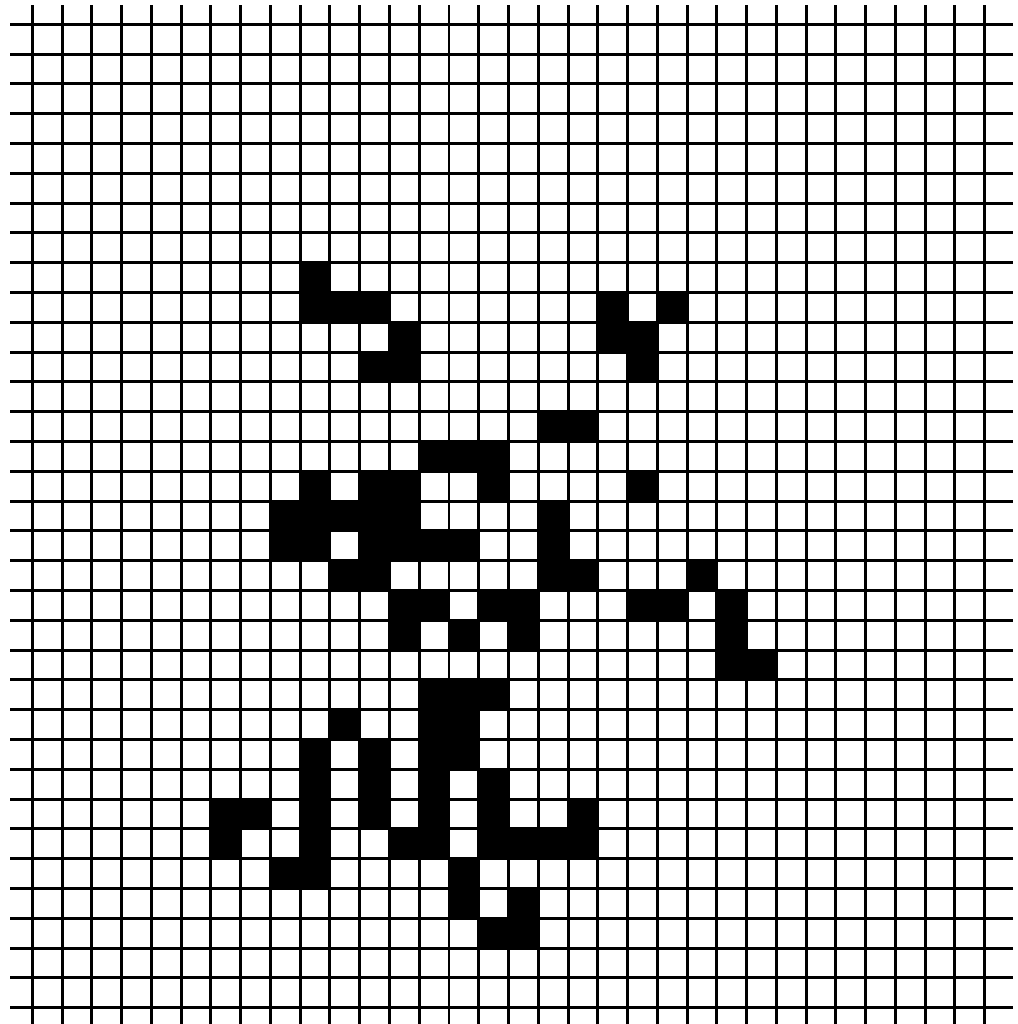
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

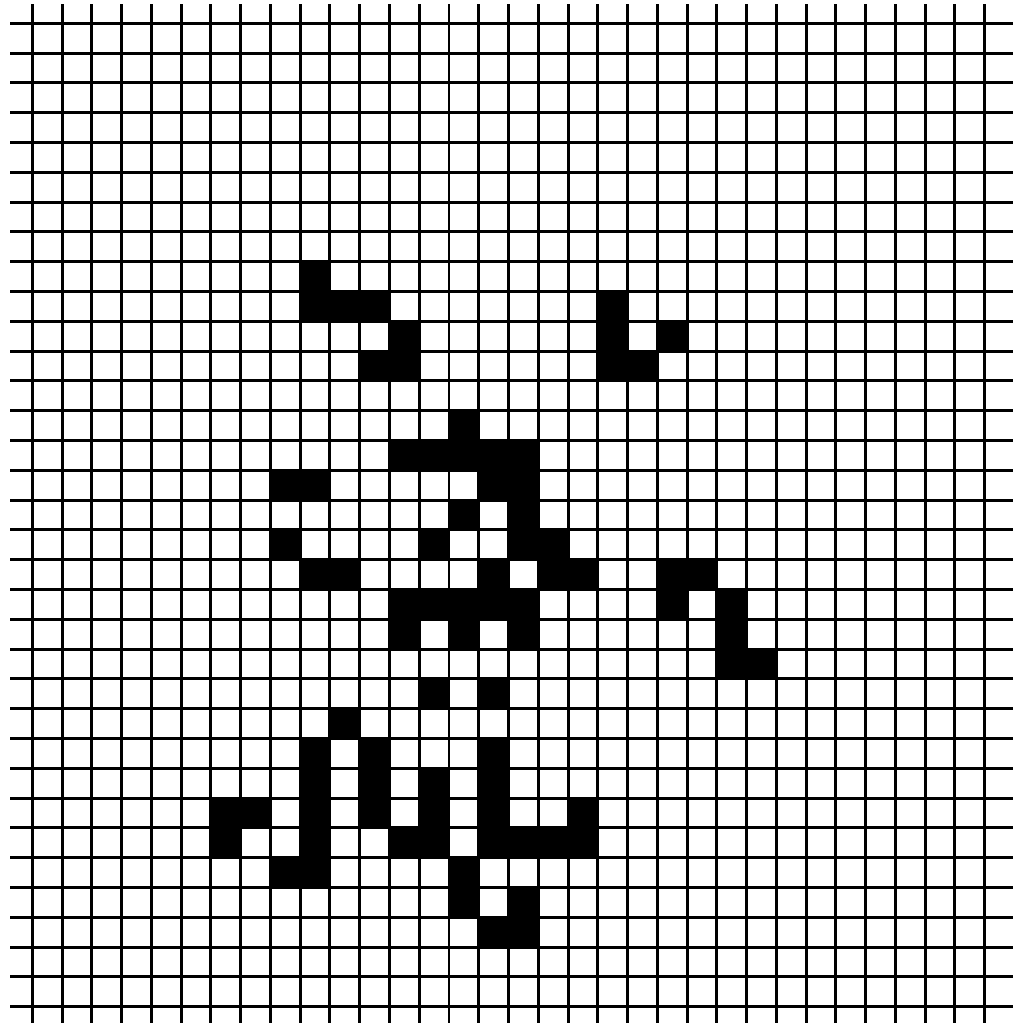
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

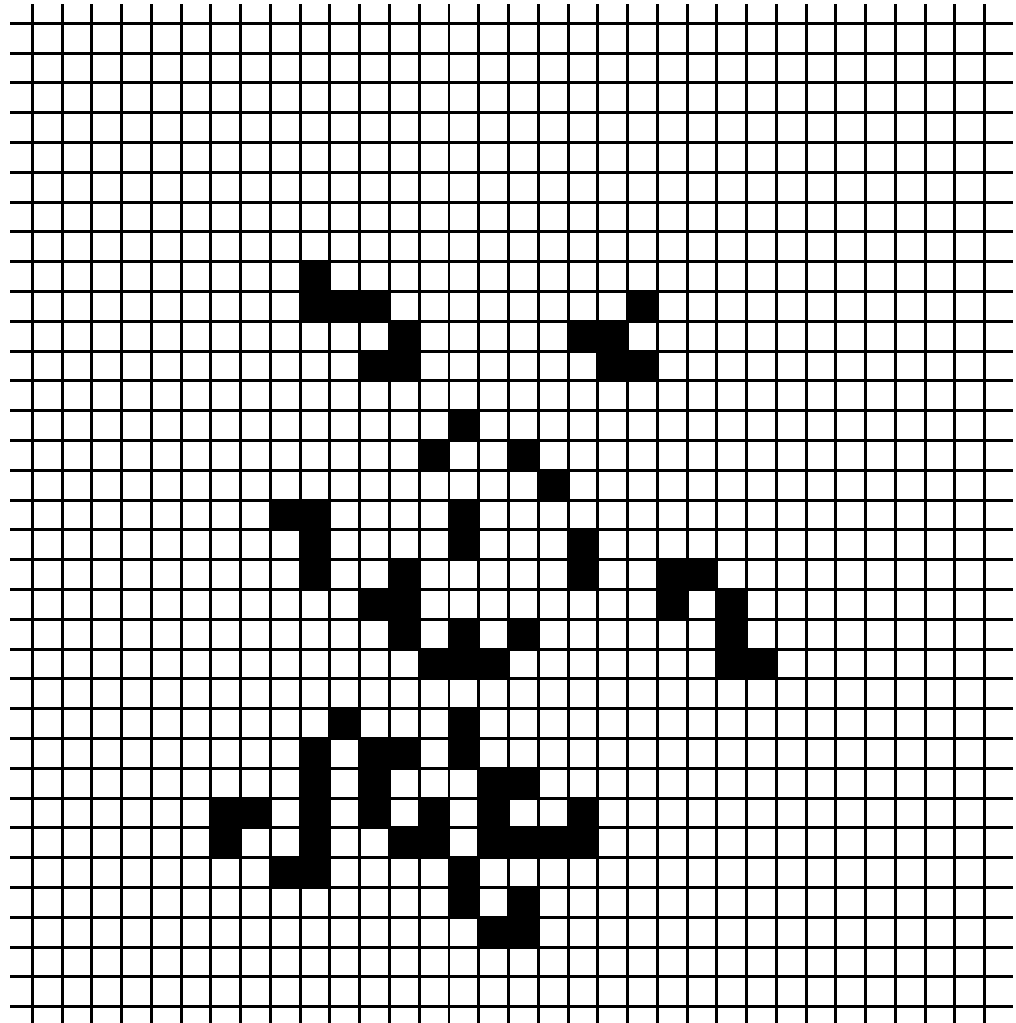
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

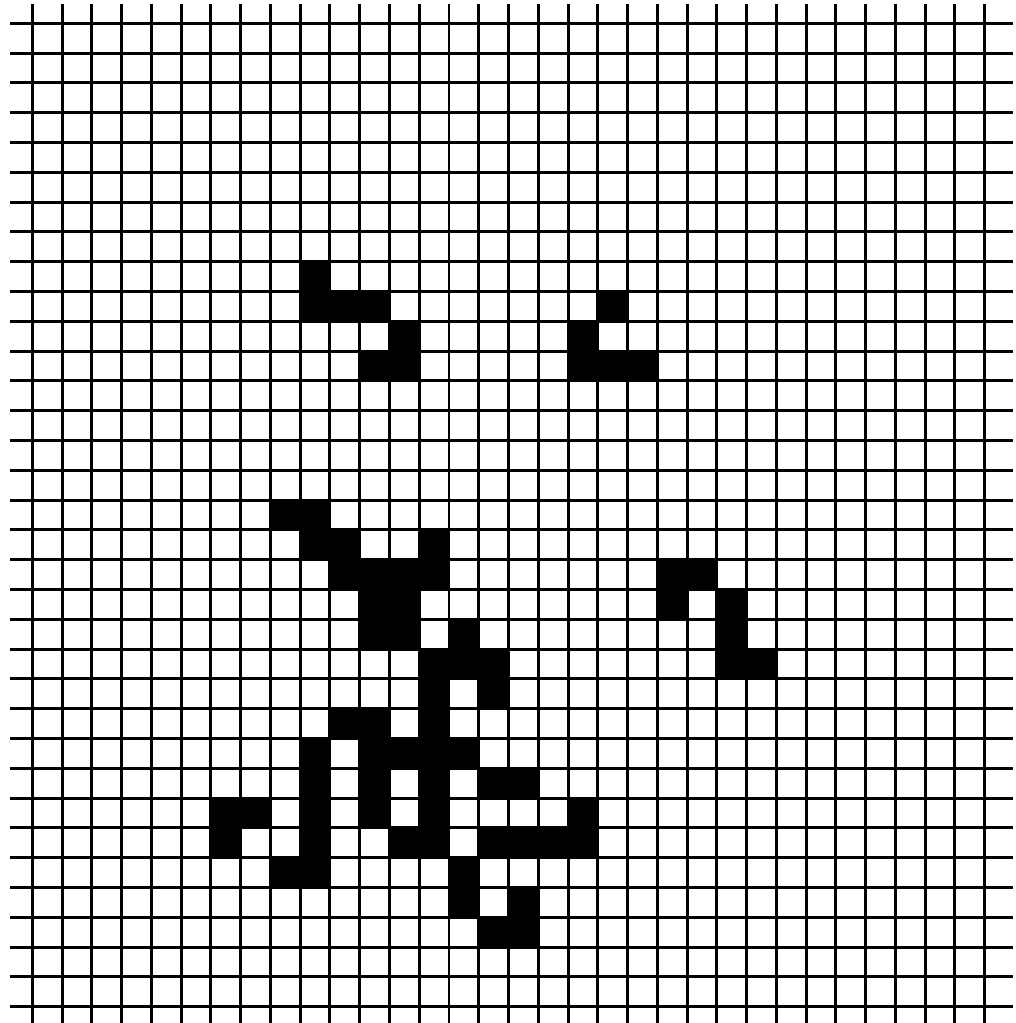
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

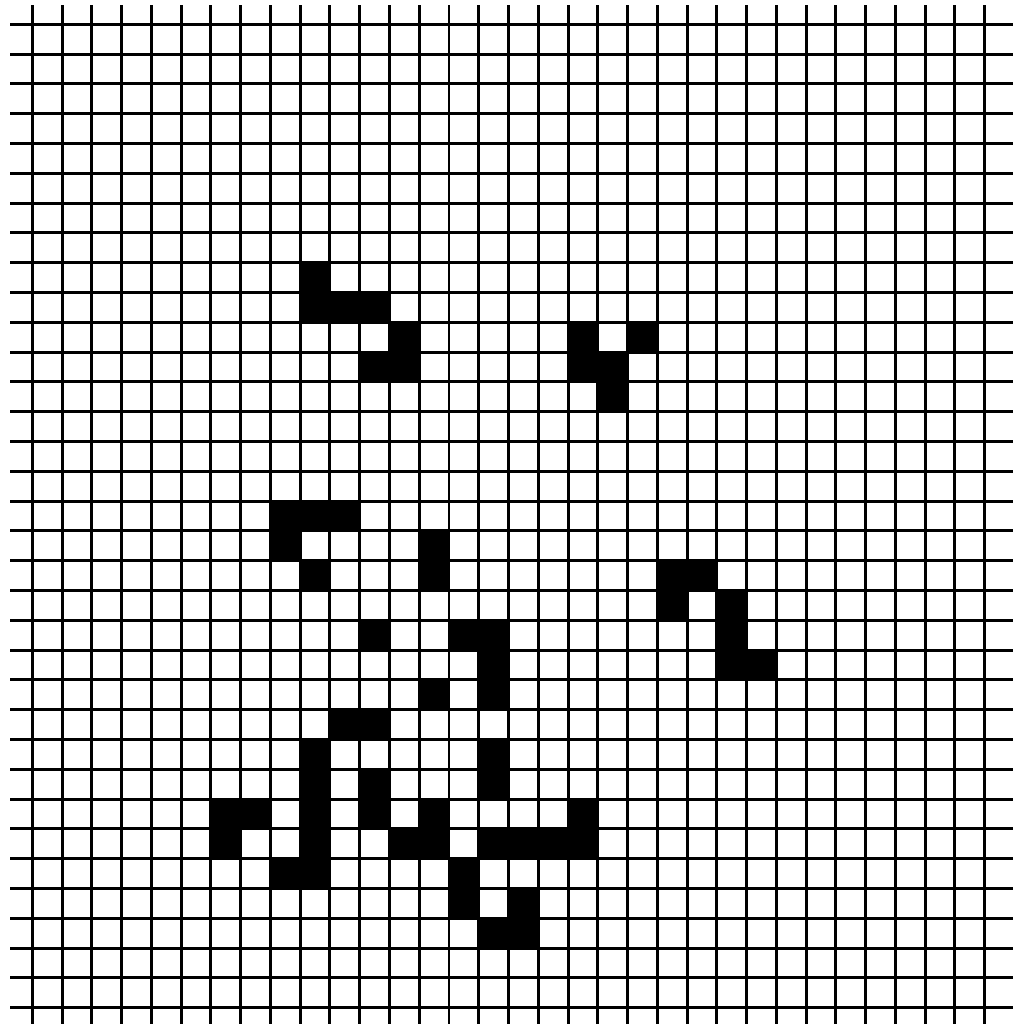
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

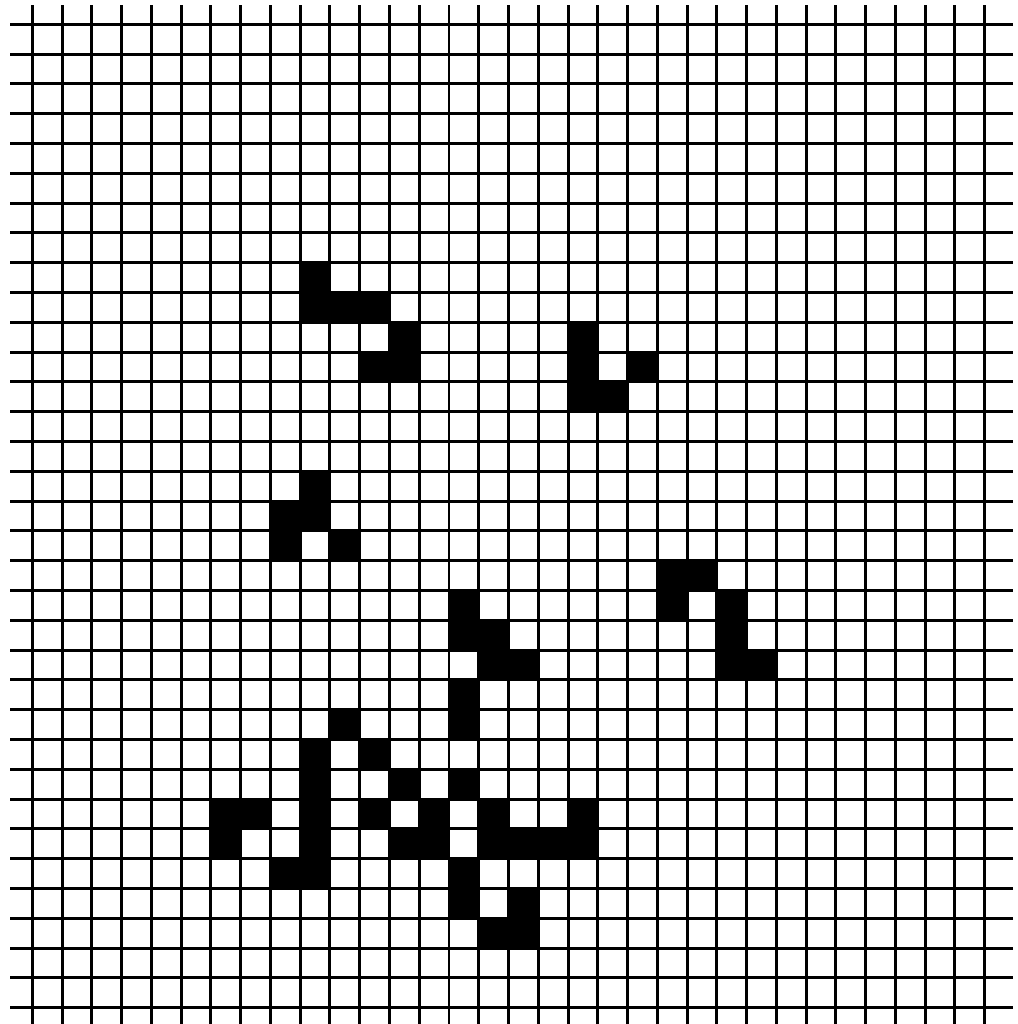
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

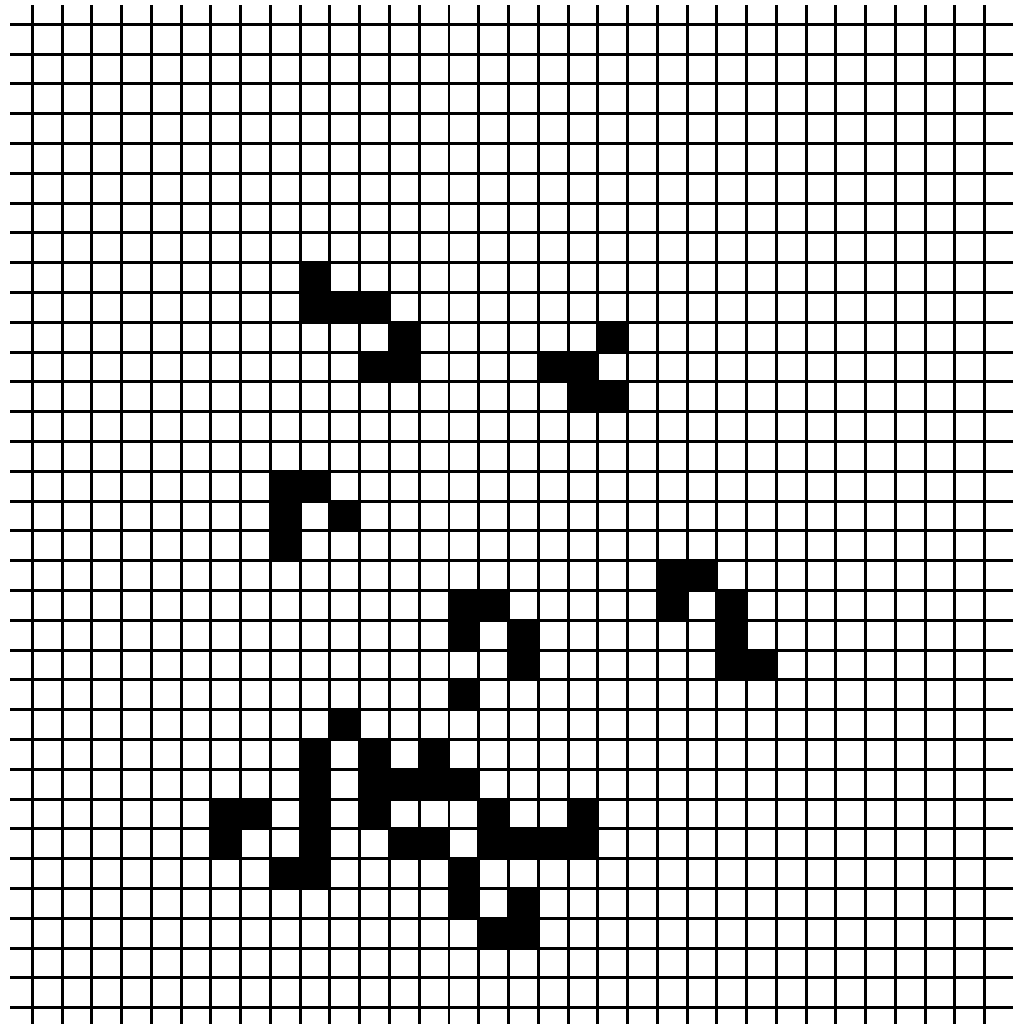
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

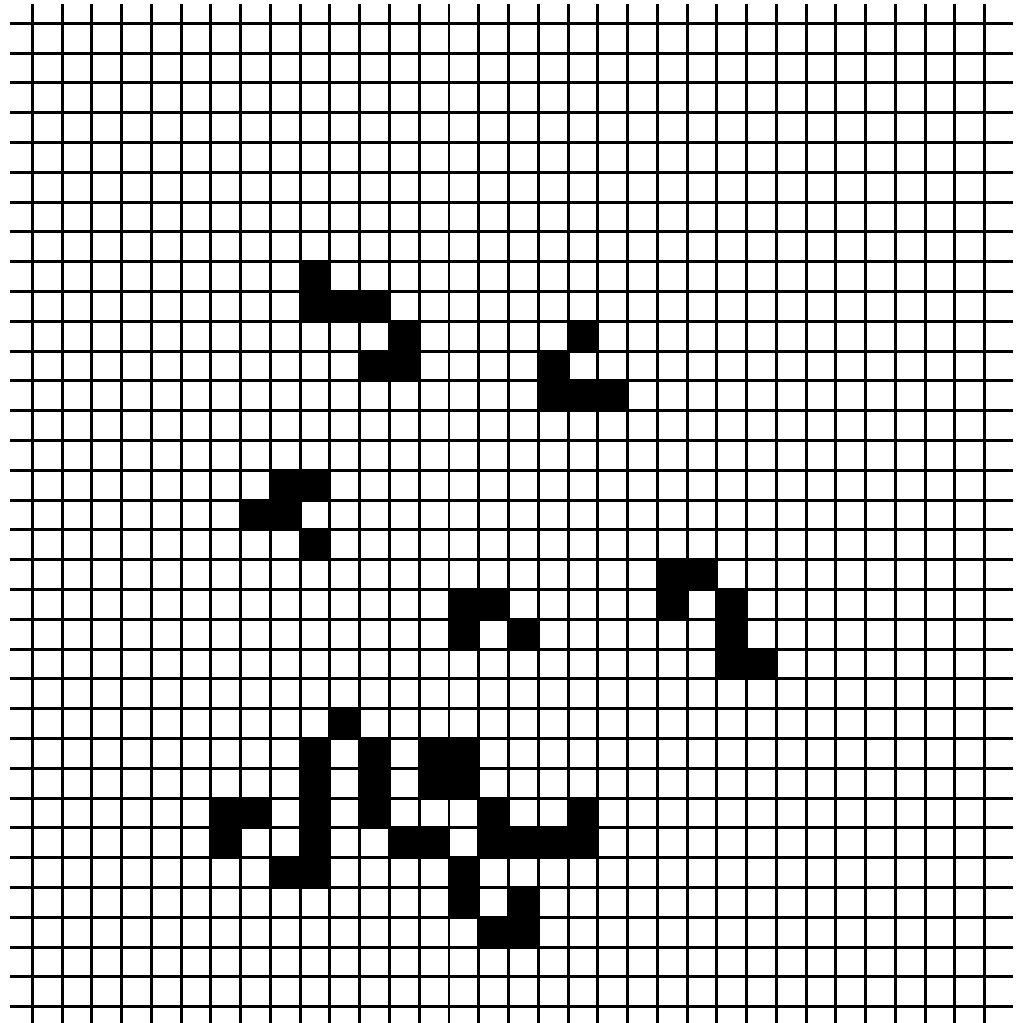
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

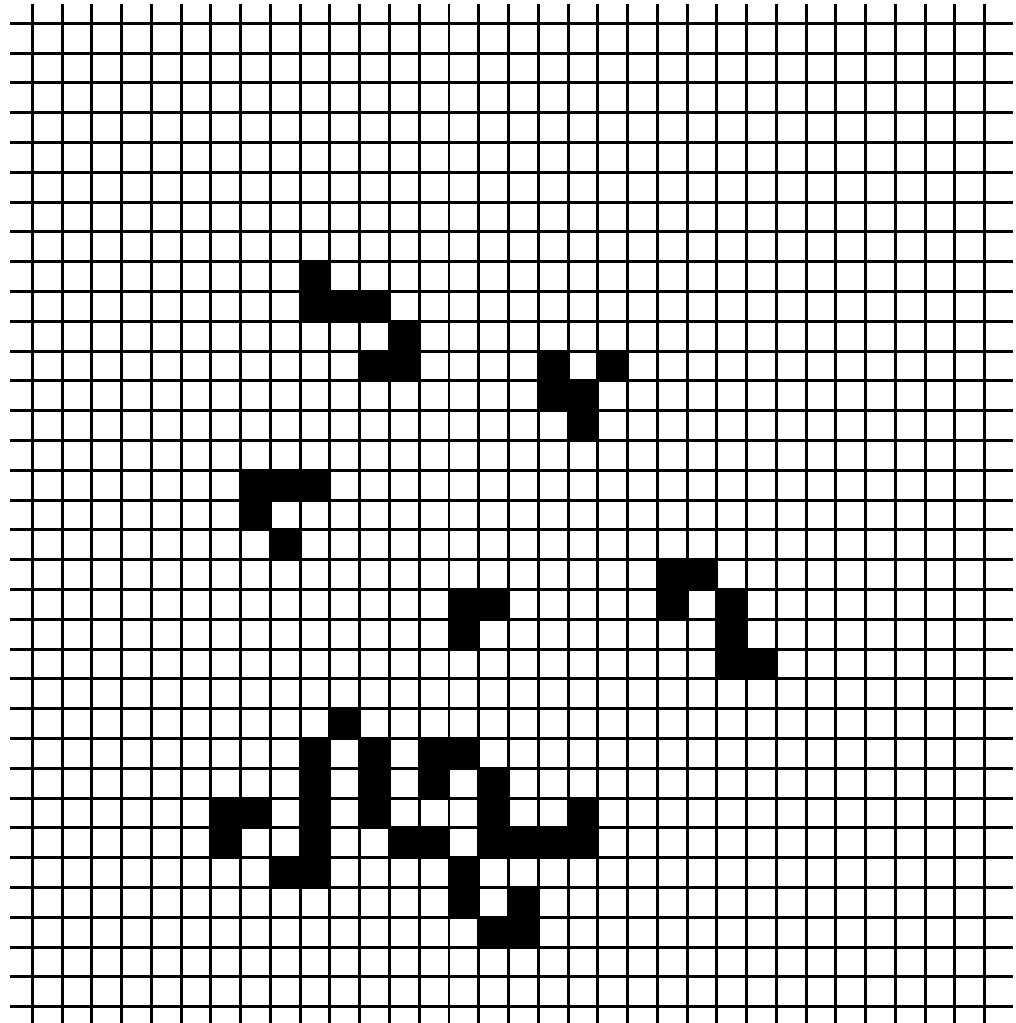
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

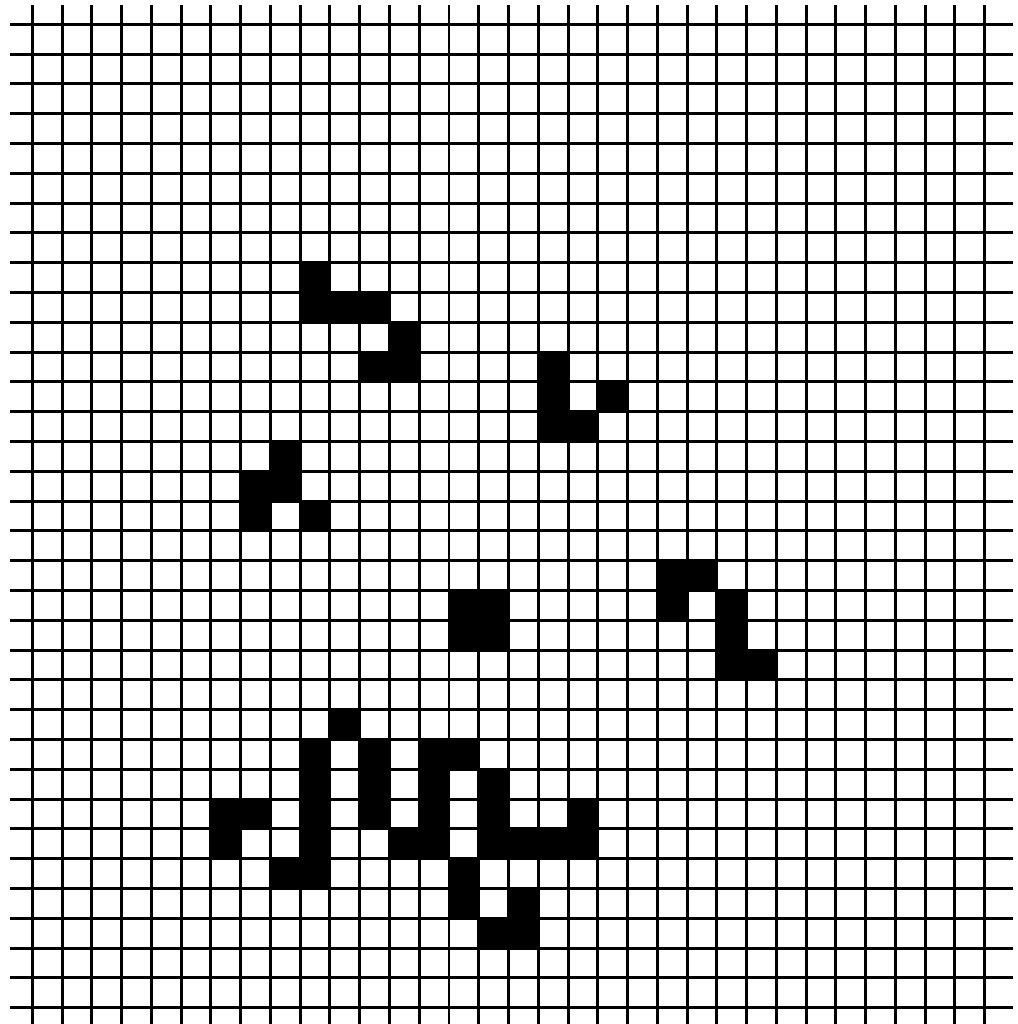
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

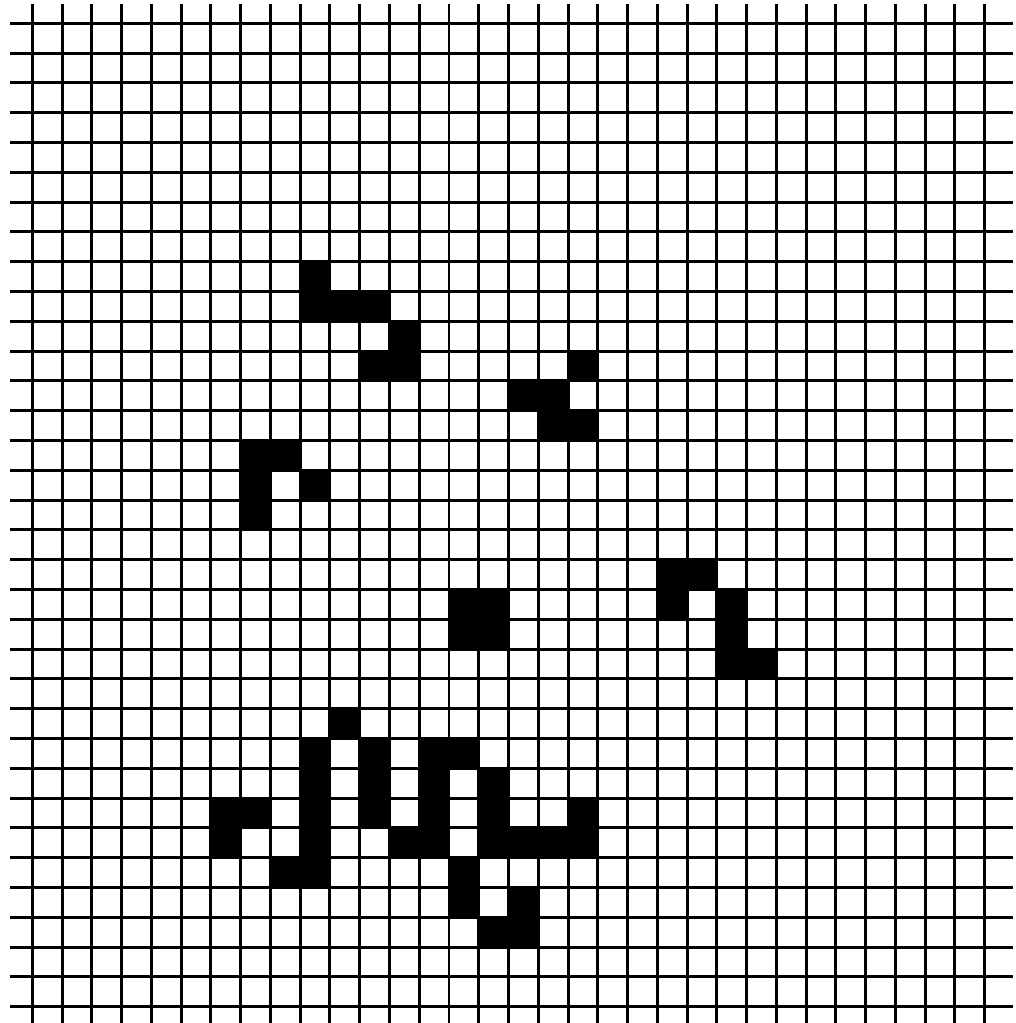
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

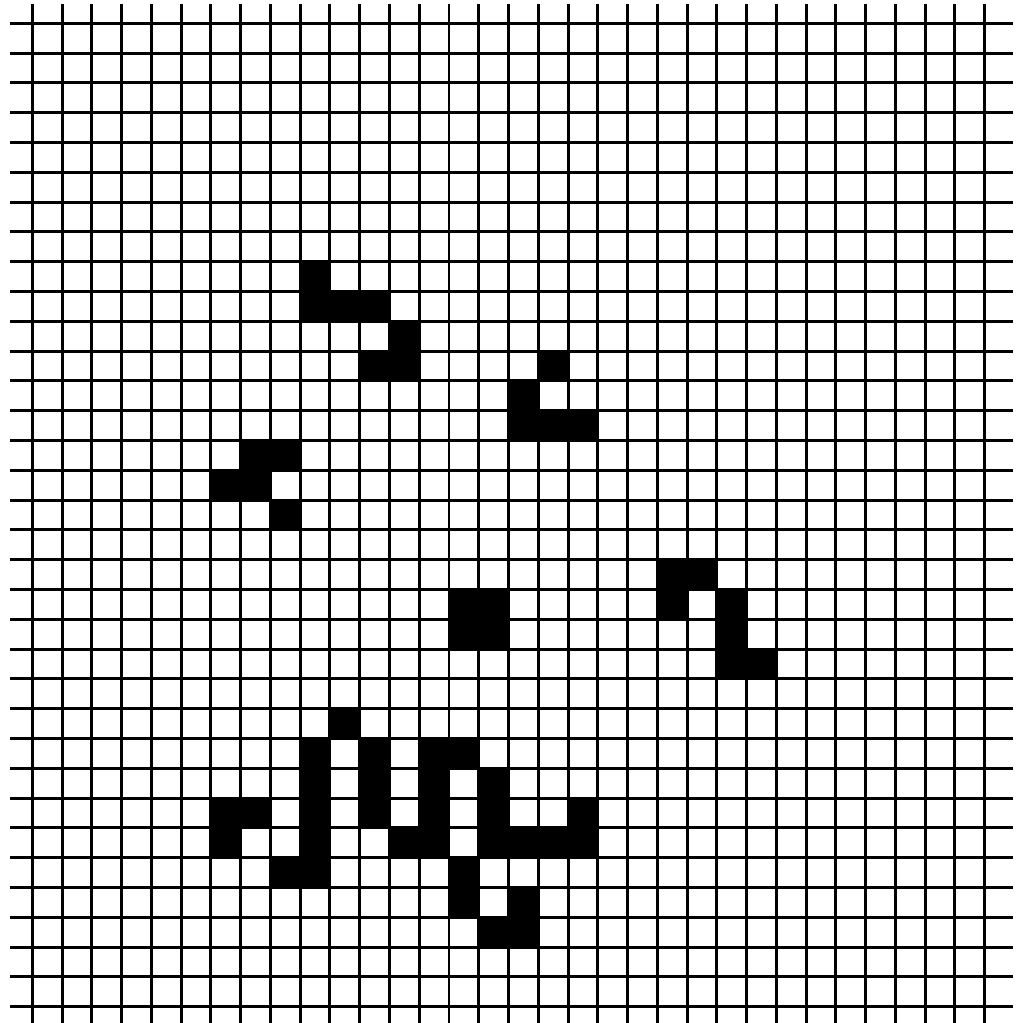
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

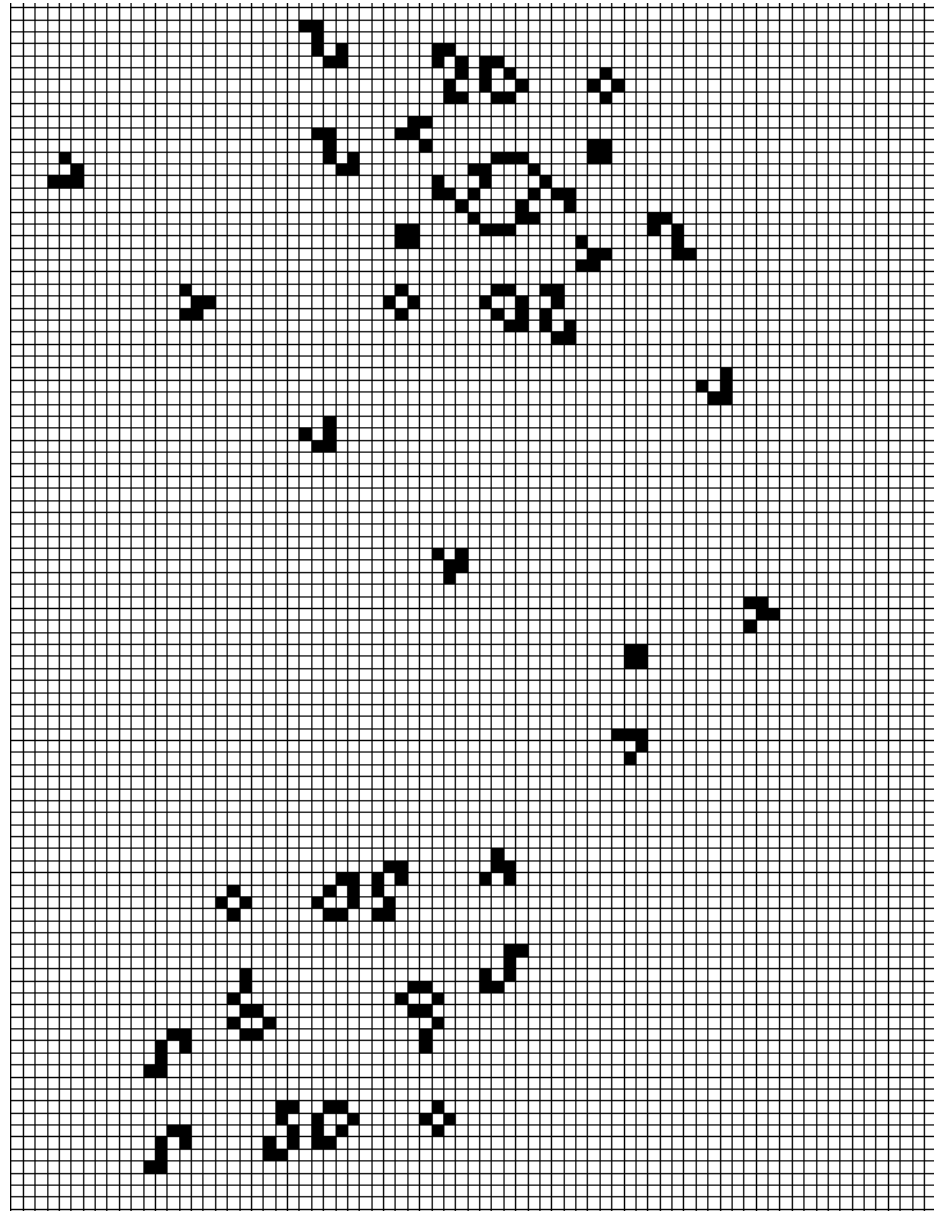
Reflector :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

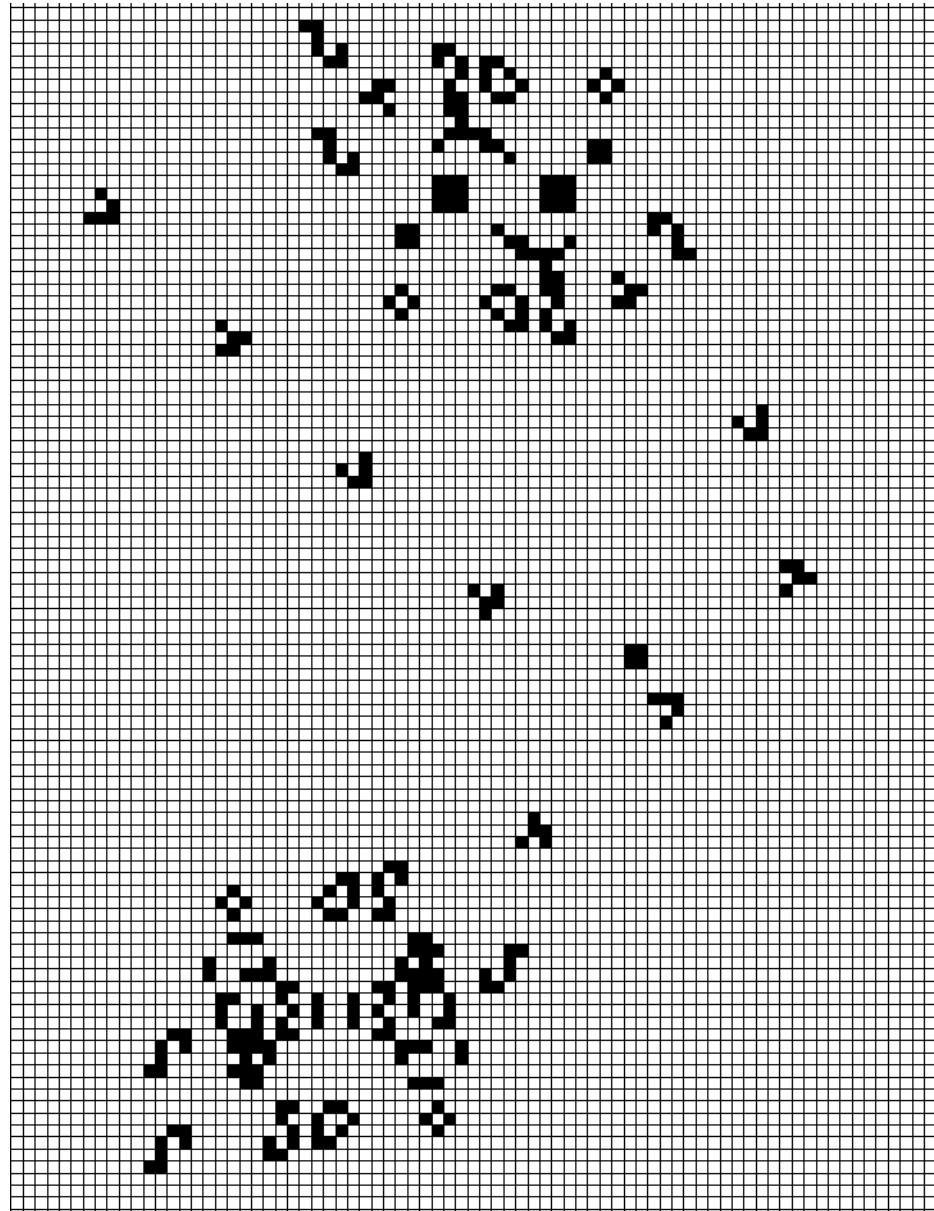
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

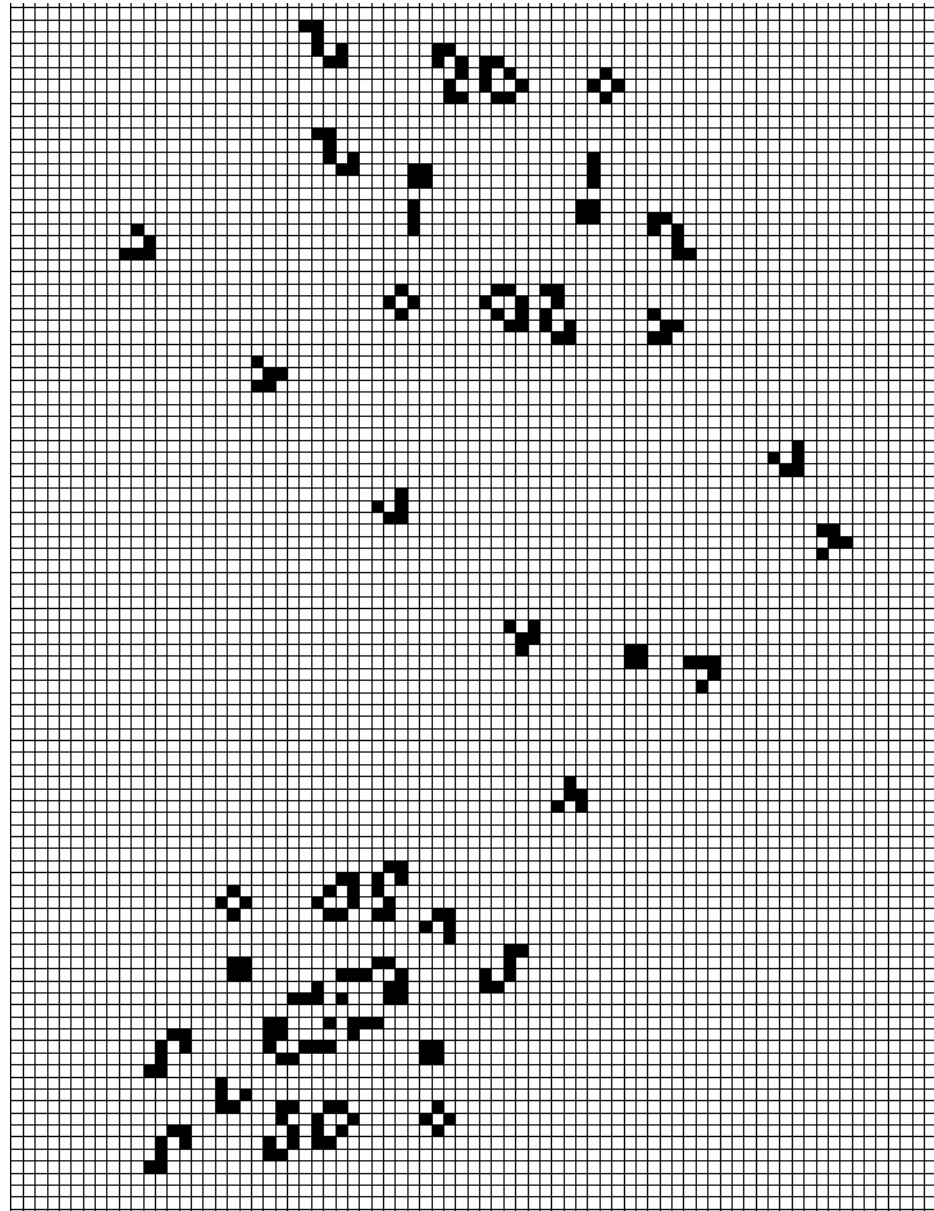
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

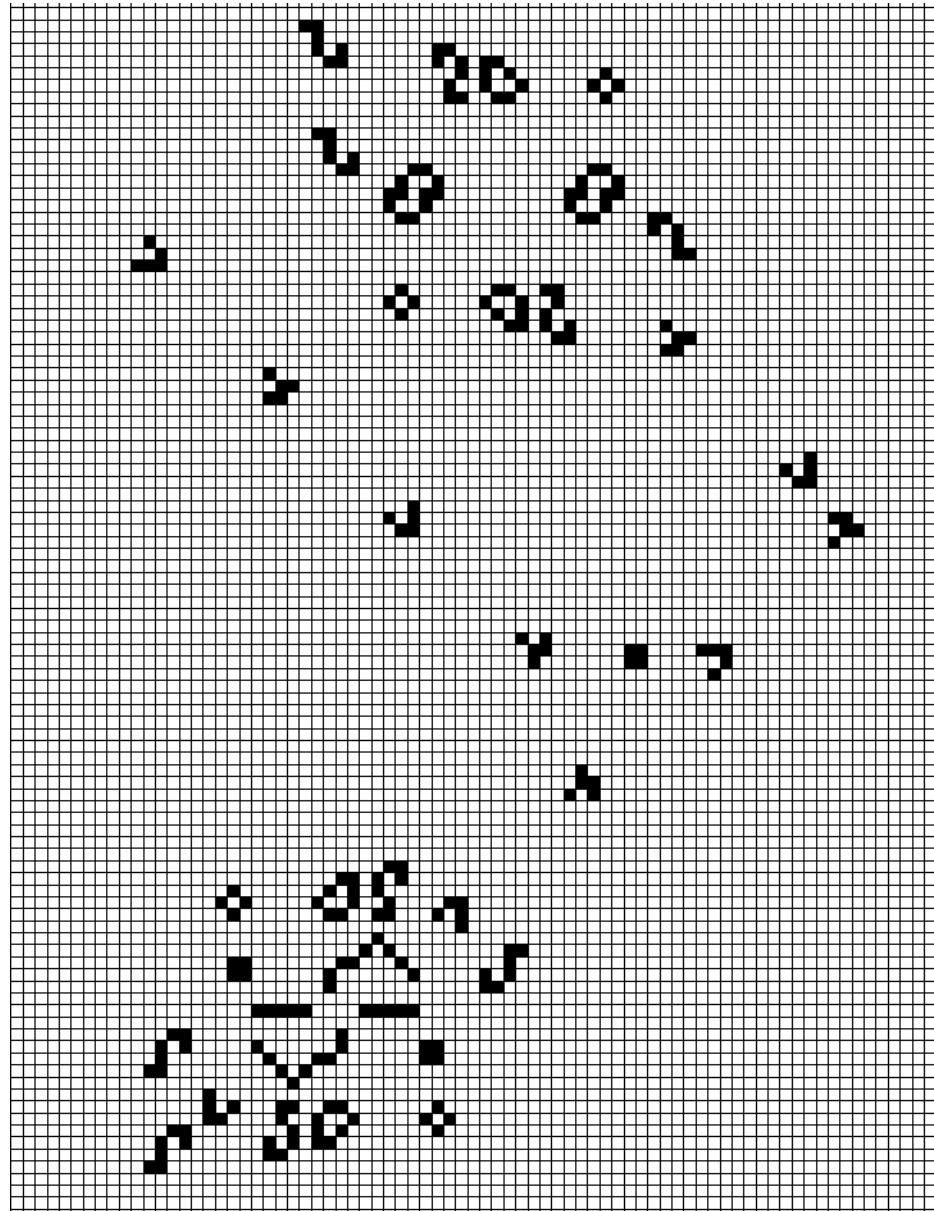
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

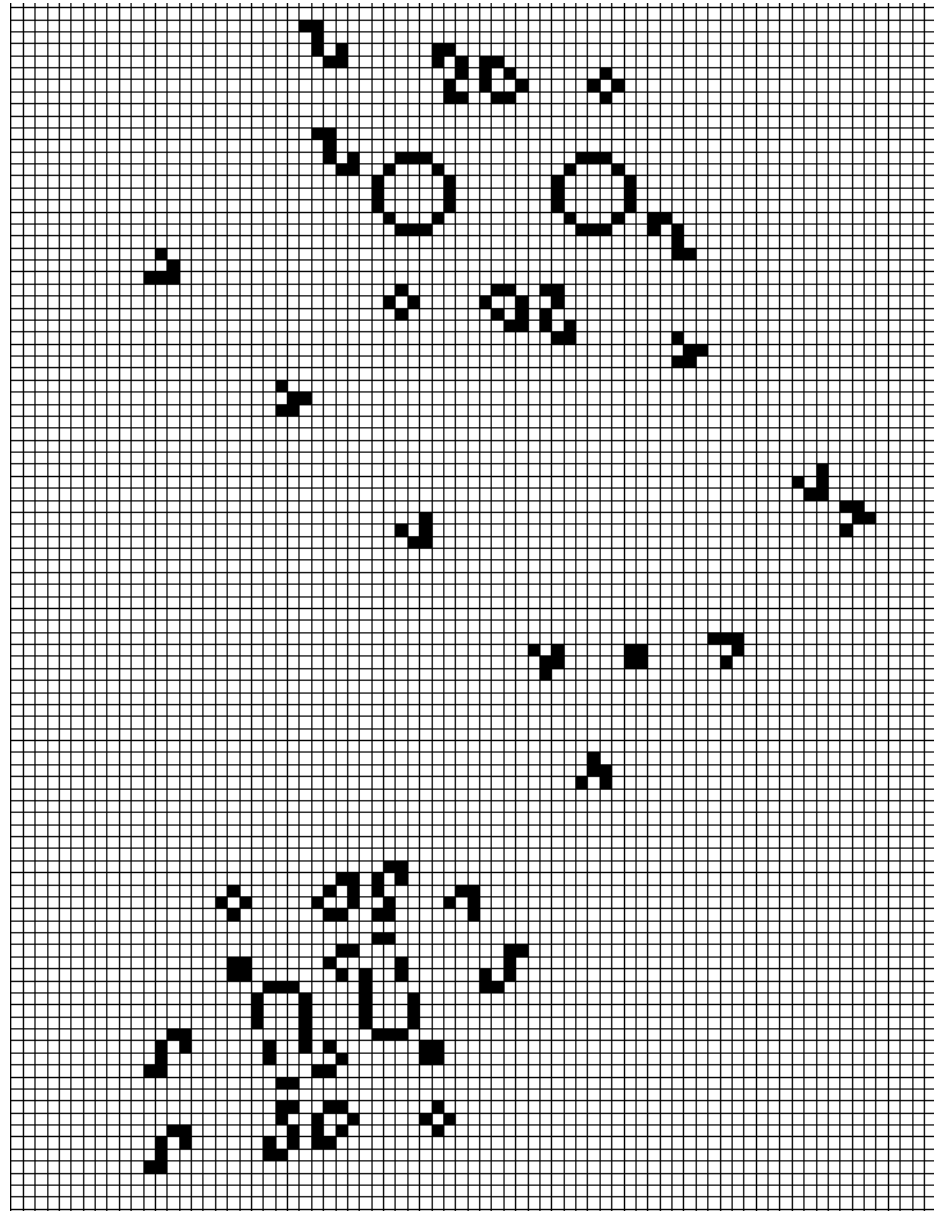
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

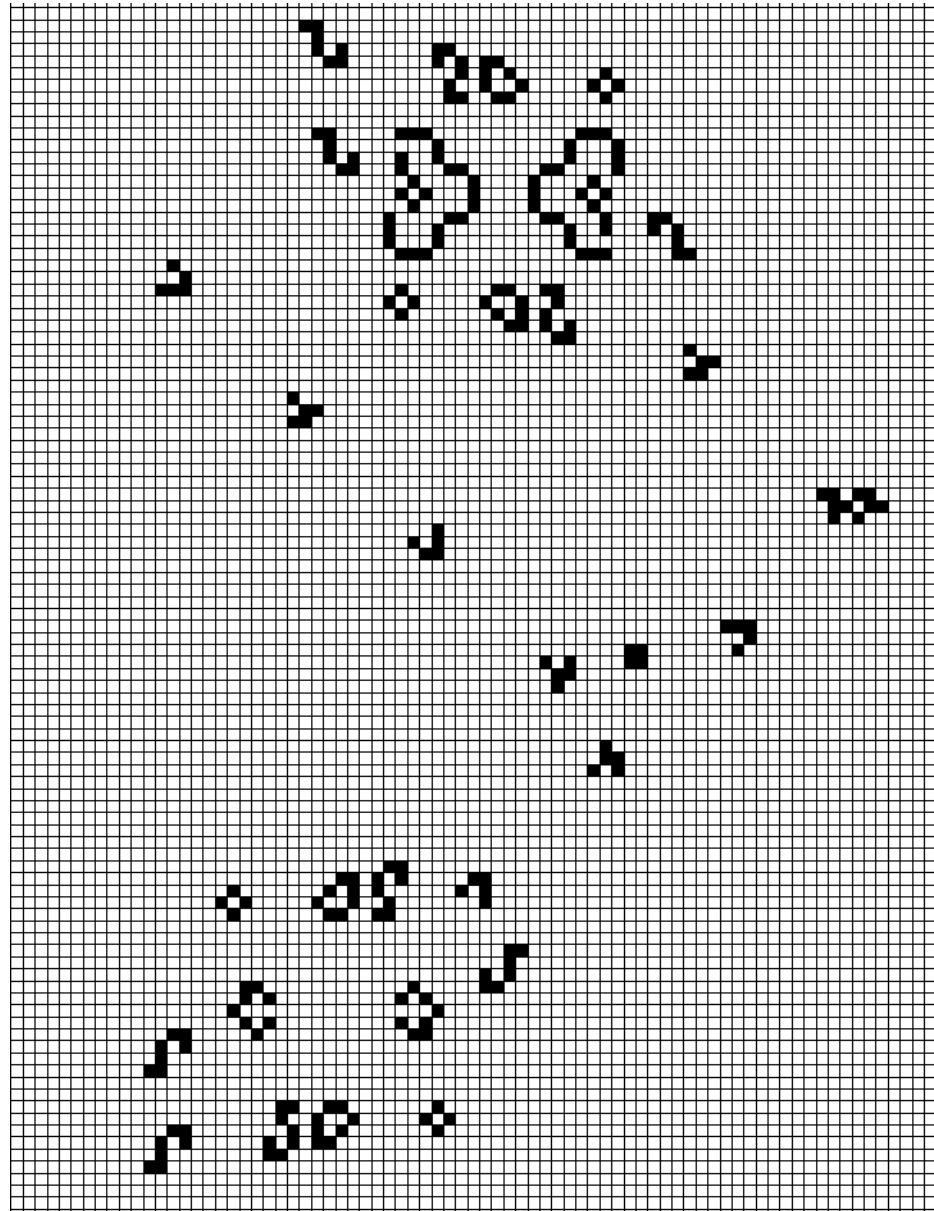
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

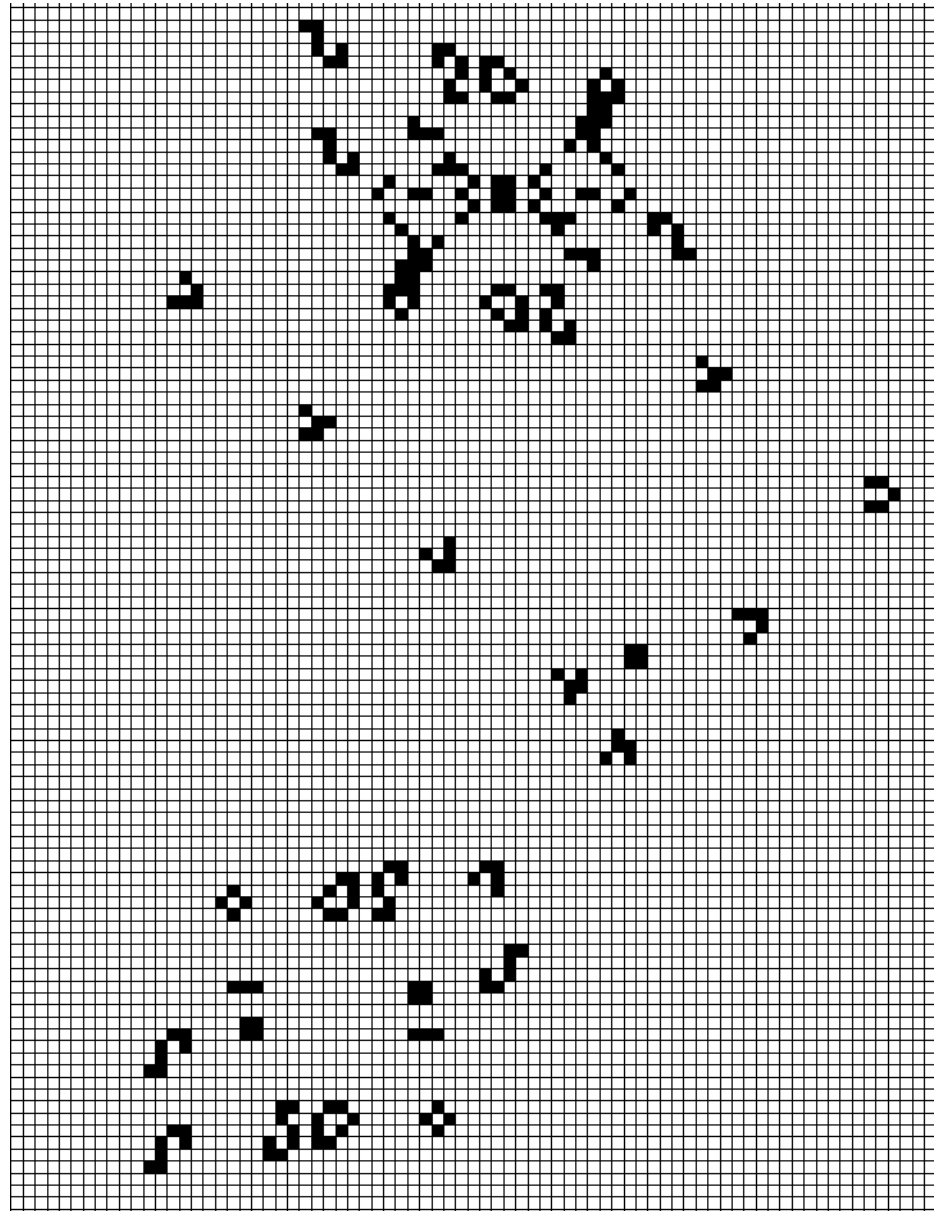
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

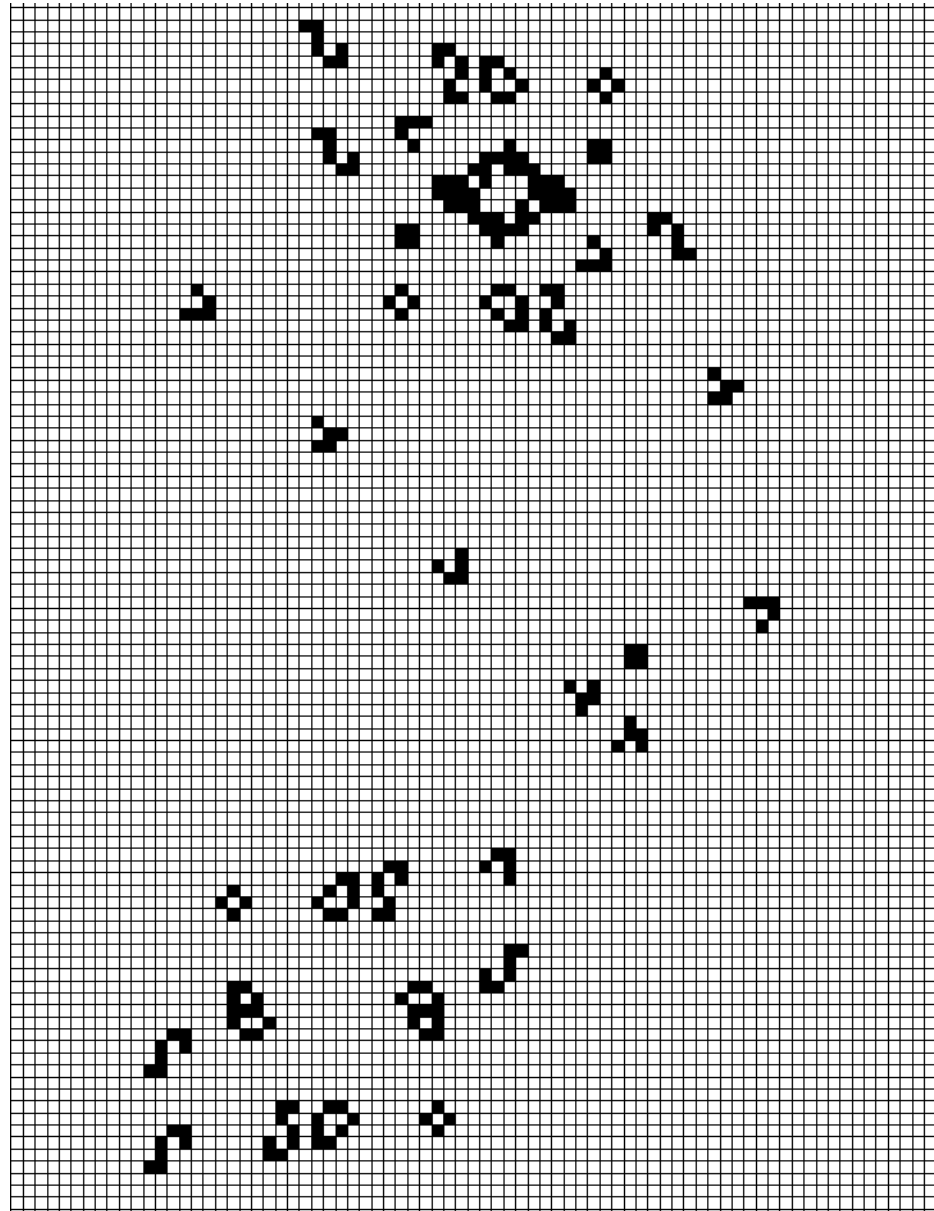
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

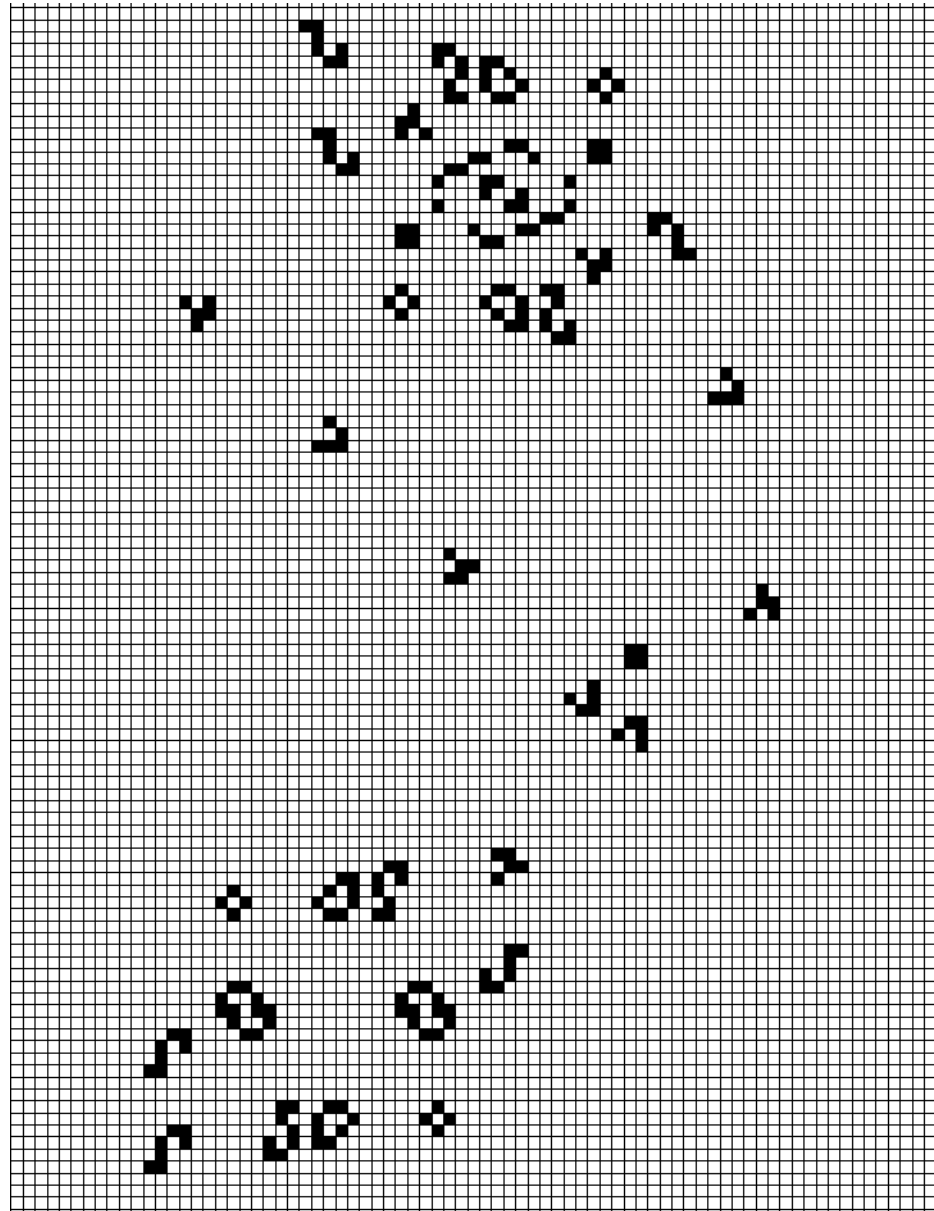
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

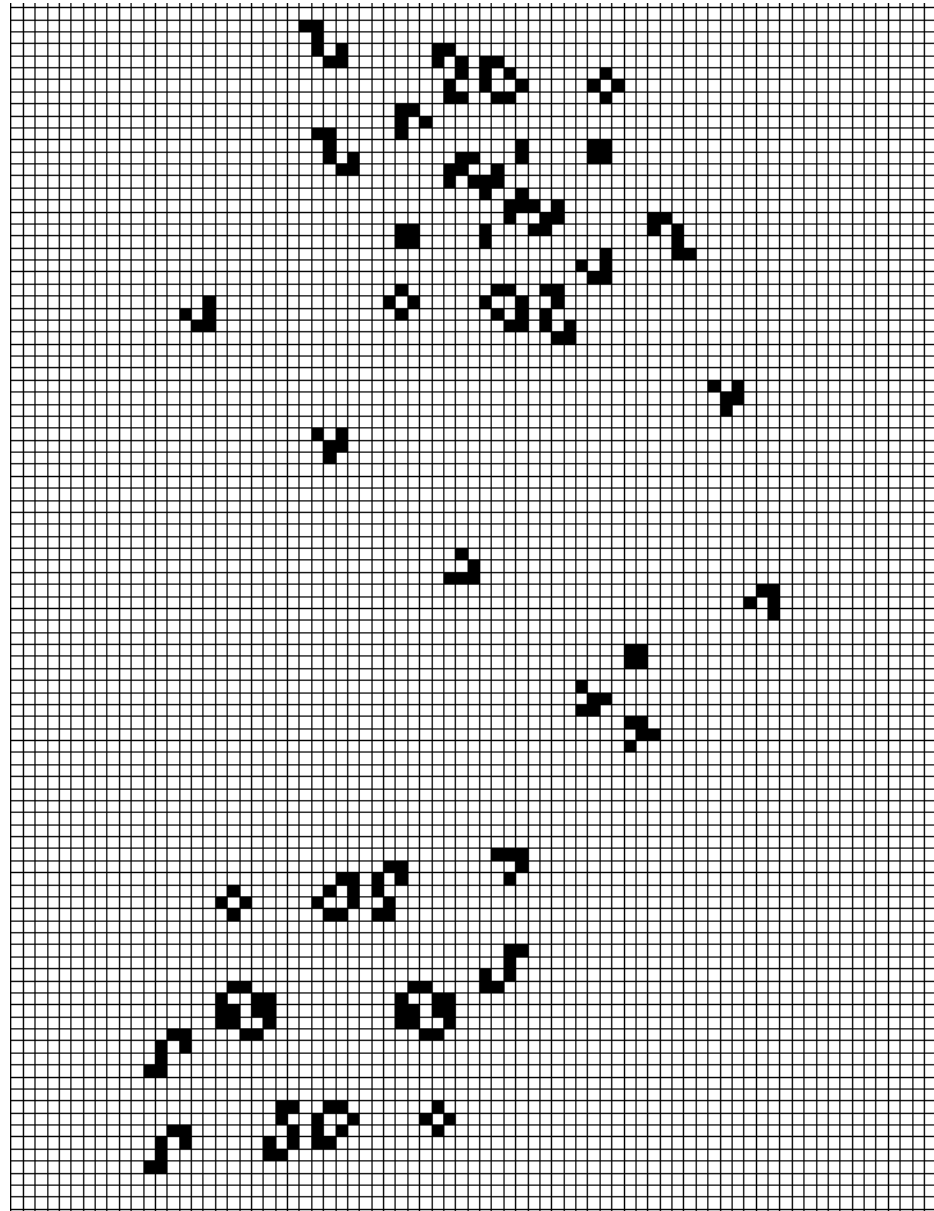
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

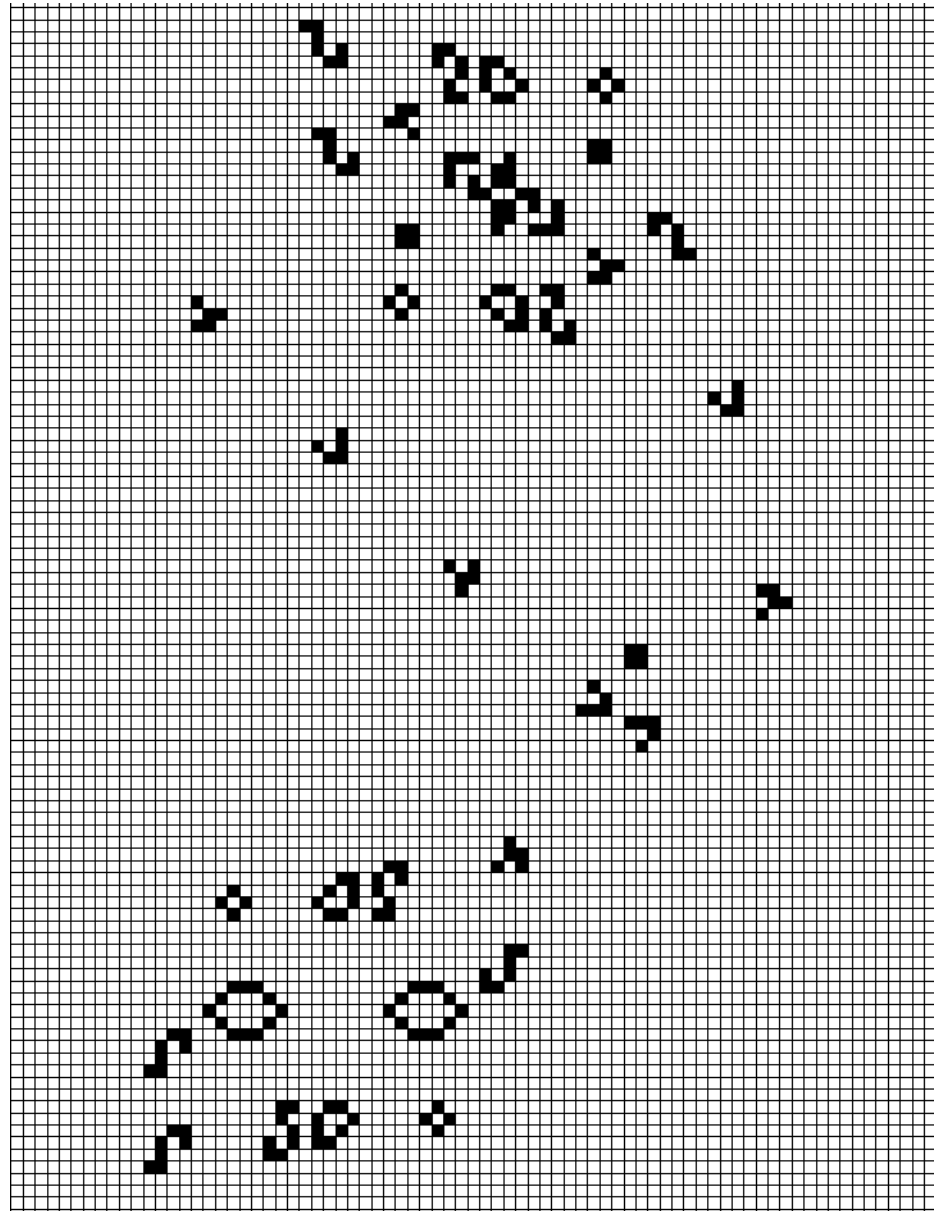
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

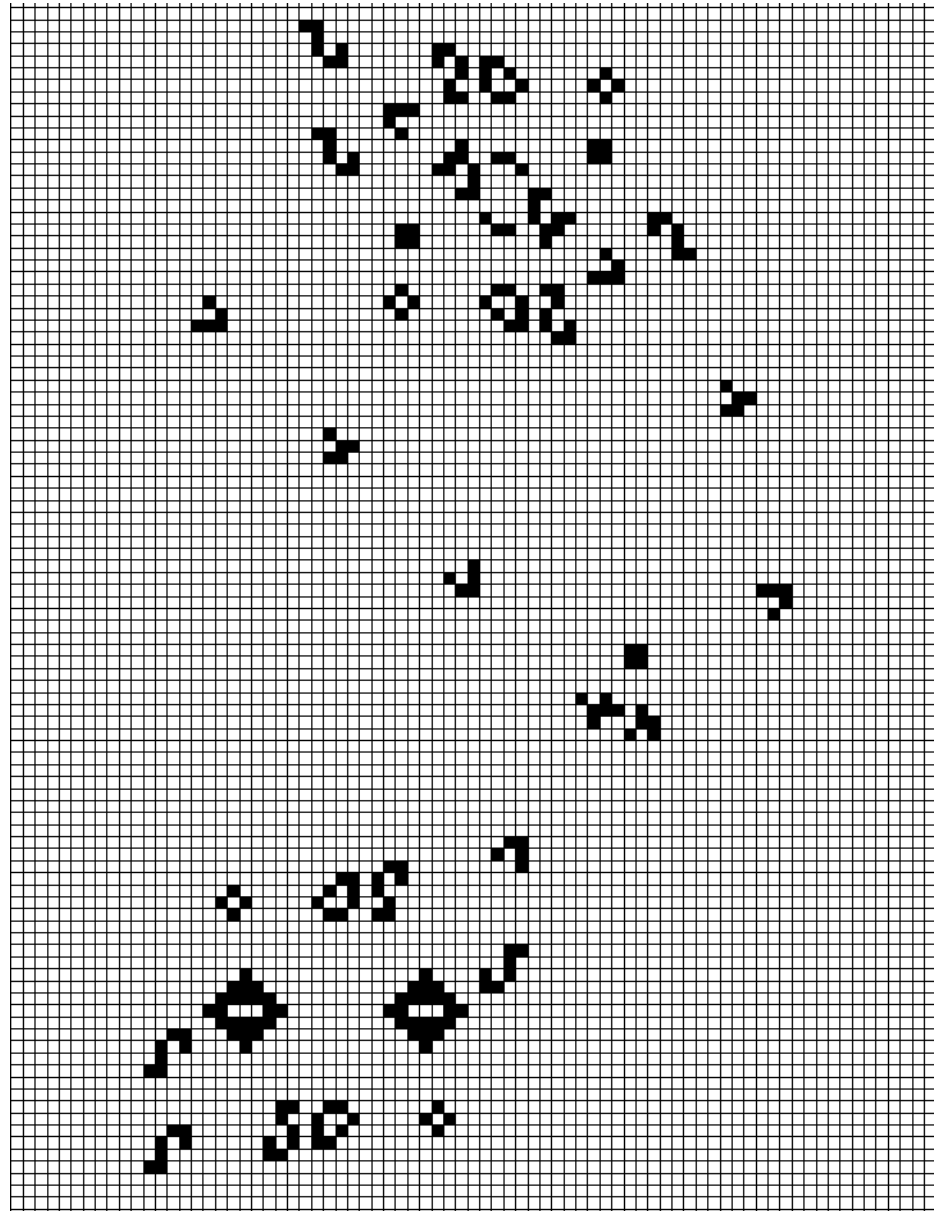
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

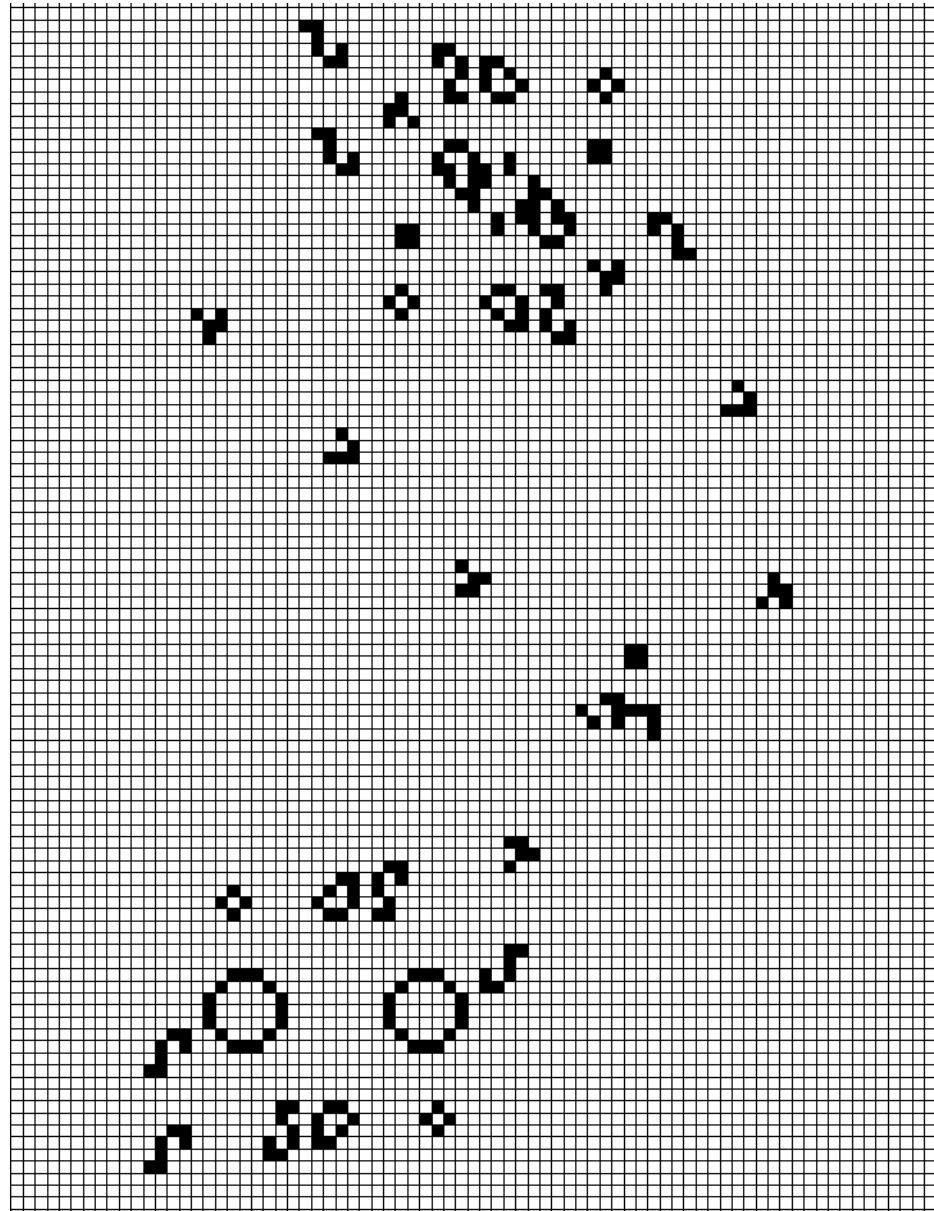
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

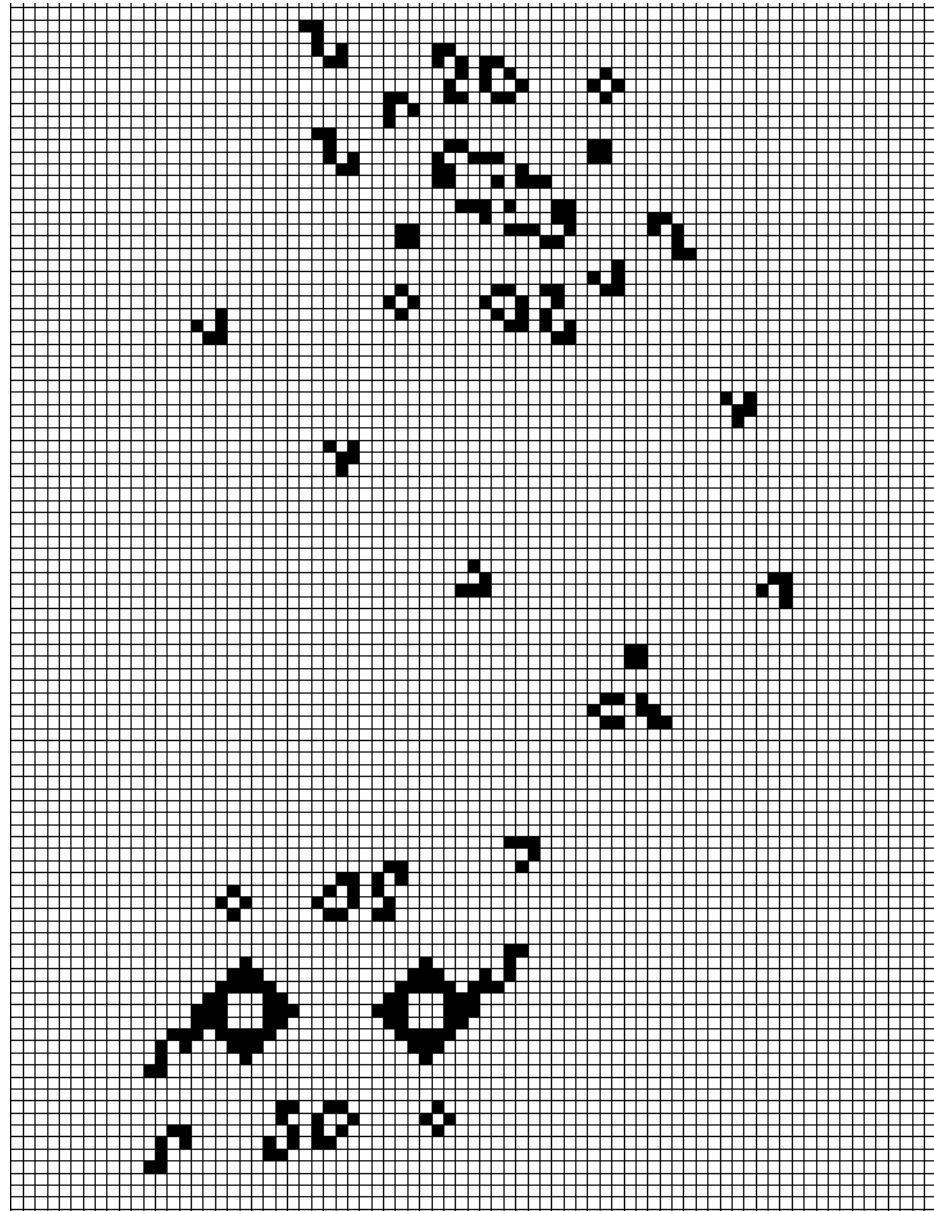
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

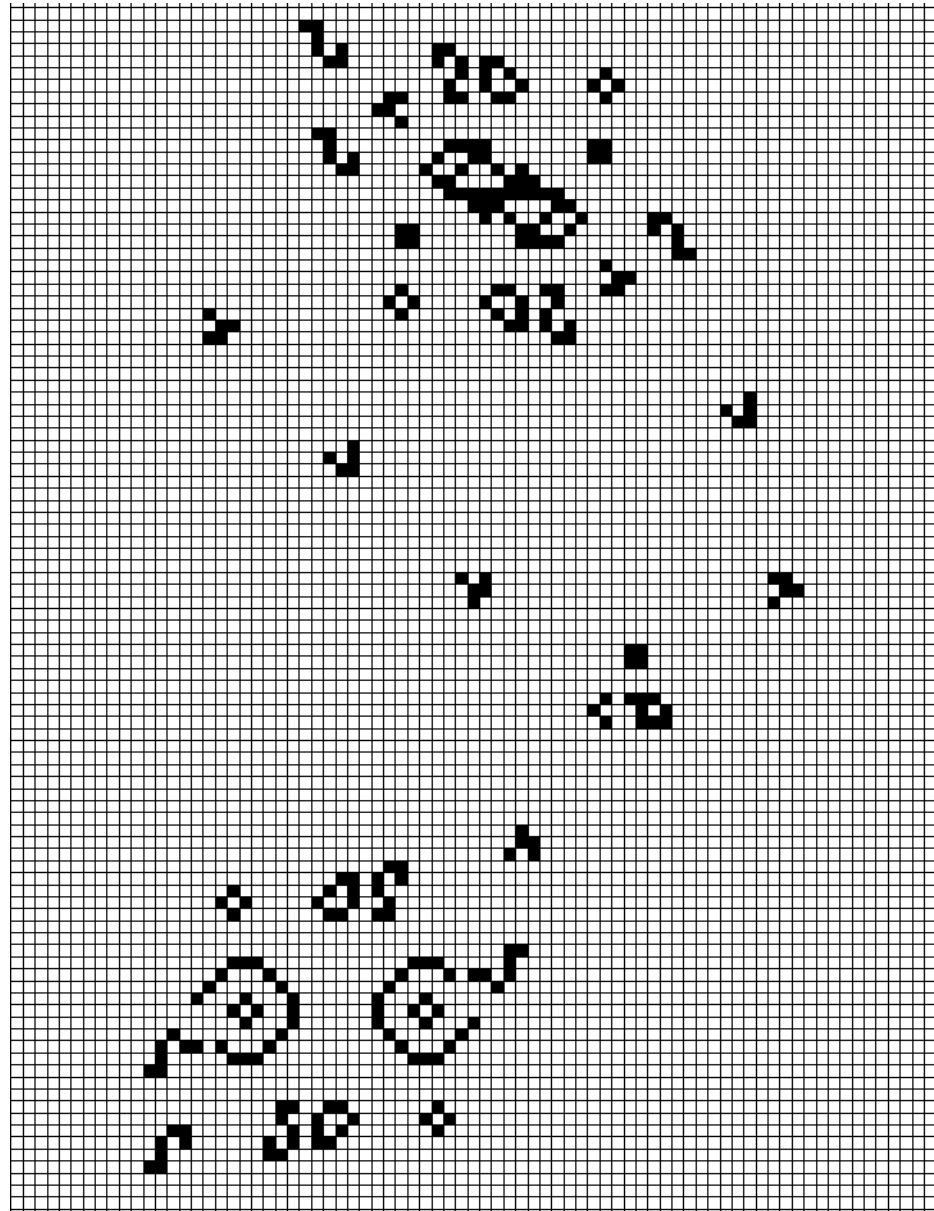
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

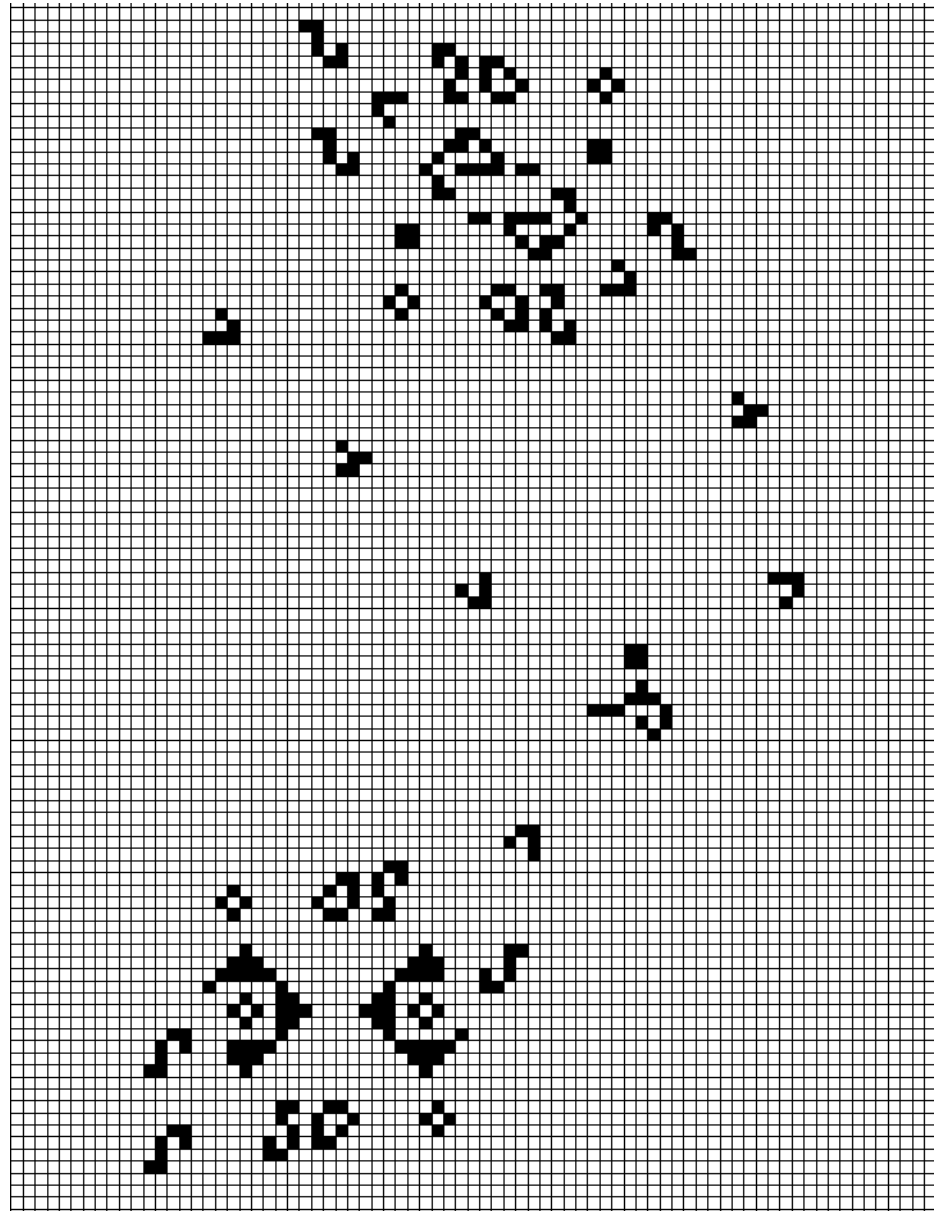
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

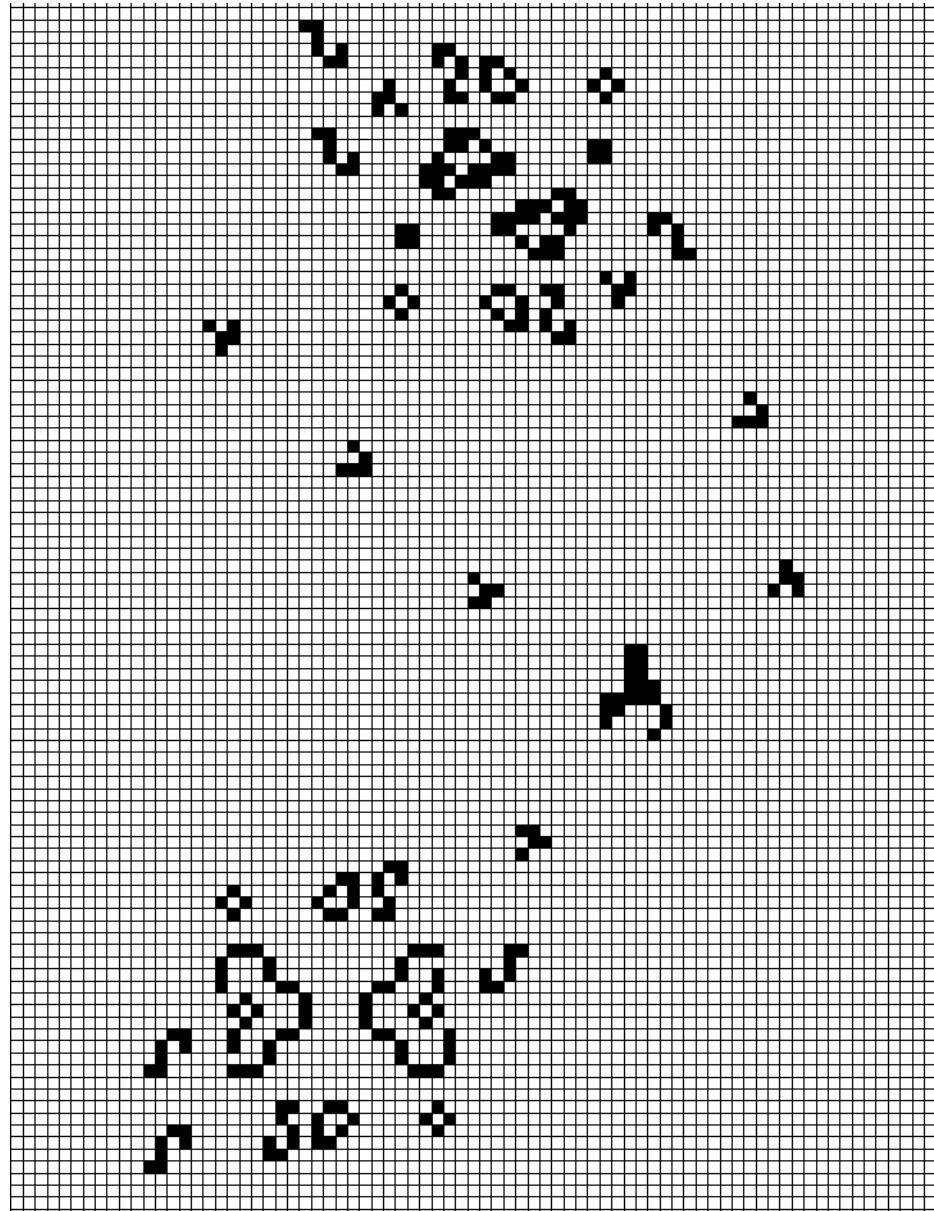
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

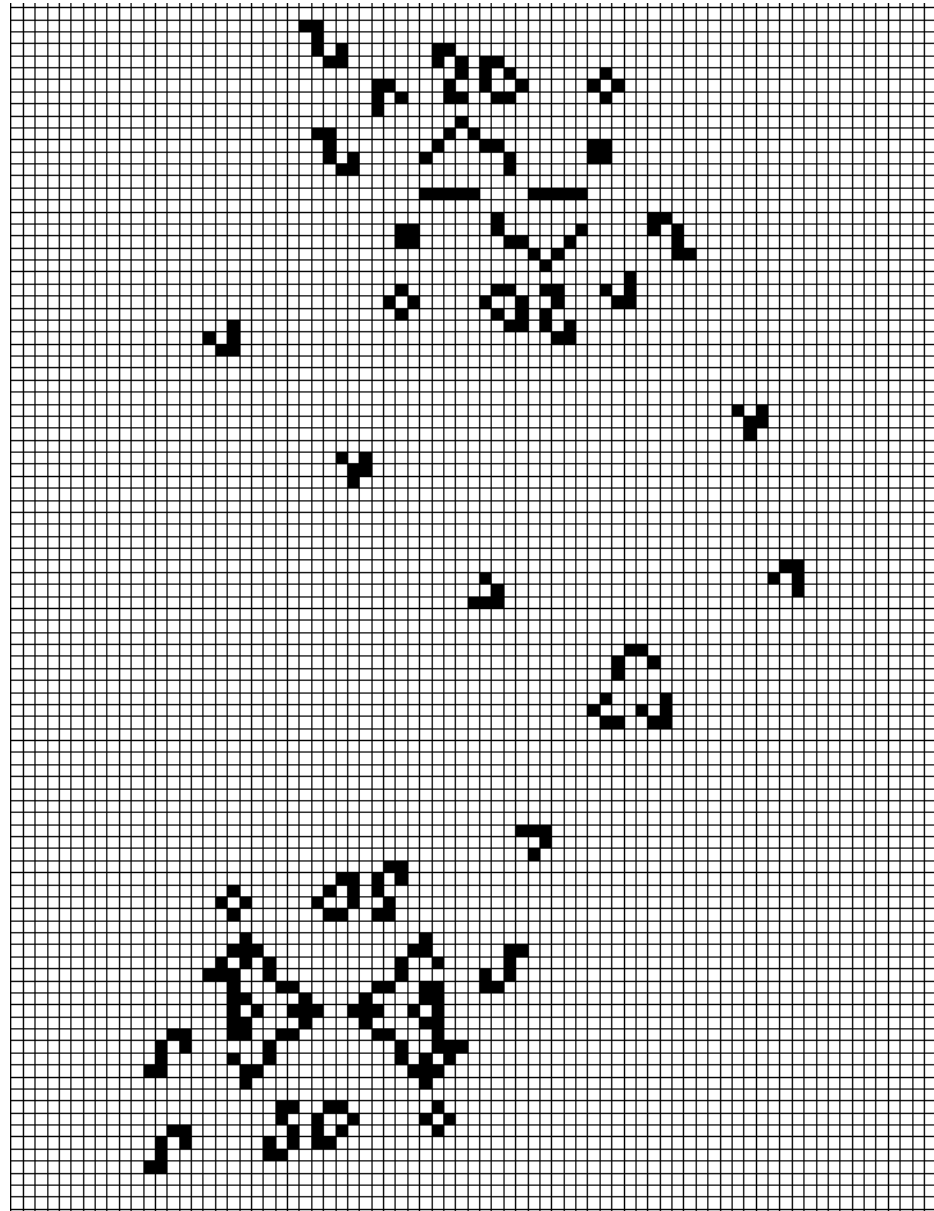
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

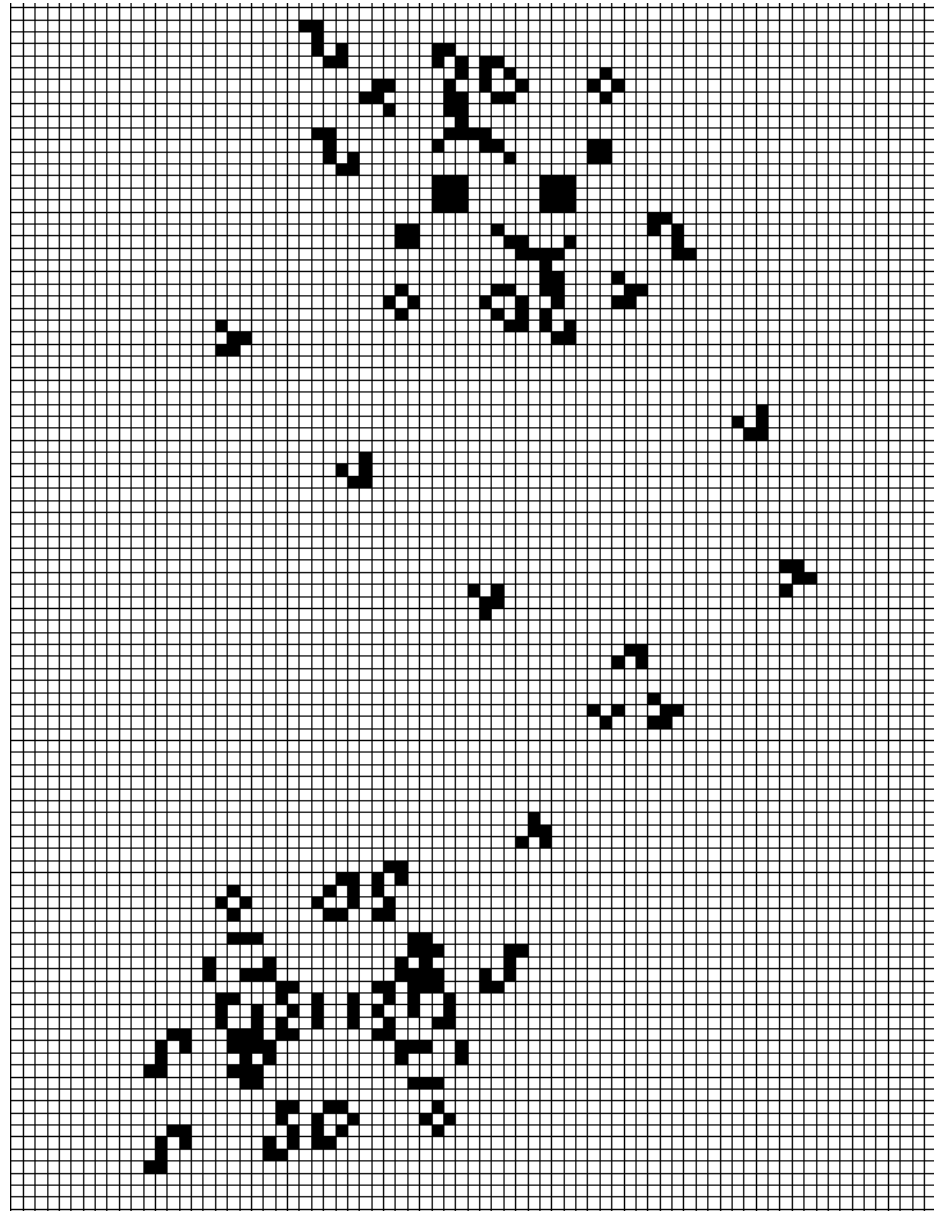
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

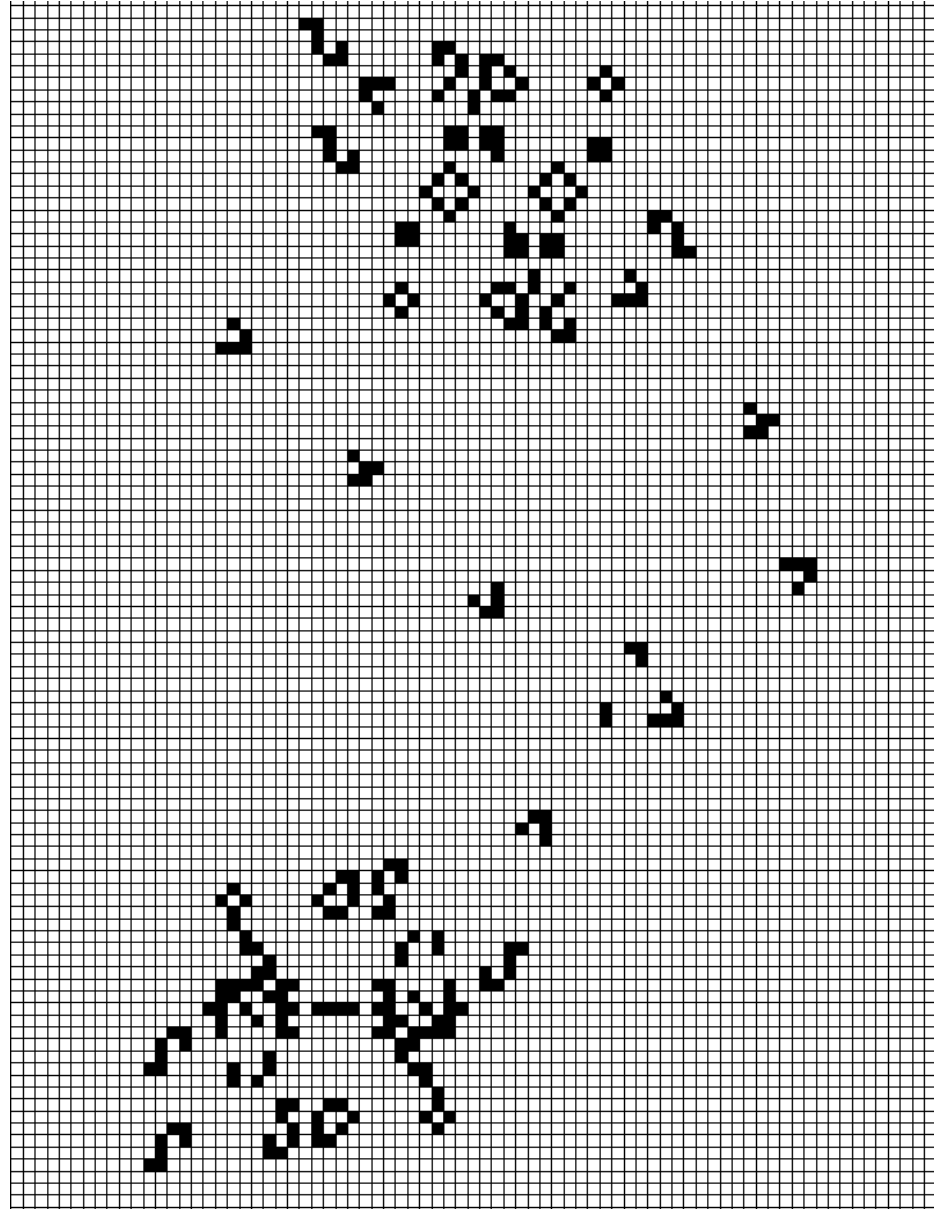
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

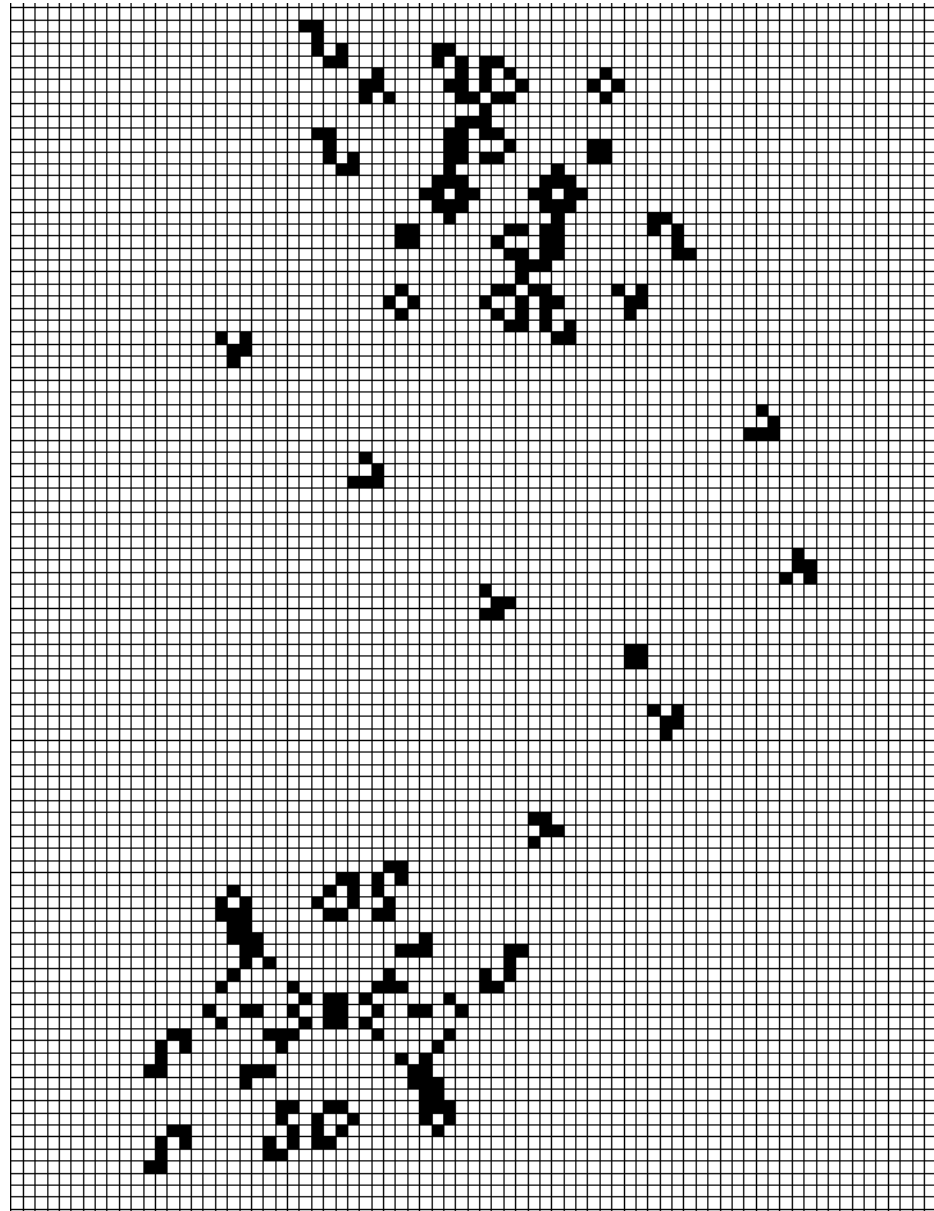
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

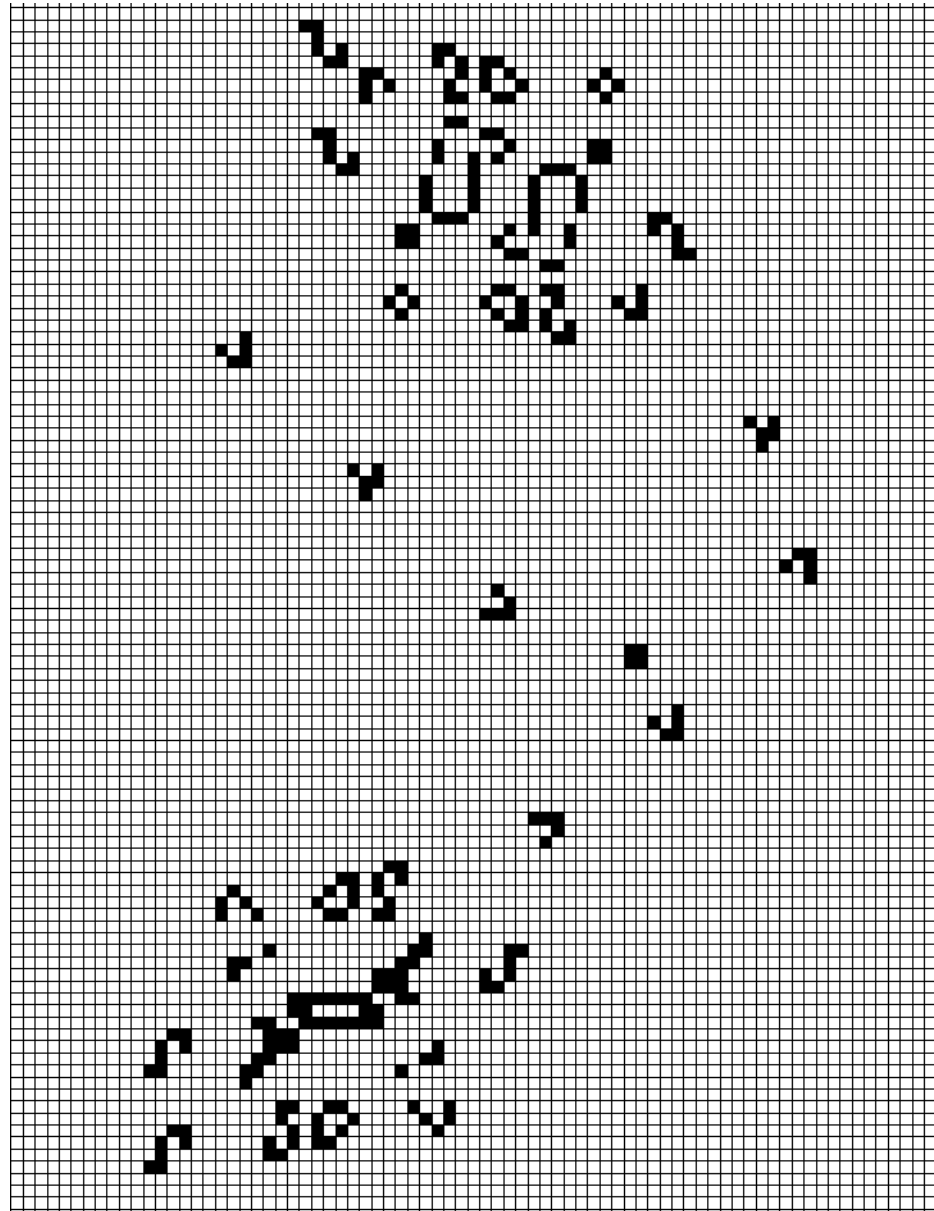
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

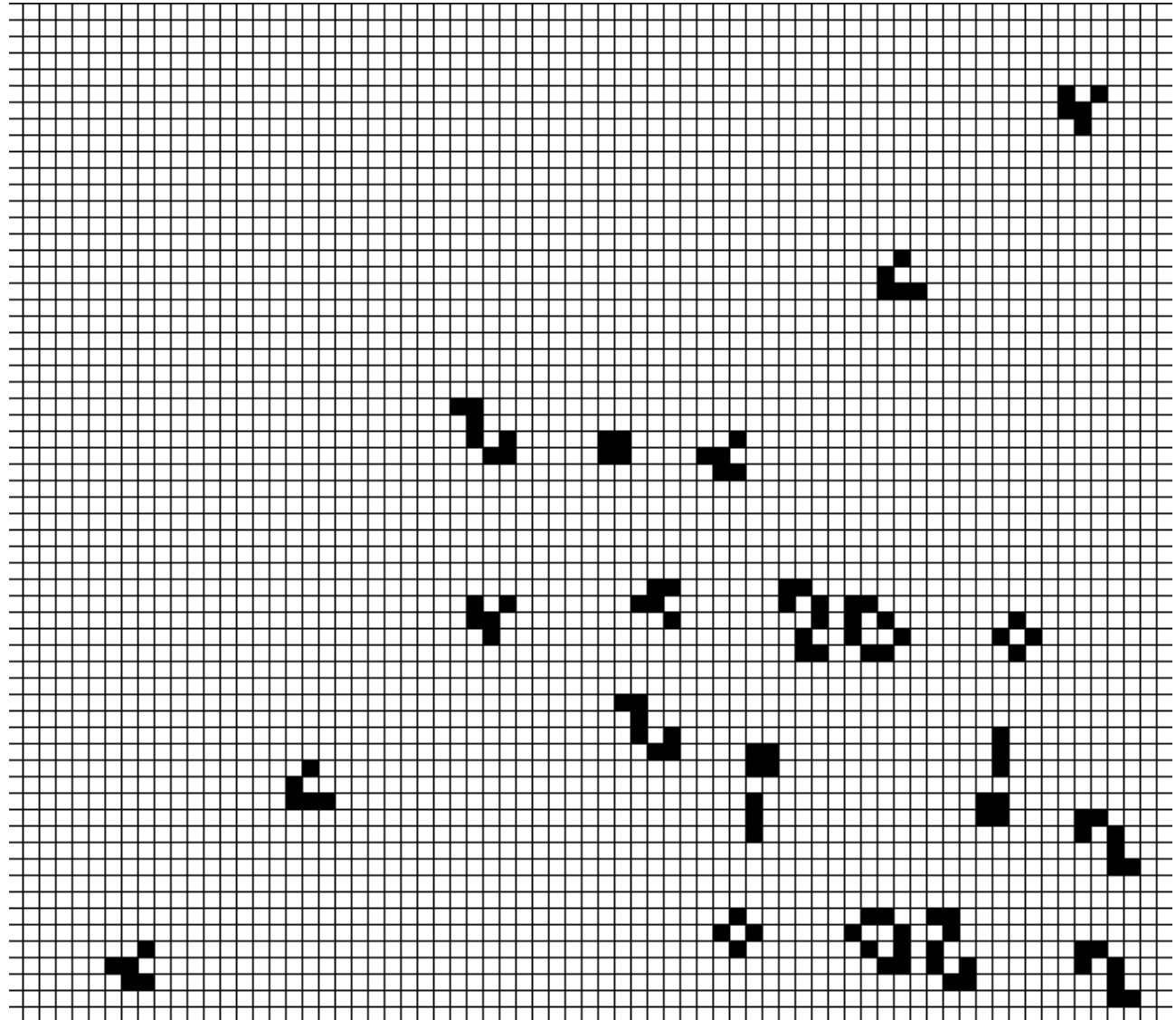
Duplicator :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

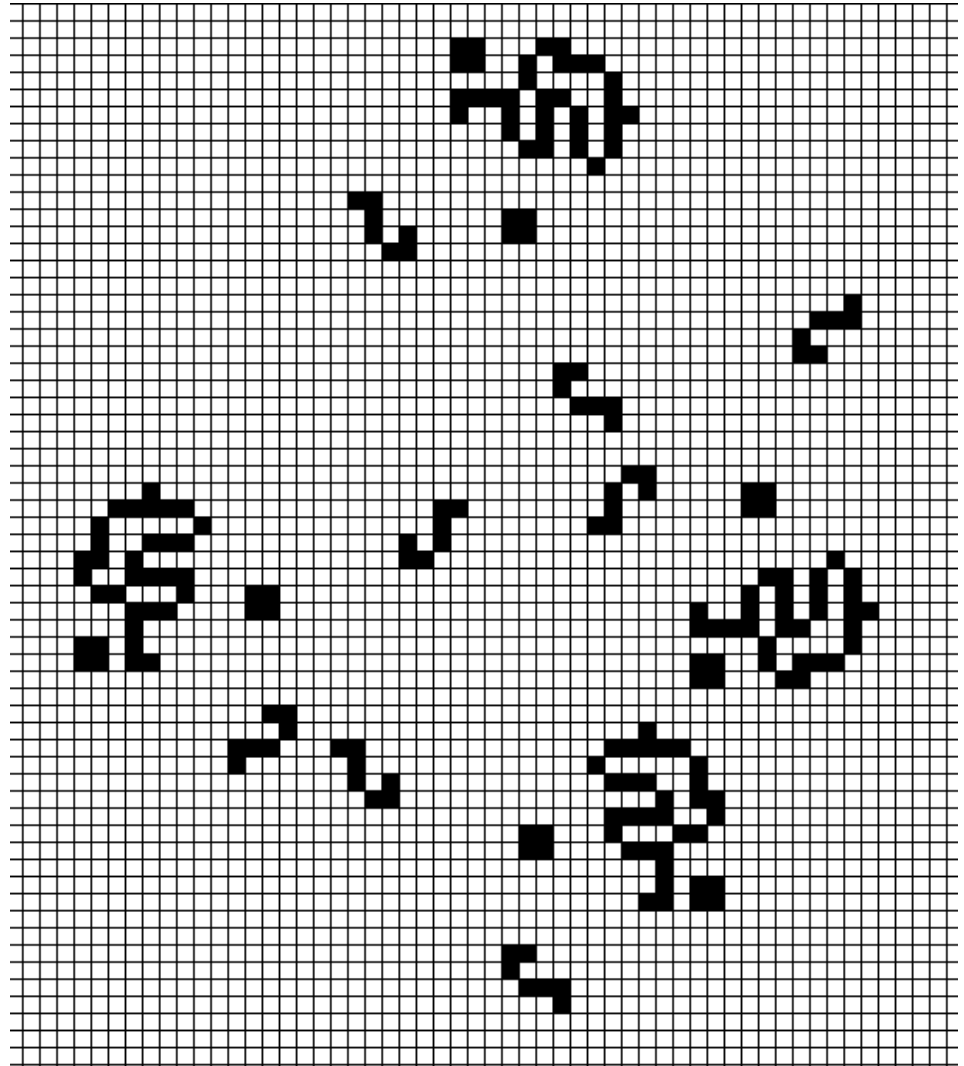
Shifter :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone
Slide :

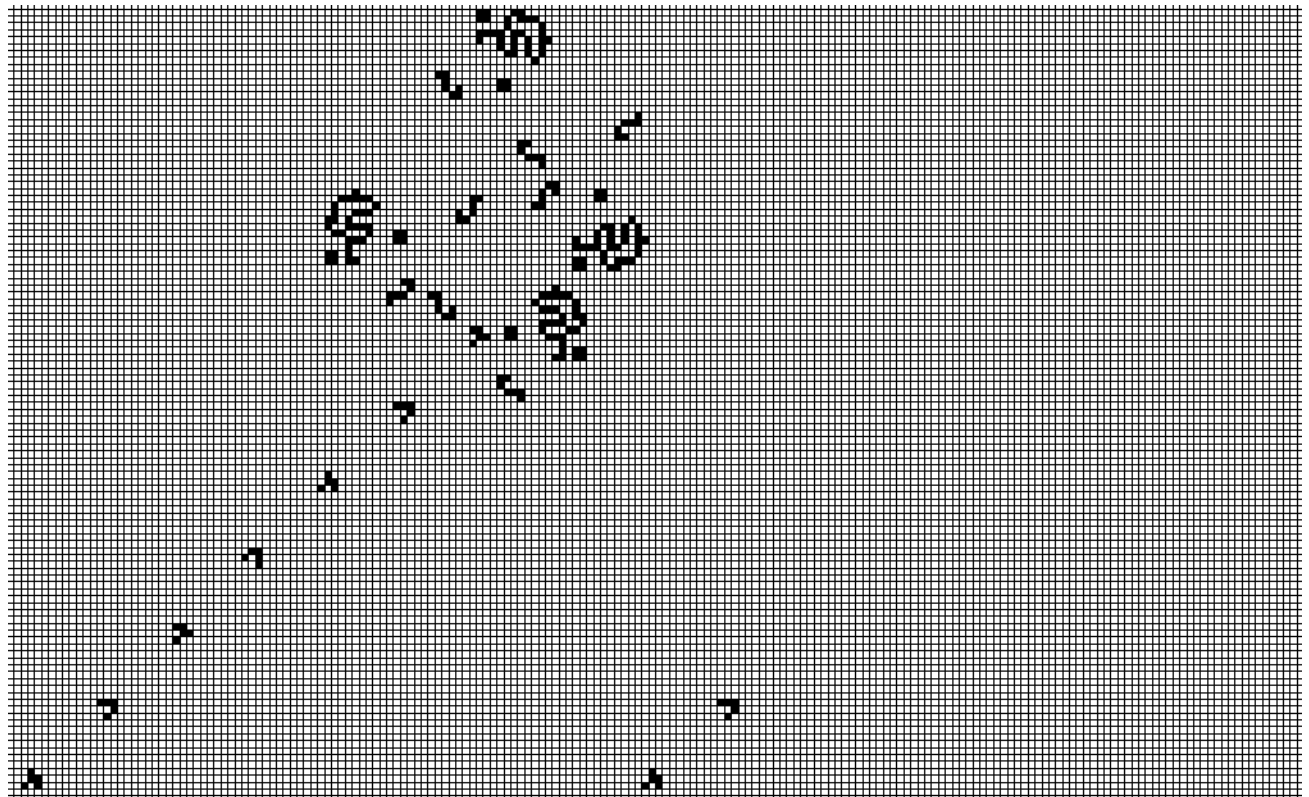


II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

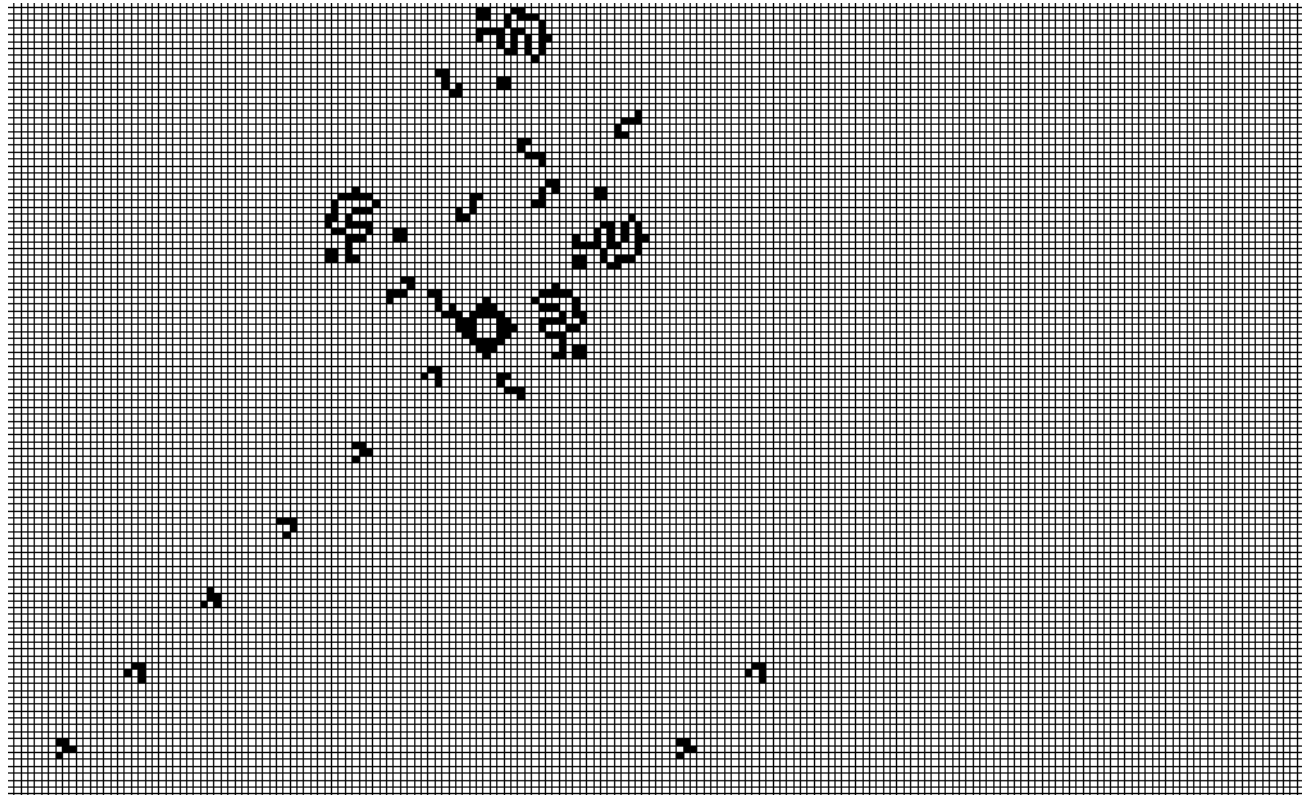
90 cases horizontales



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

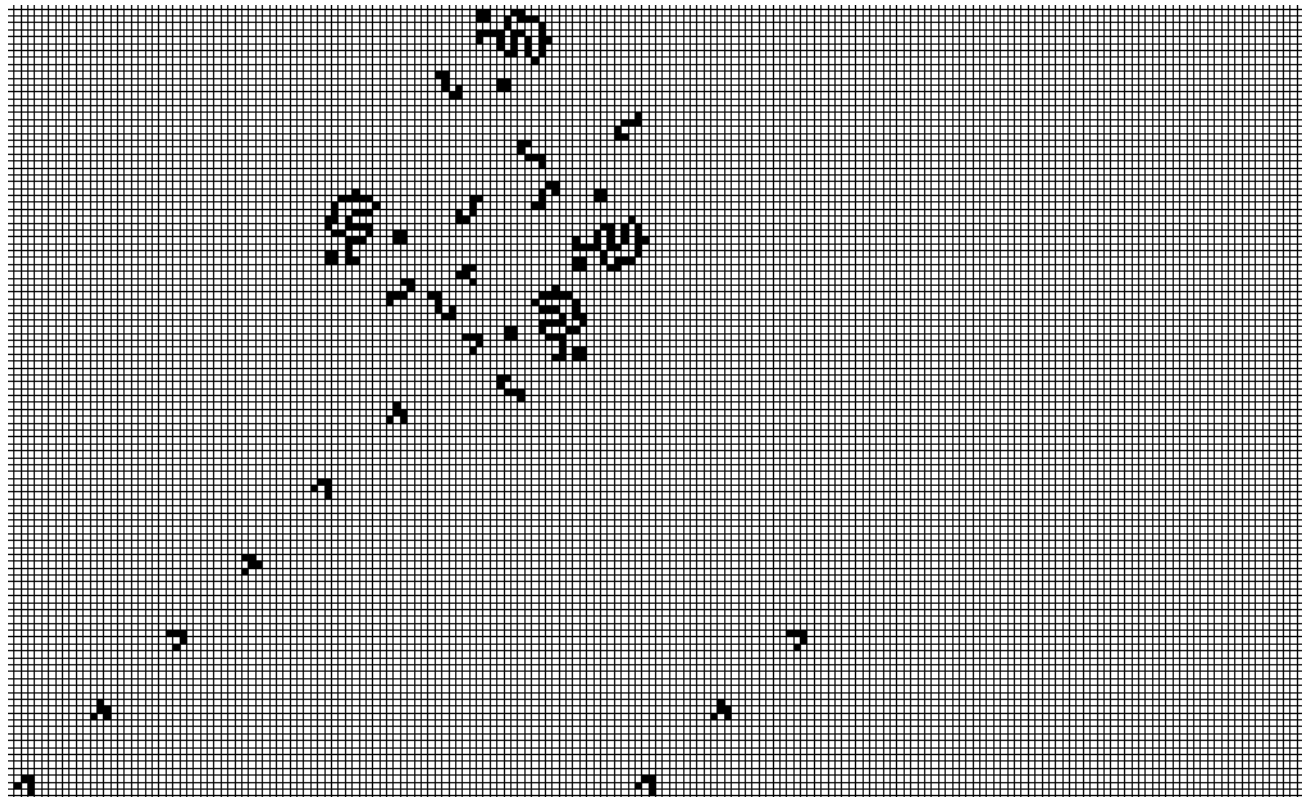
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

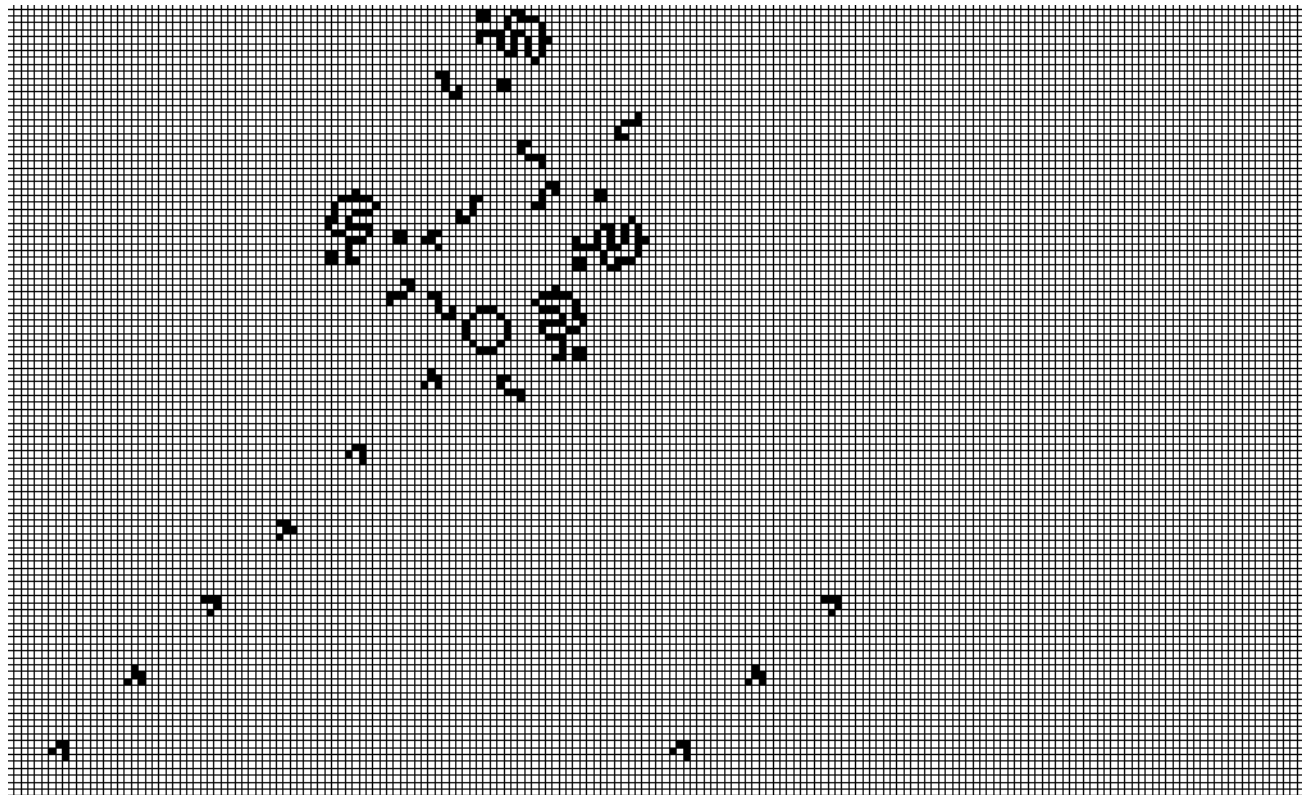
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

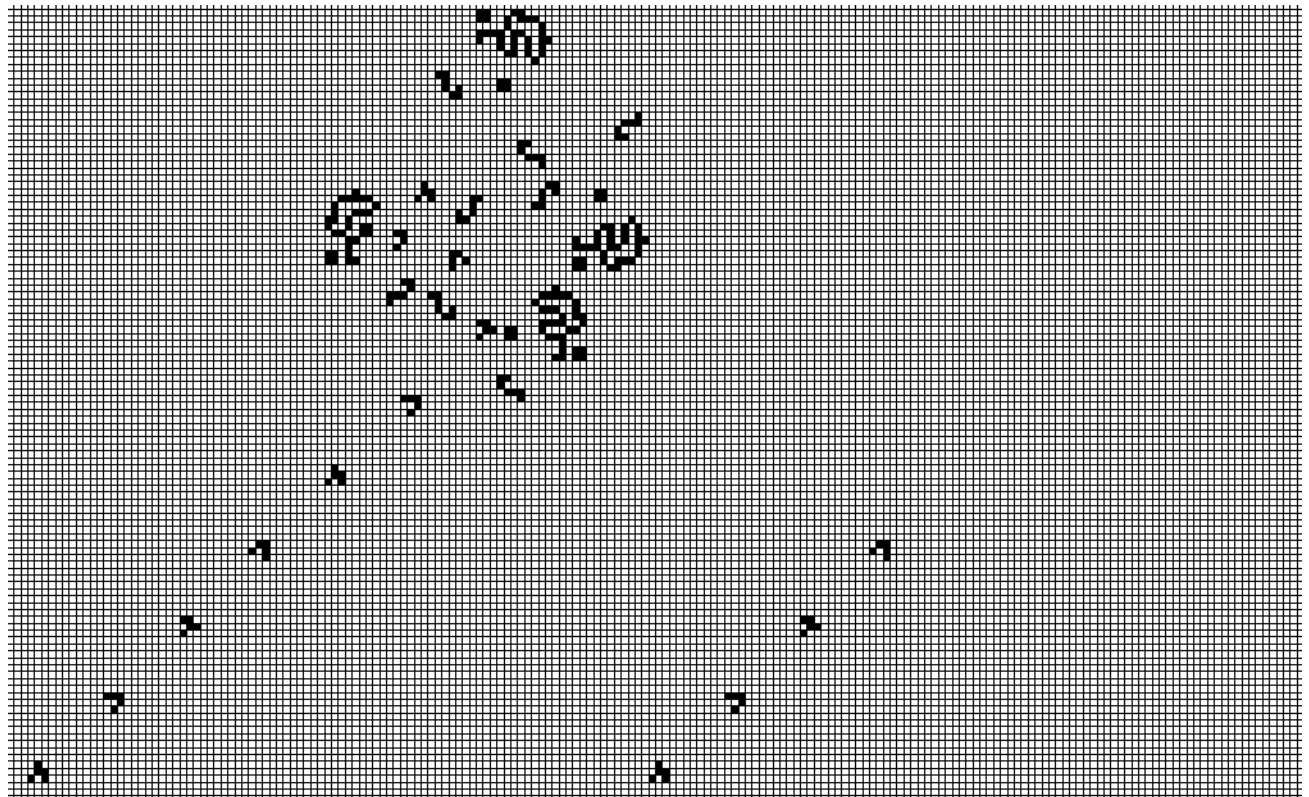
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

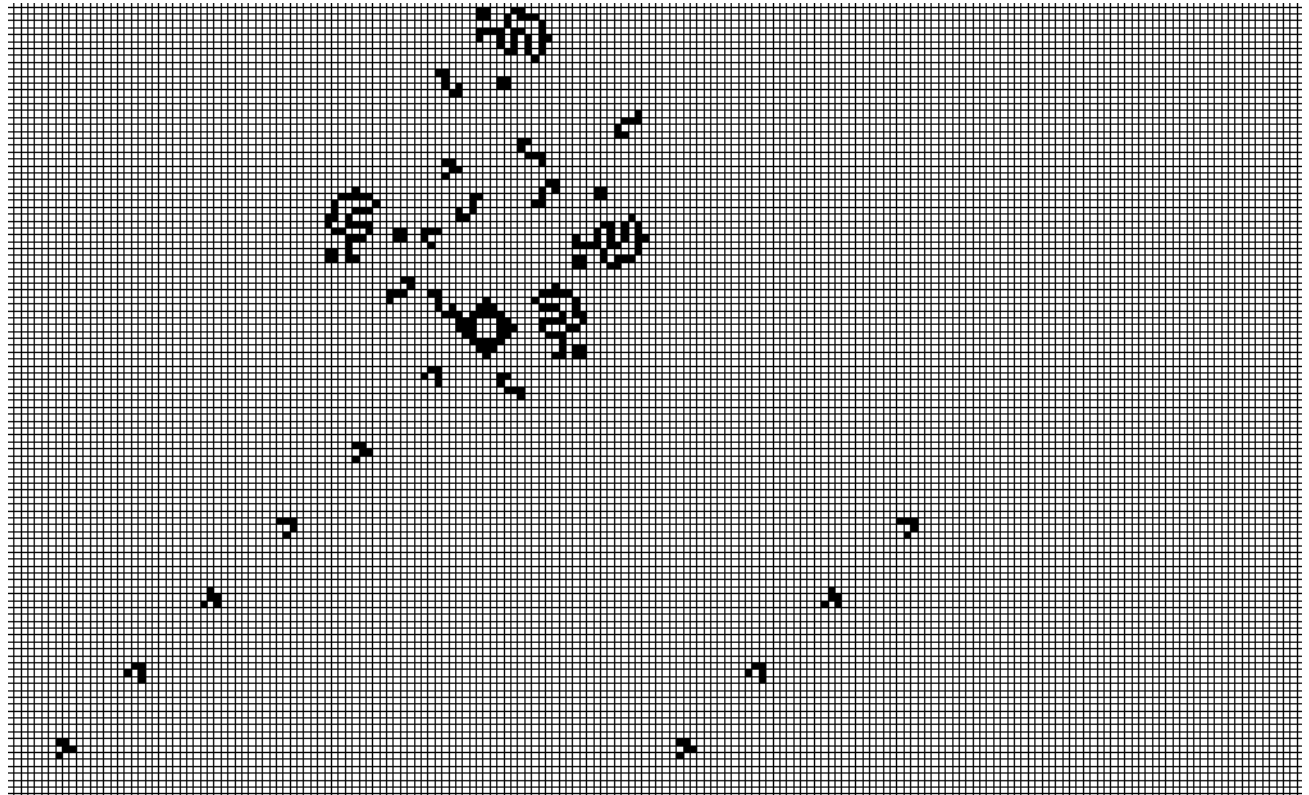
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

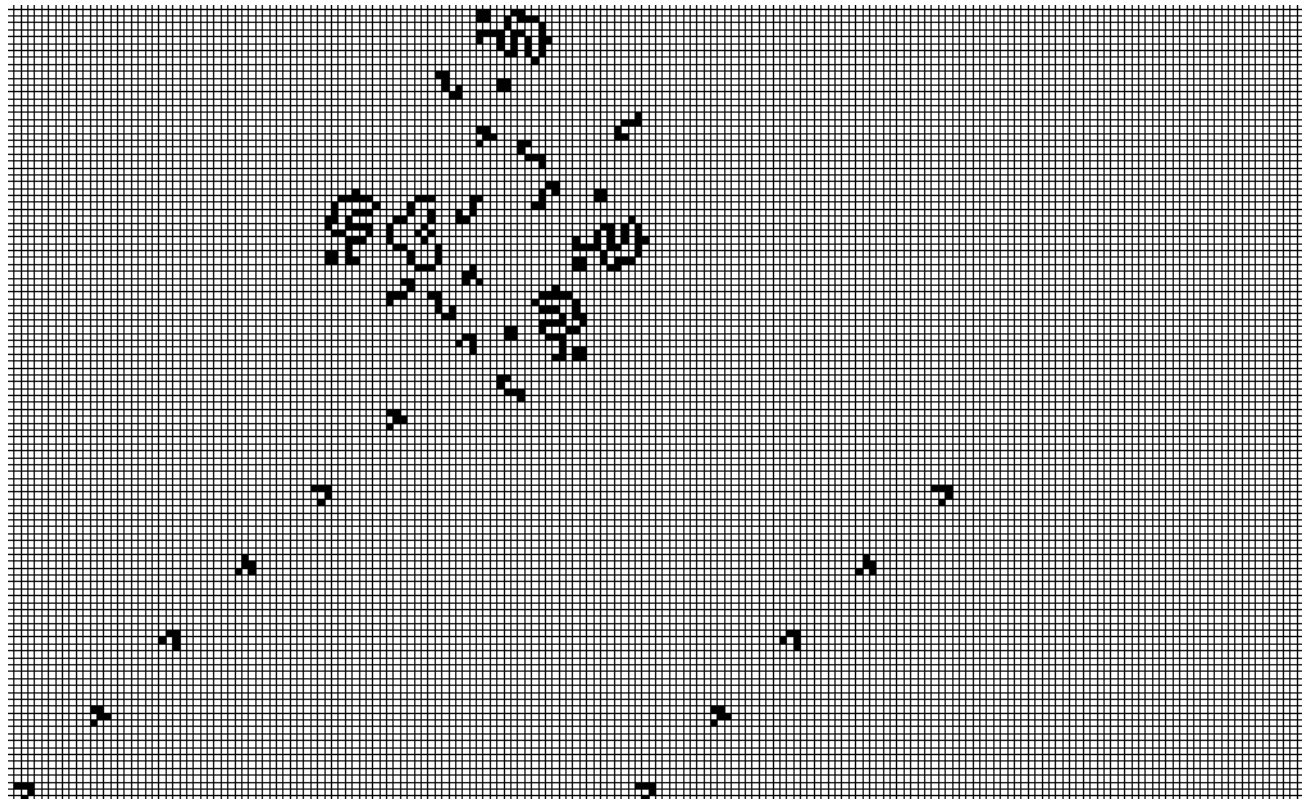
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

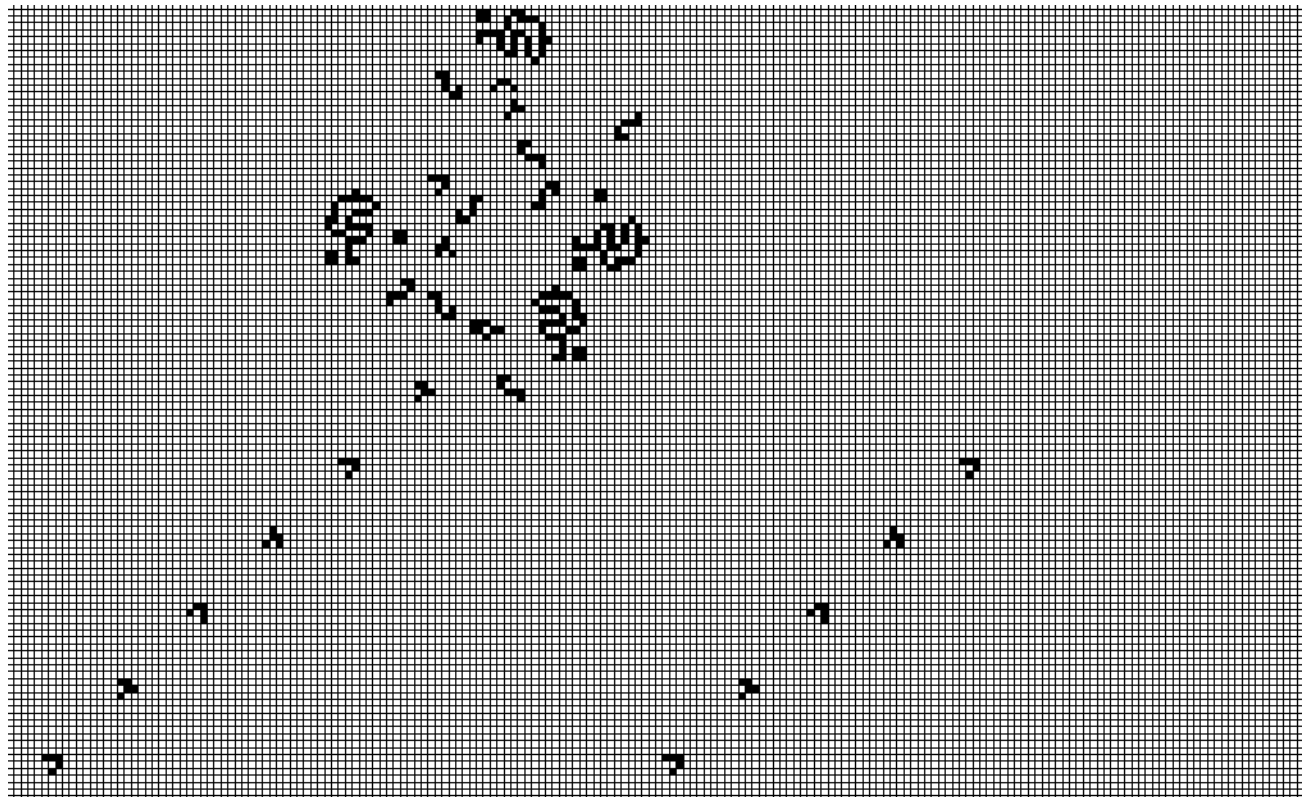
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

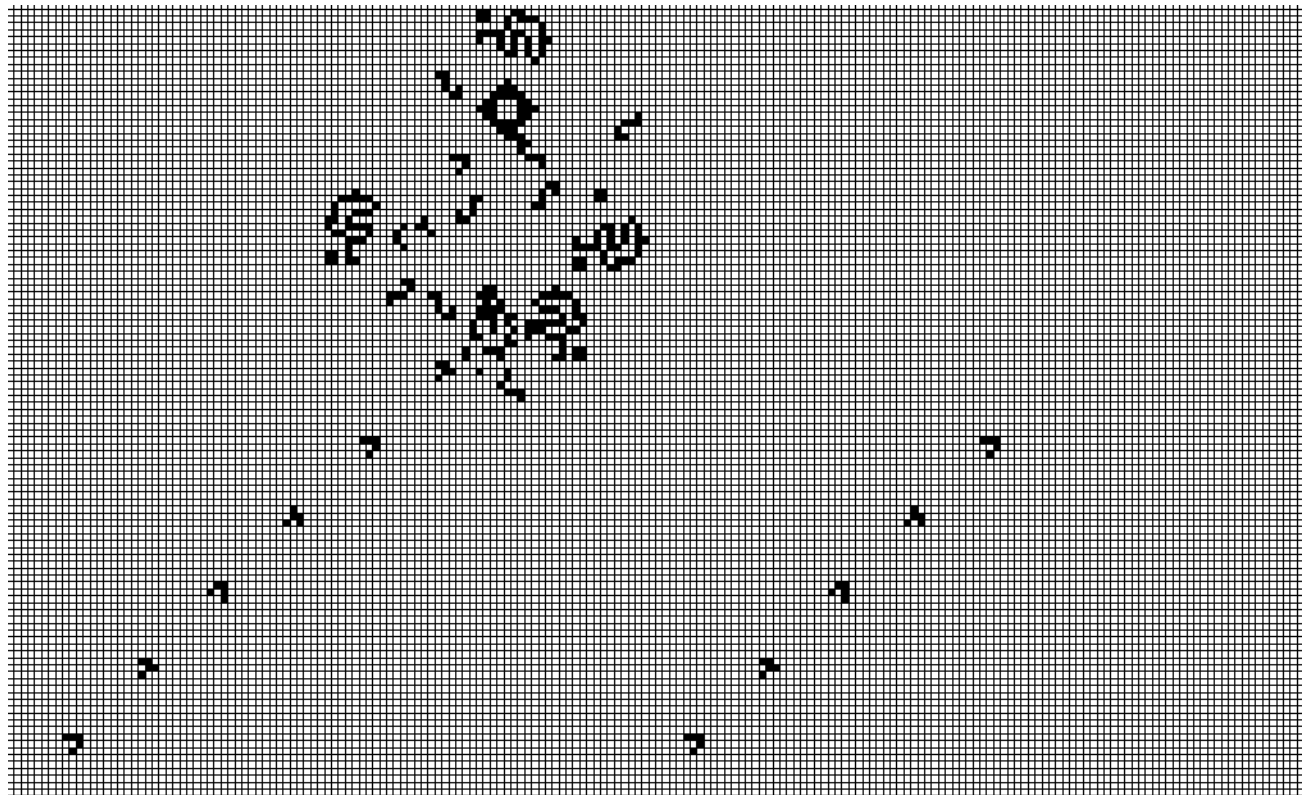
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

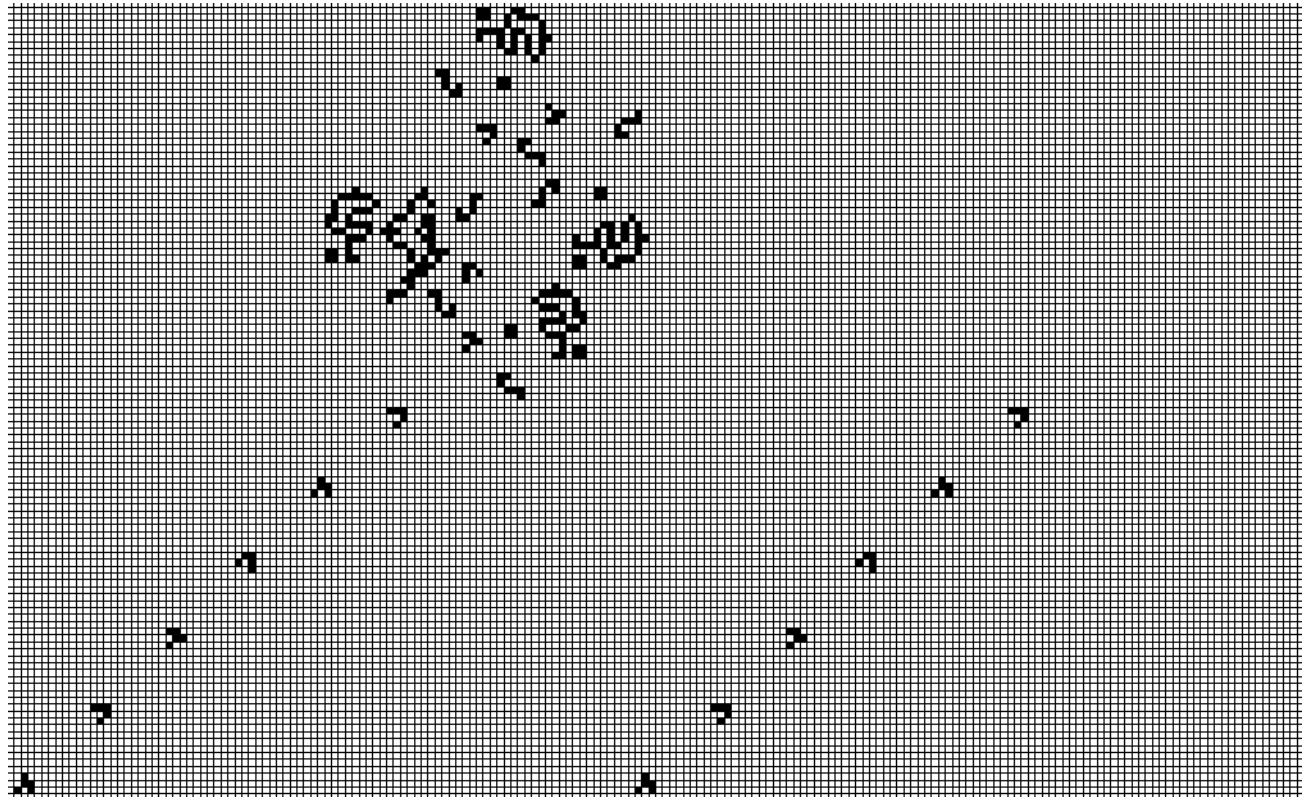
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

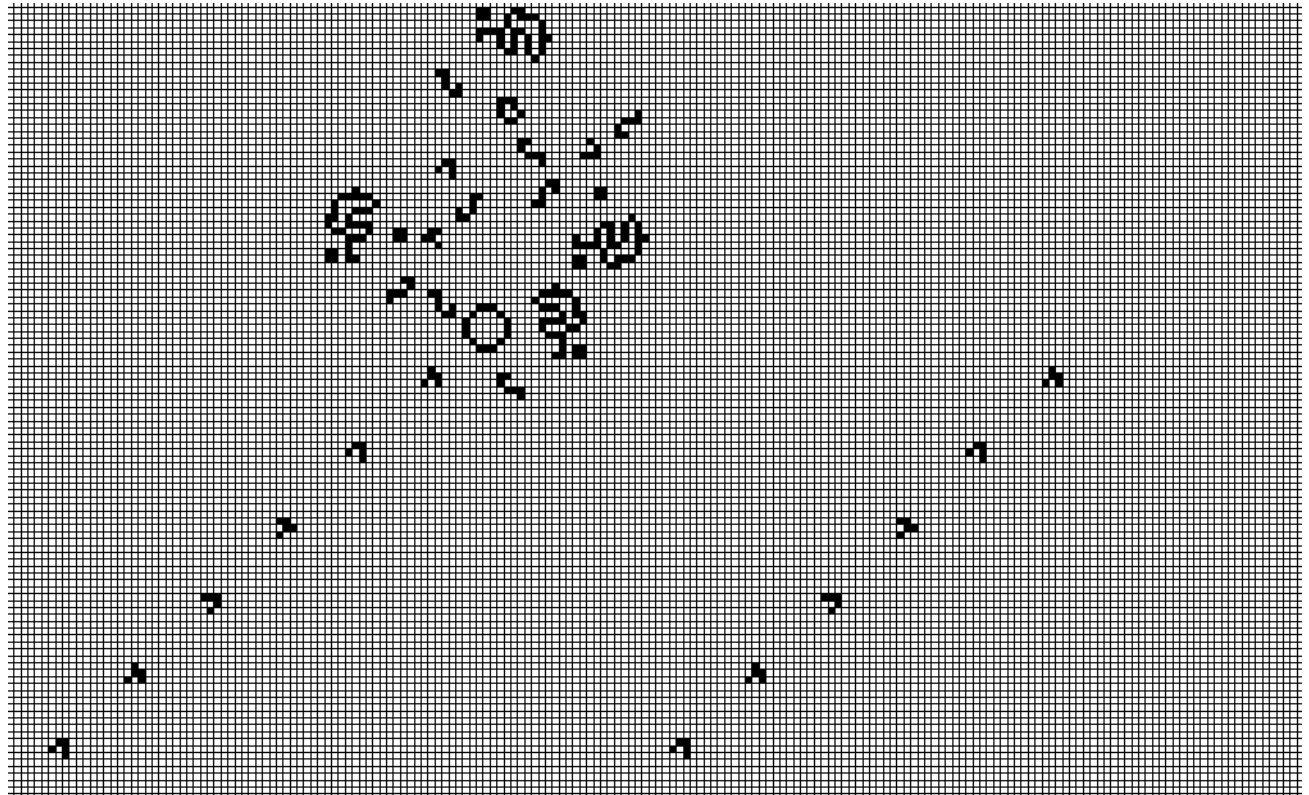
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

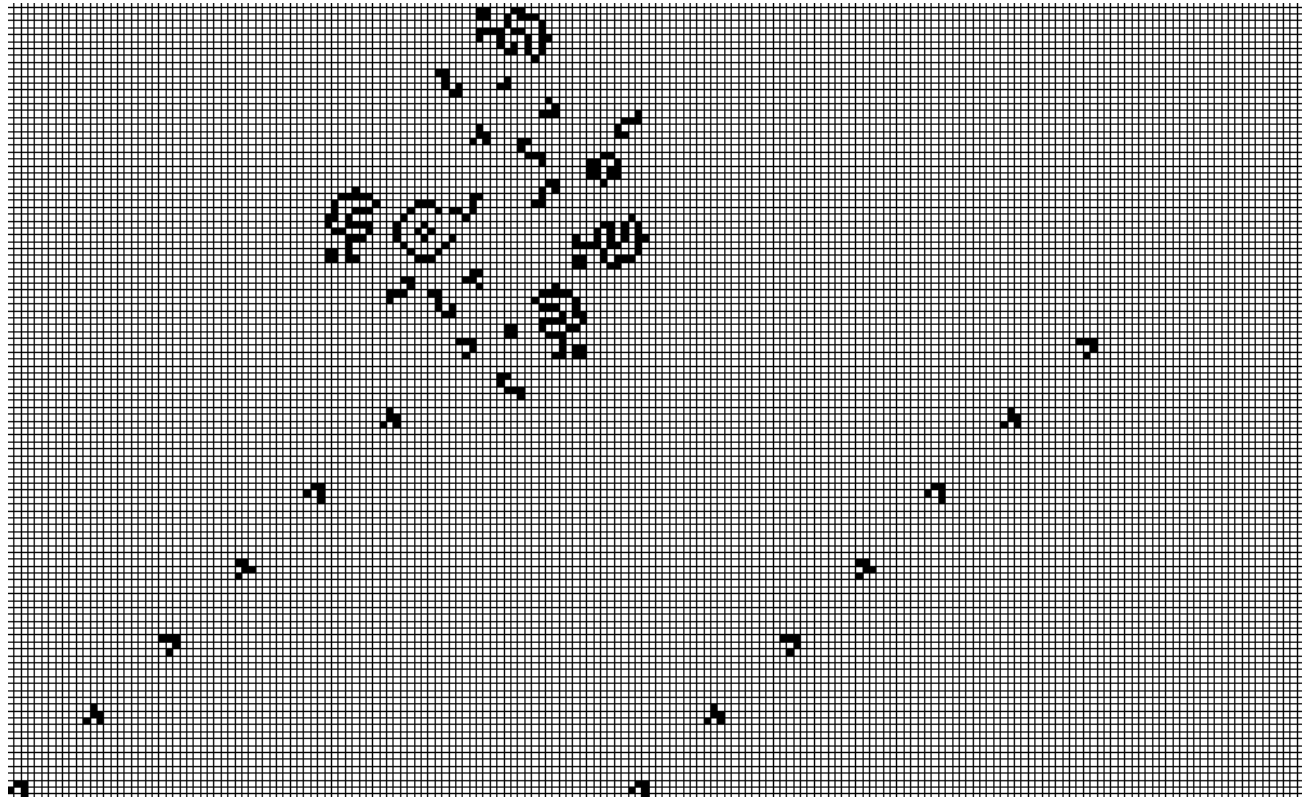
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

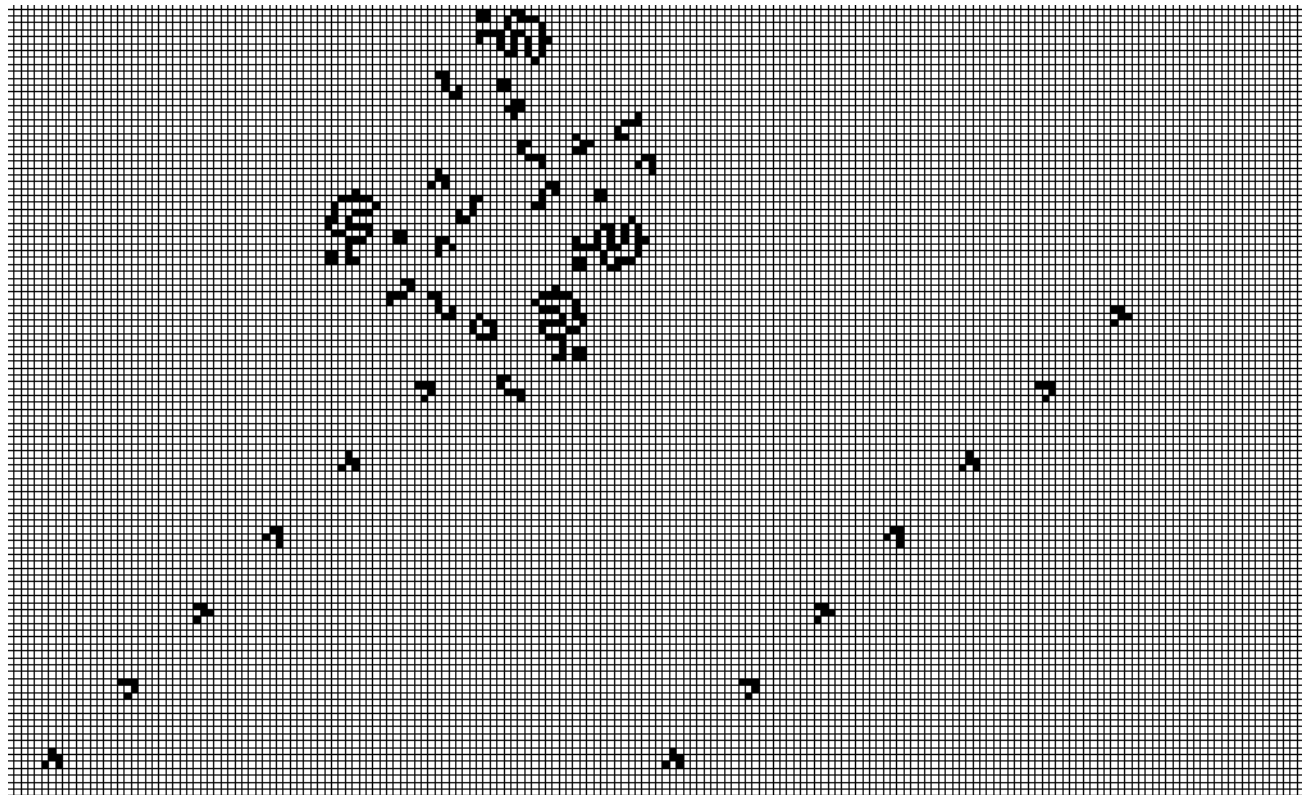
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

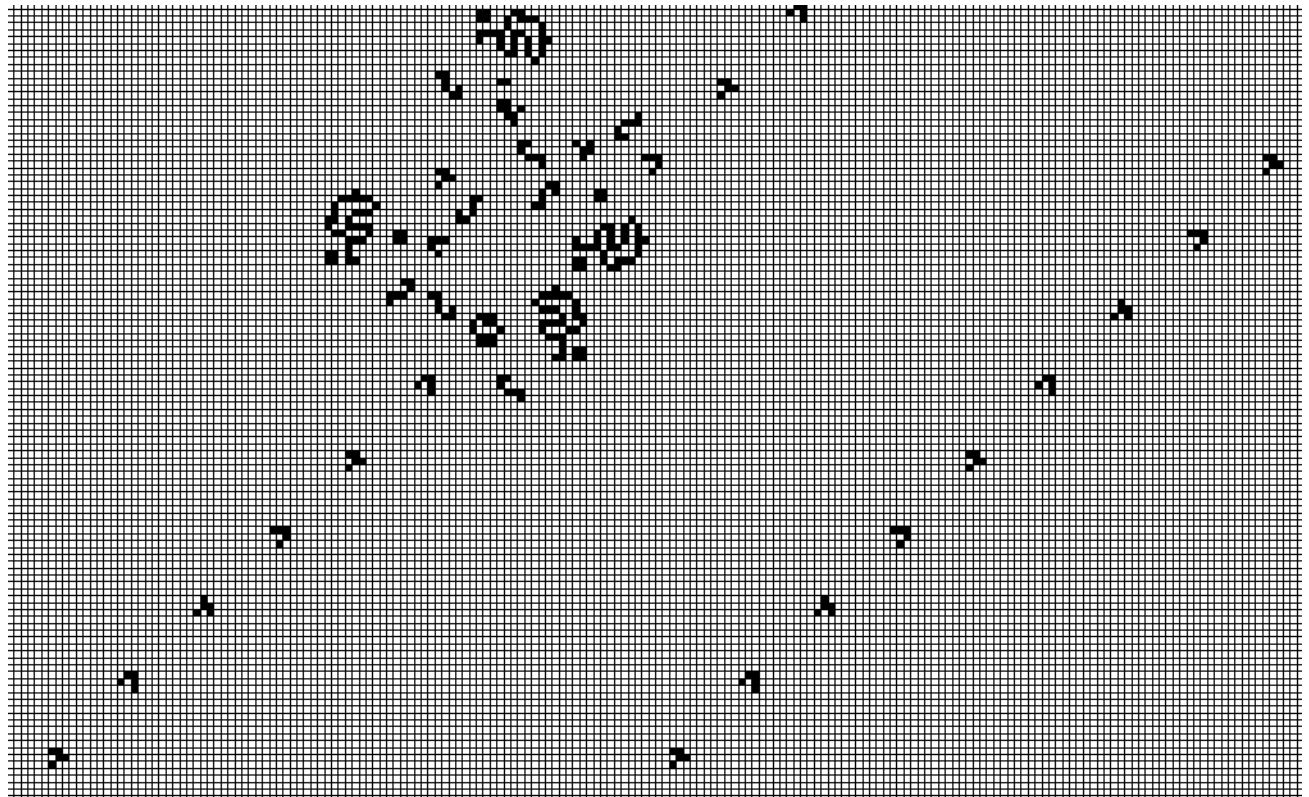
Trombone
Slide :



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone
Slide :



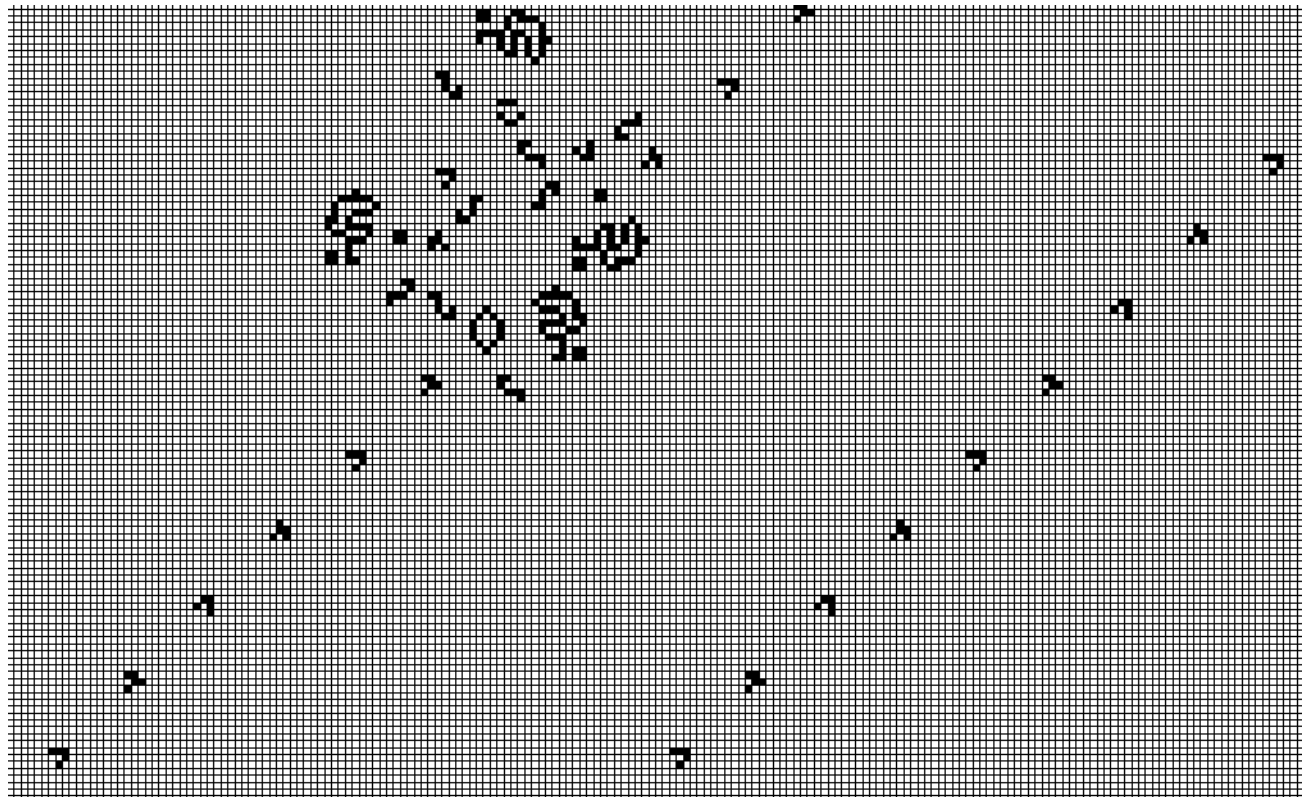
II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

Soit $d \in \mathbb{Z}/43\mathbb{Z}$ le nombre de génération qui sépare le faisceau sortant (43) du faisceau entrant (43)

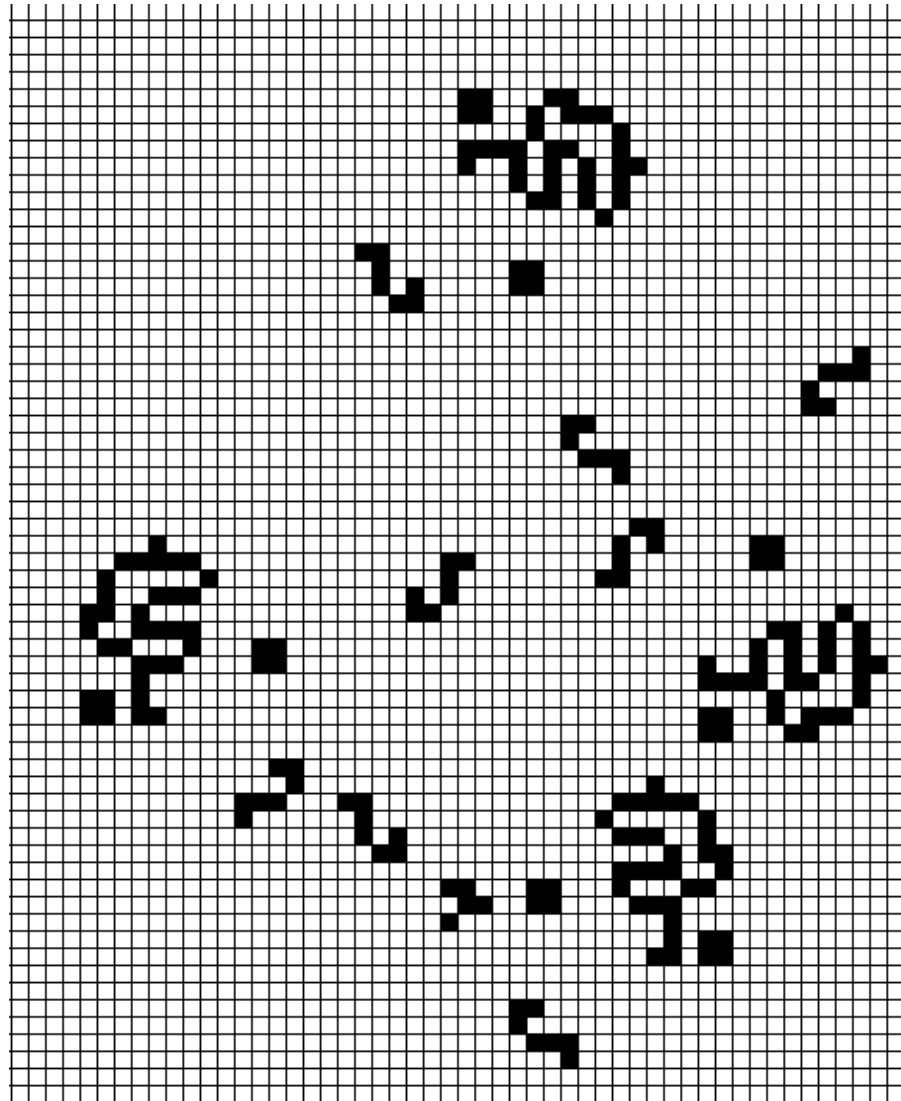
Ici : $d \equiv -1 [43]$



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone
Slide :



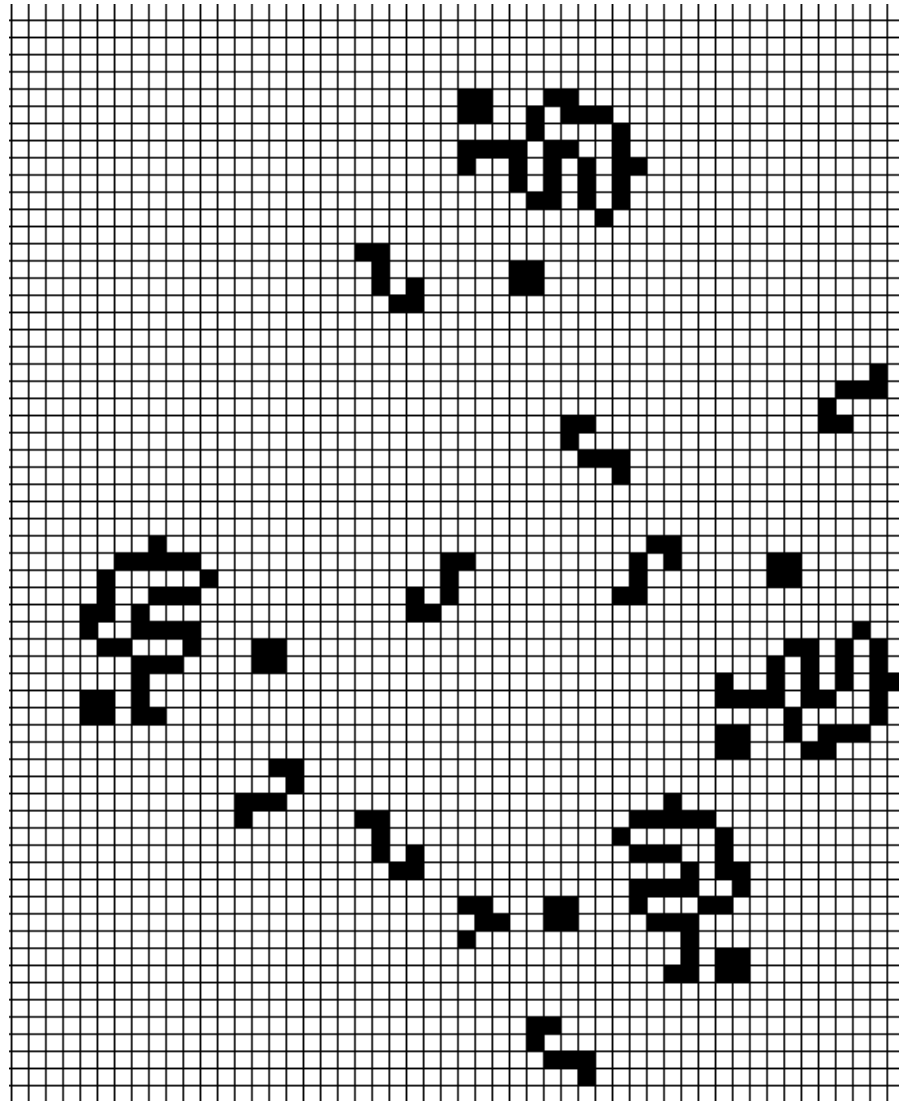
II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

+8 Delai

$d \equiv 7 [43]$



II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

À chaque déplacement de cette partie du Trombone Slide, d augmente de 8 alors, si on cherche : n le nombre de fois que l'on a besoin de déplacer cette partie pour que l'on ait un décalage d :

$$d \equiv -1 + 8n[43] \Leftrightarrow 8n \equiv d + 1[43]$$

On cherche alors à trouver l'inverse de 8 dans $\mathbb{Z} / 43 \mathbb{Z}$, on applique alors l'algorithme d'euclide étendu pour résoudre : $8k \equiv 1[43]$

II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

$$43 = 5 \cdot 8 + 3$$

$$8 = 2 \cdot 3 + 2$$

$$3 = 1 \cdot 2 + 1$$

$$2 = 2 \cdot 1 + 0$$

$$1 = 3 - 1 \cdot 2$$

$$1 = 3 - 1(8 - 2 \cdot 3) = 3 \cdot 3 - 1 \cdot 8$$

$$1 = 3 \times (43 - 5 \cdot 8) - 1 \cdot 8 = 3 \cdot 43 - 16 \cdot 8$$

$$\text{Donc, } k \equiv -16 [43] \equiv 27 [43]$$

II. Outils nécessaires et composition du filtre

1. Gestion des faisceaux de gliders (avec l'exemple du Trombone Slide)

Trombone Slide :

$$n \equiv 27(d+1) [43]$$

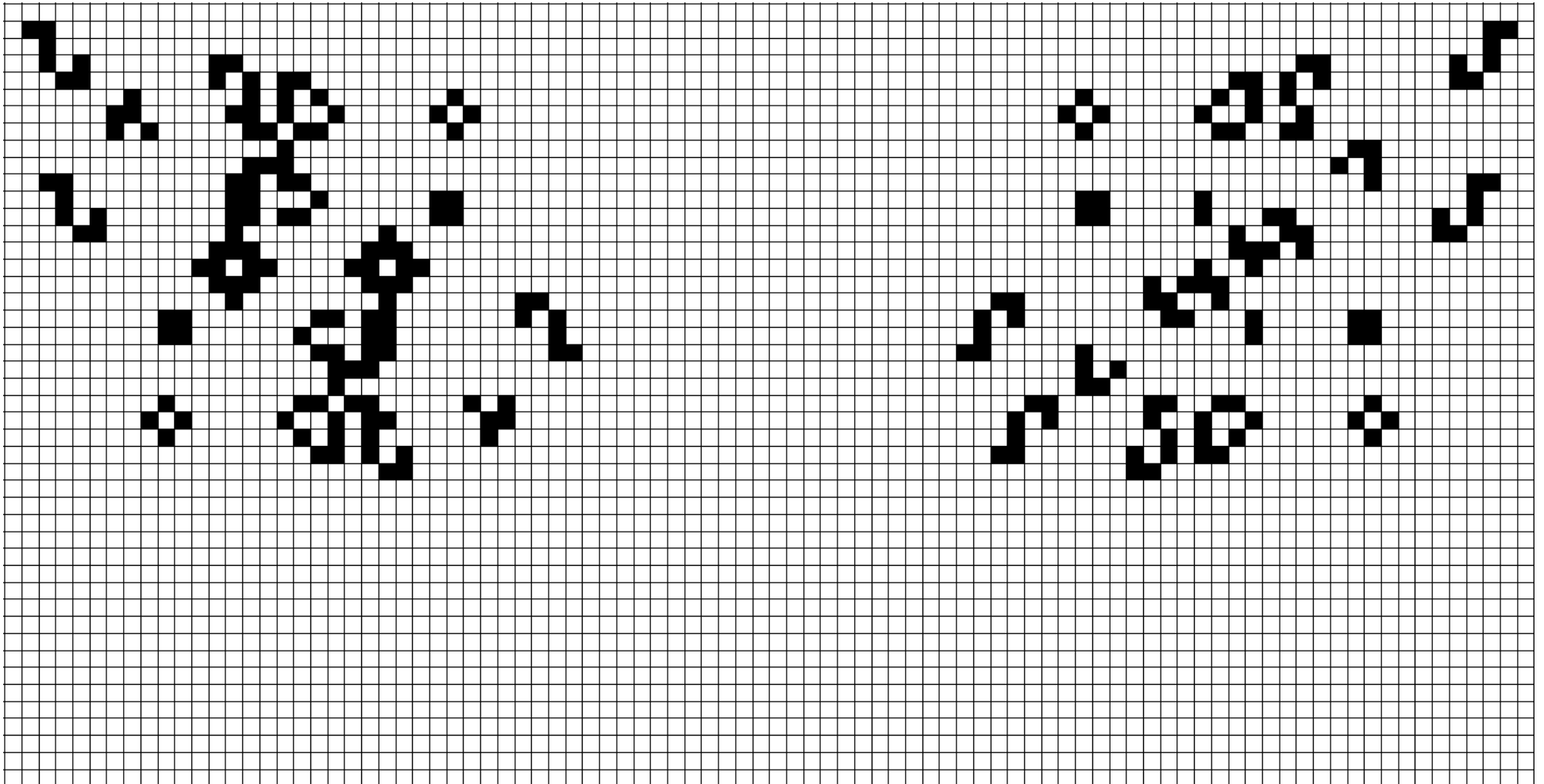
II. Outils nécessaires et composition du filtre

2. Portes logiques

II. Outils nécessaires et composition du filtre

2. Portes logiques

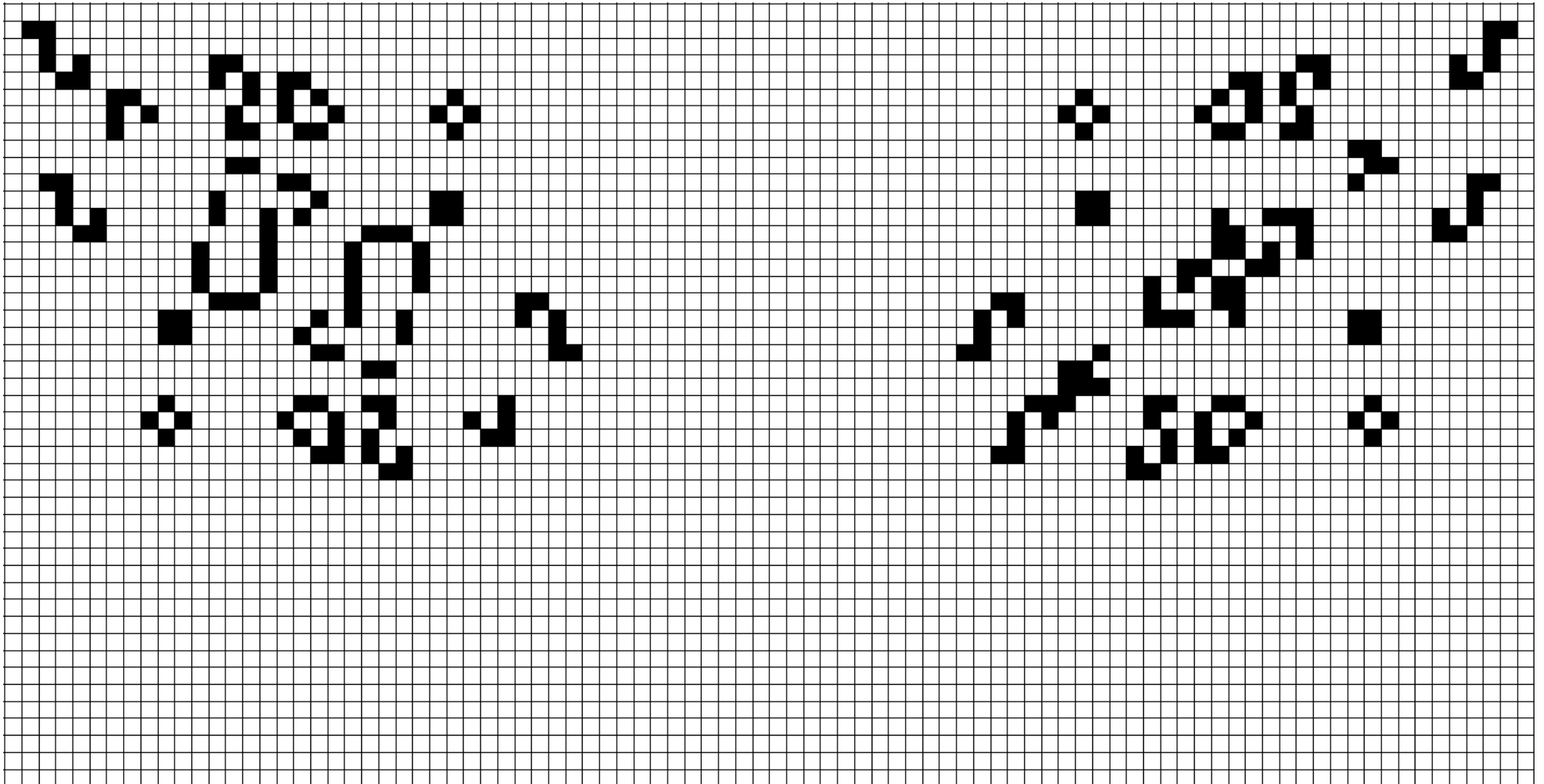
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

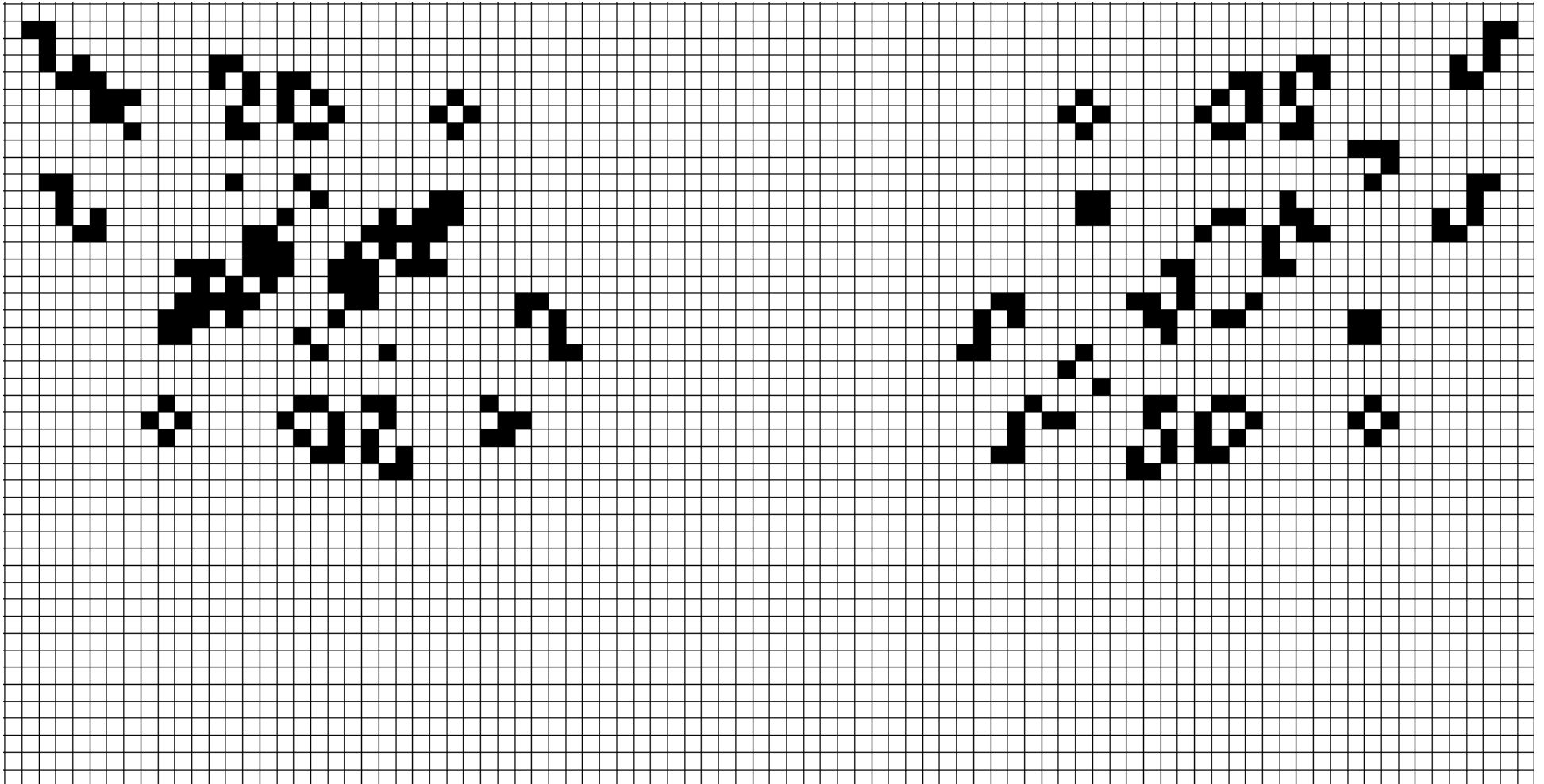
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

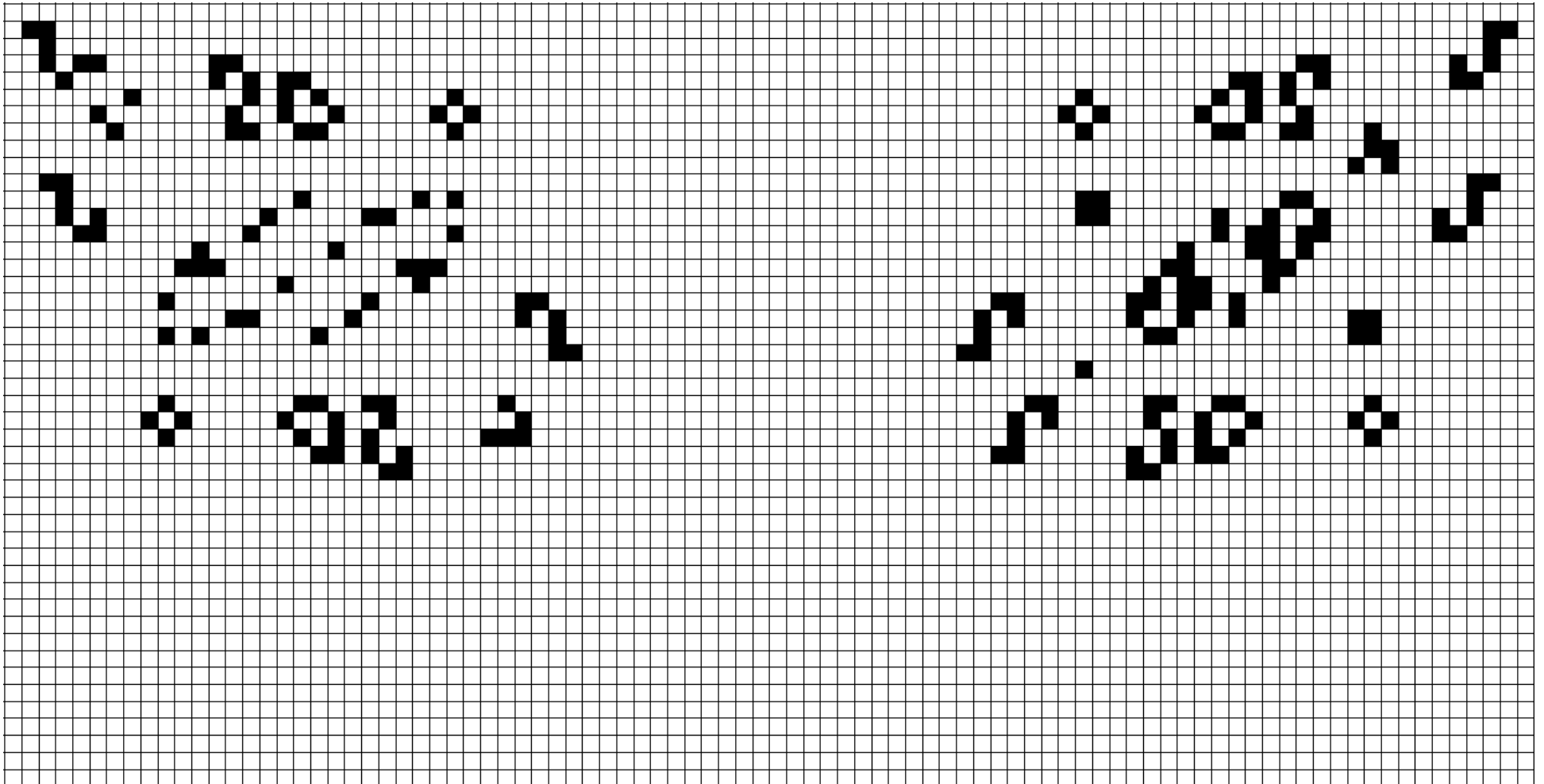
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

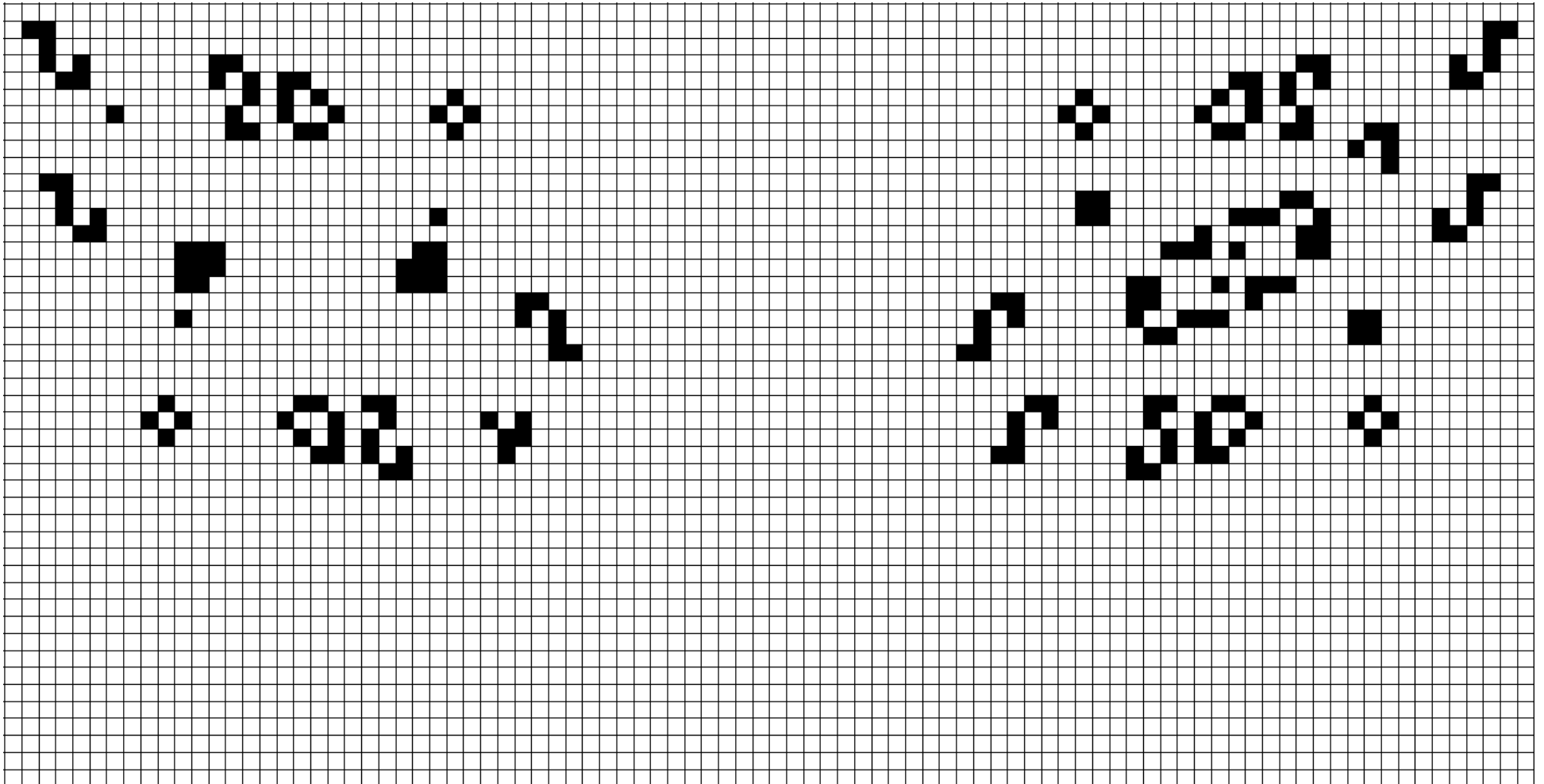
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

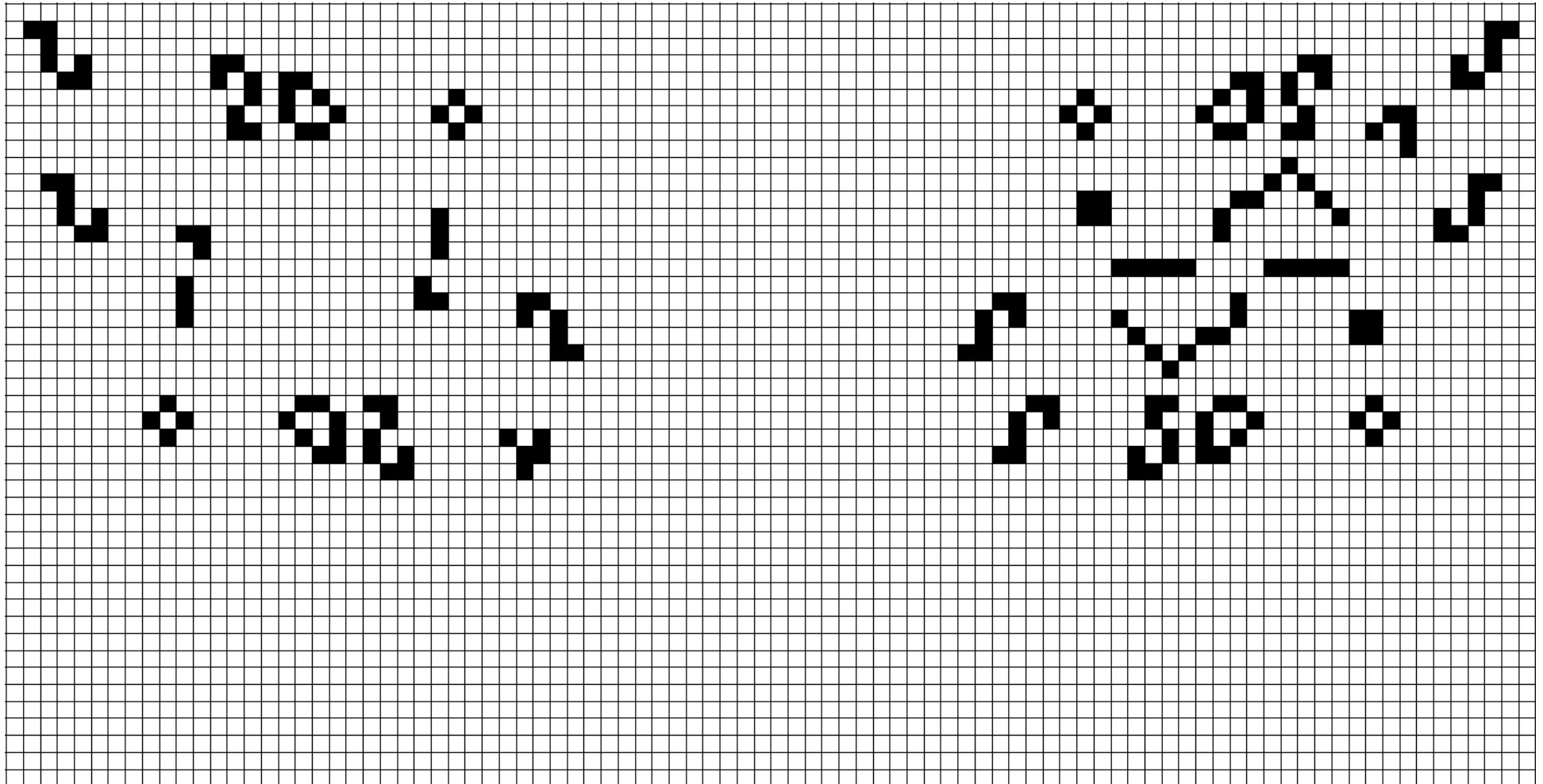
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

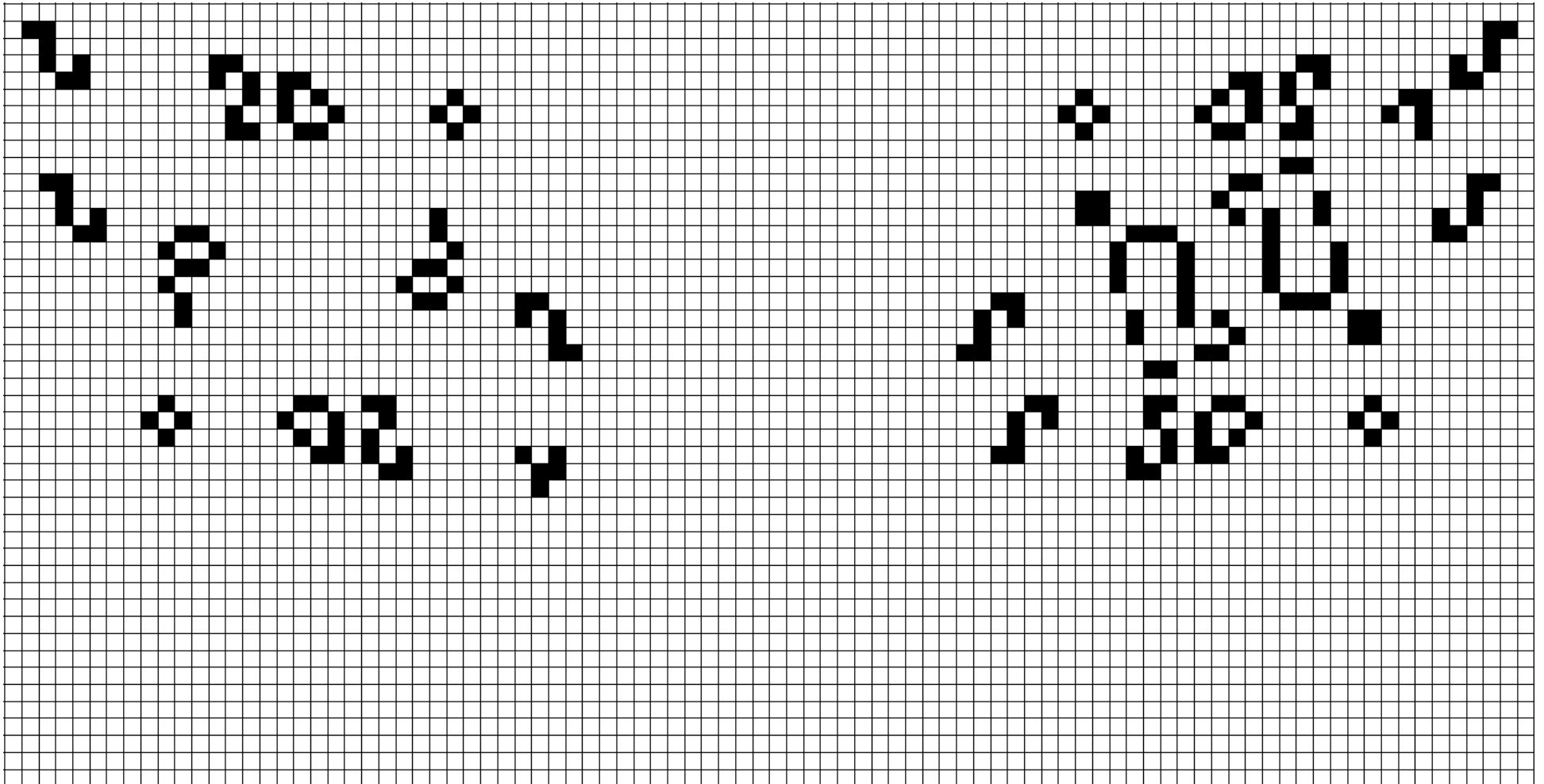
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

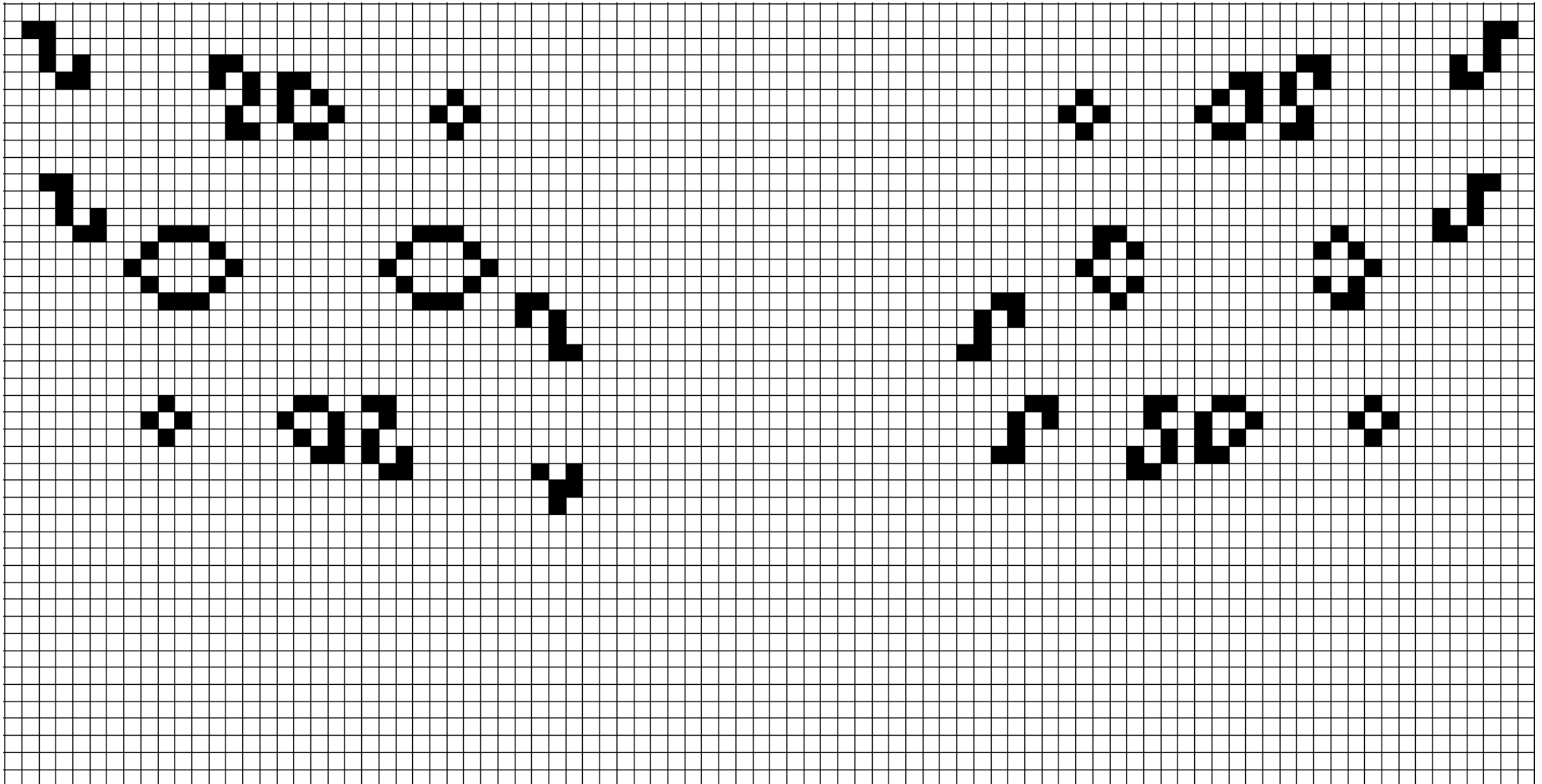
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

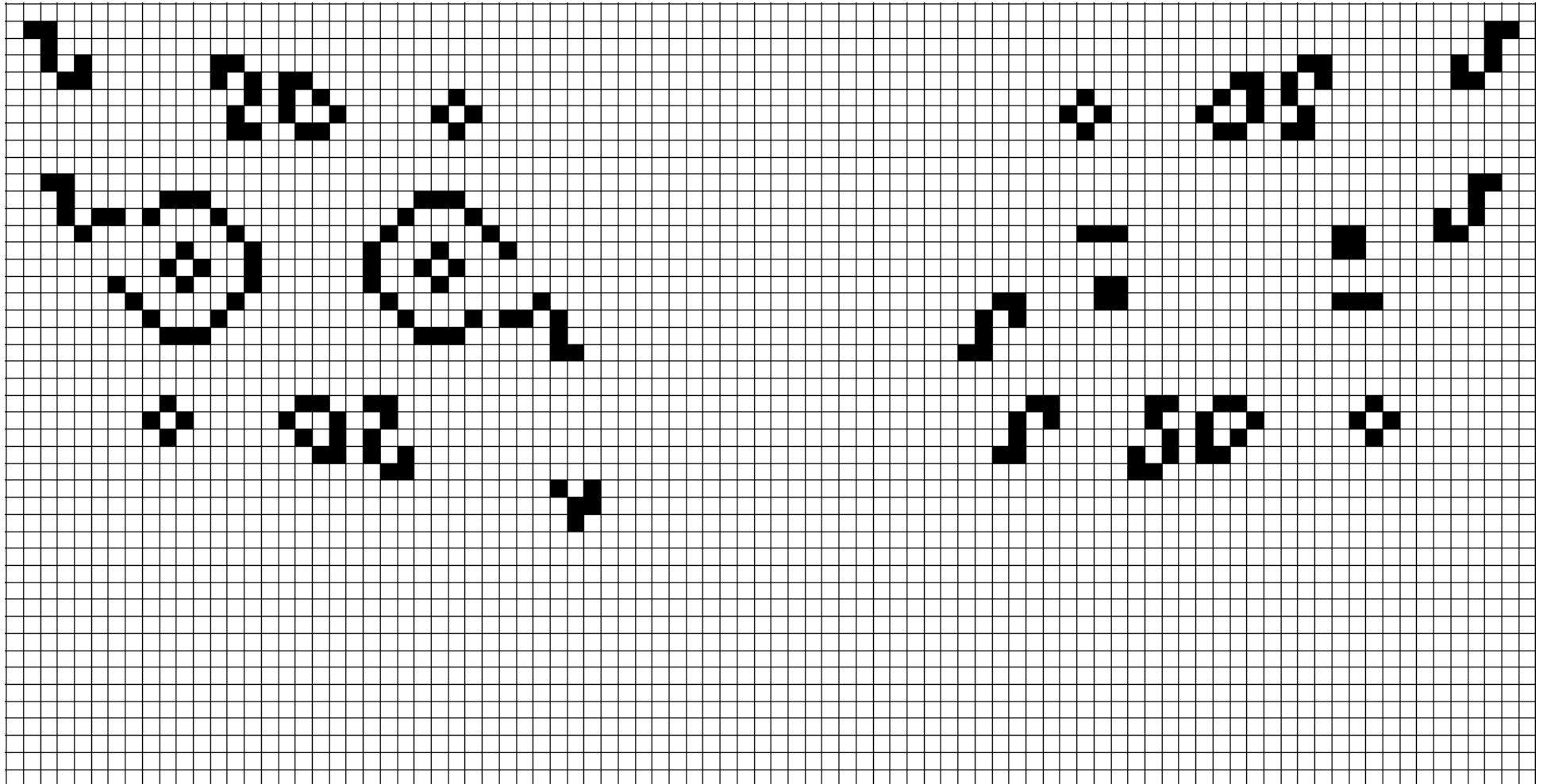
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

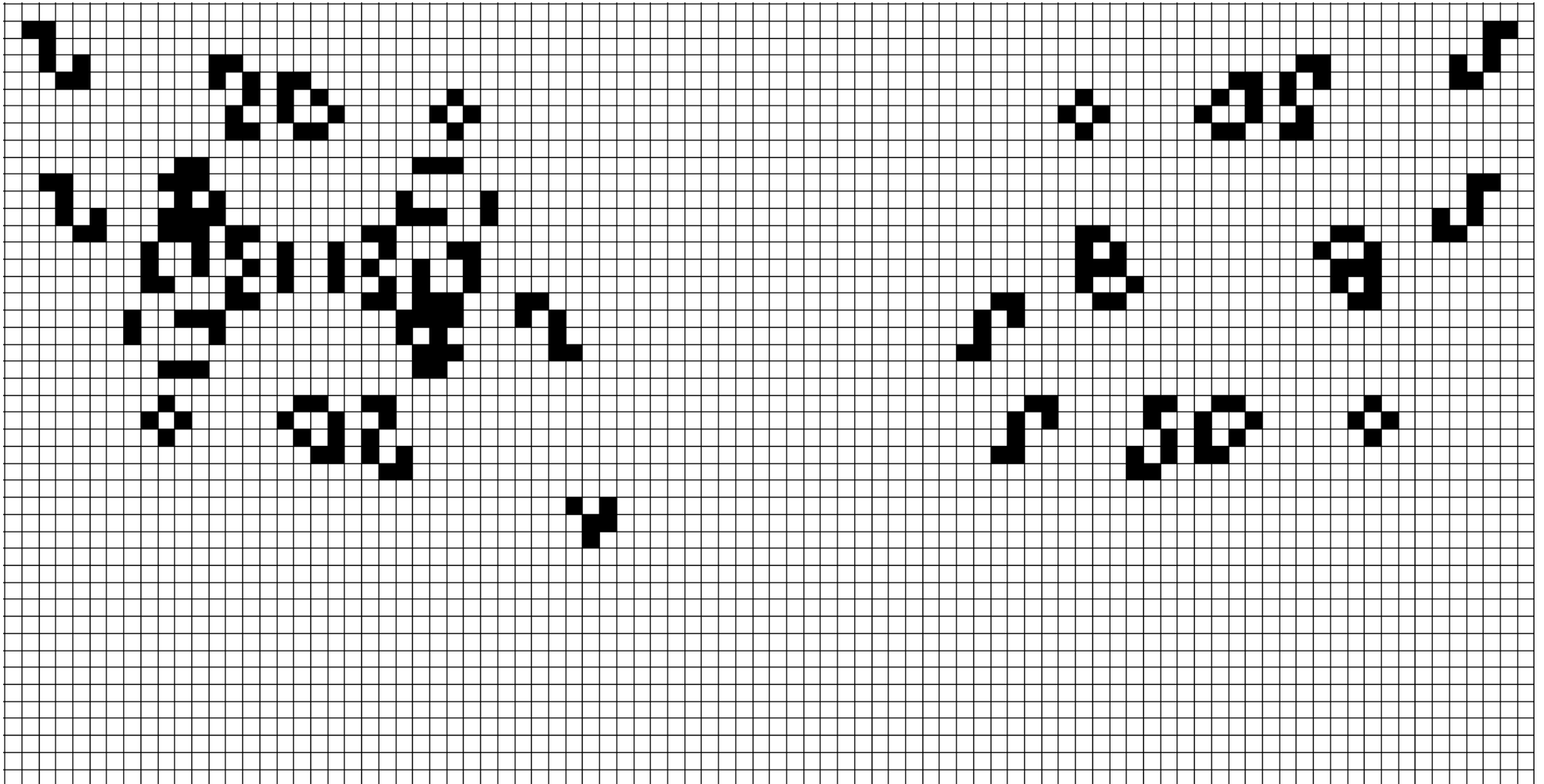
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

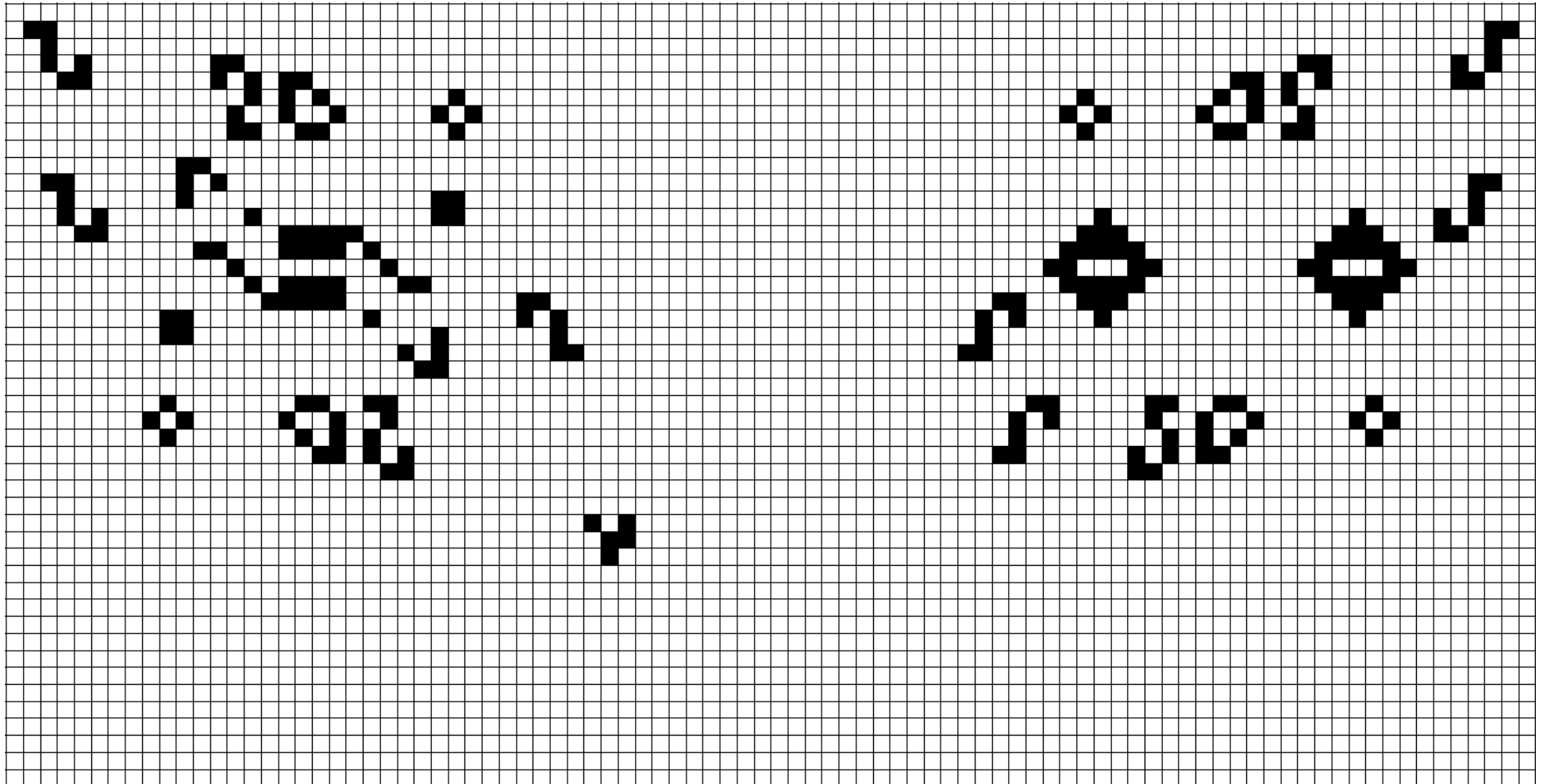
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

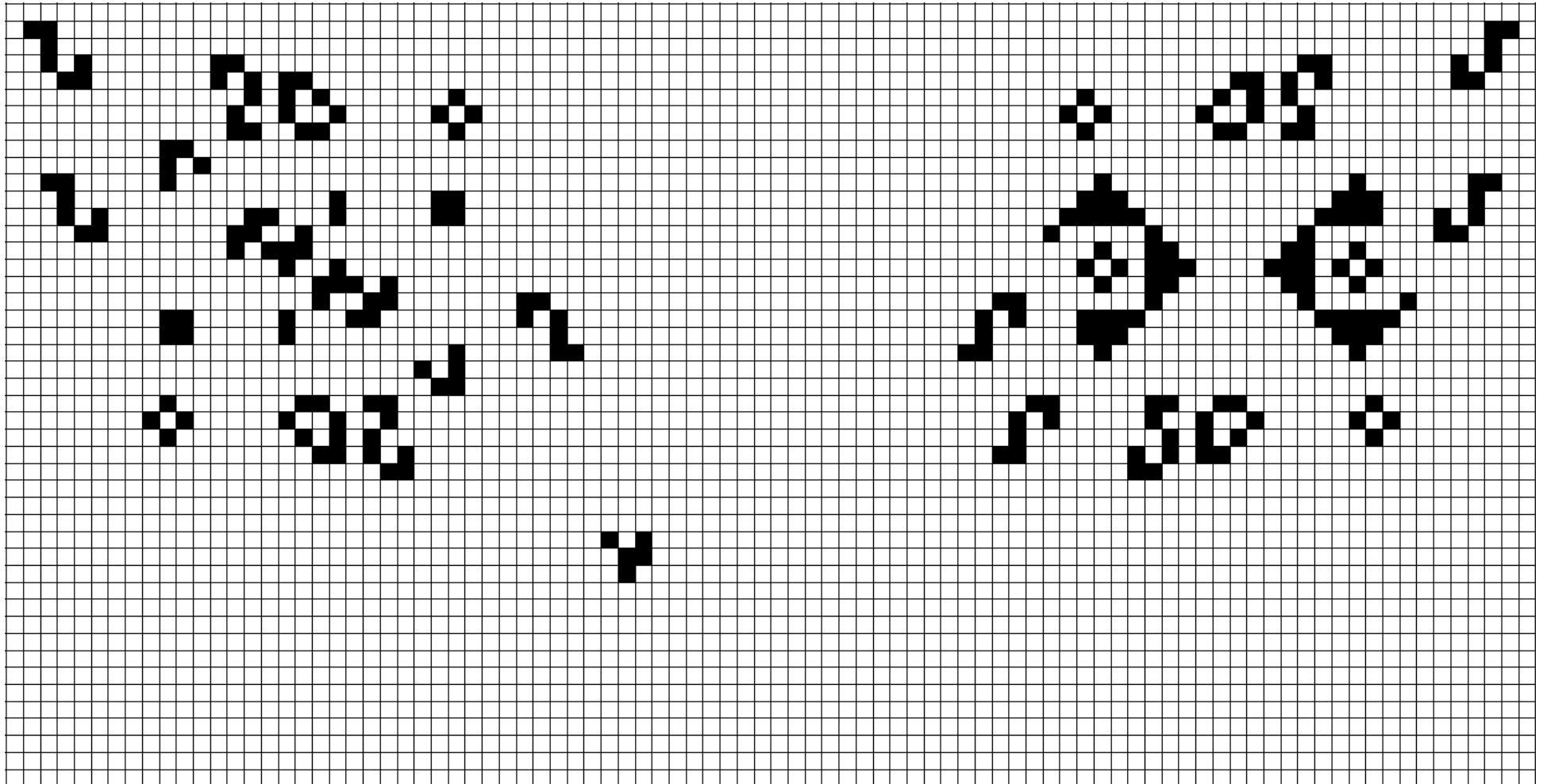
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

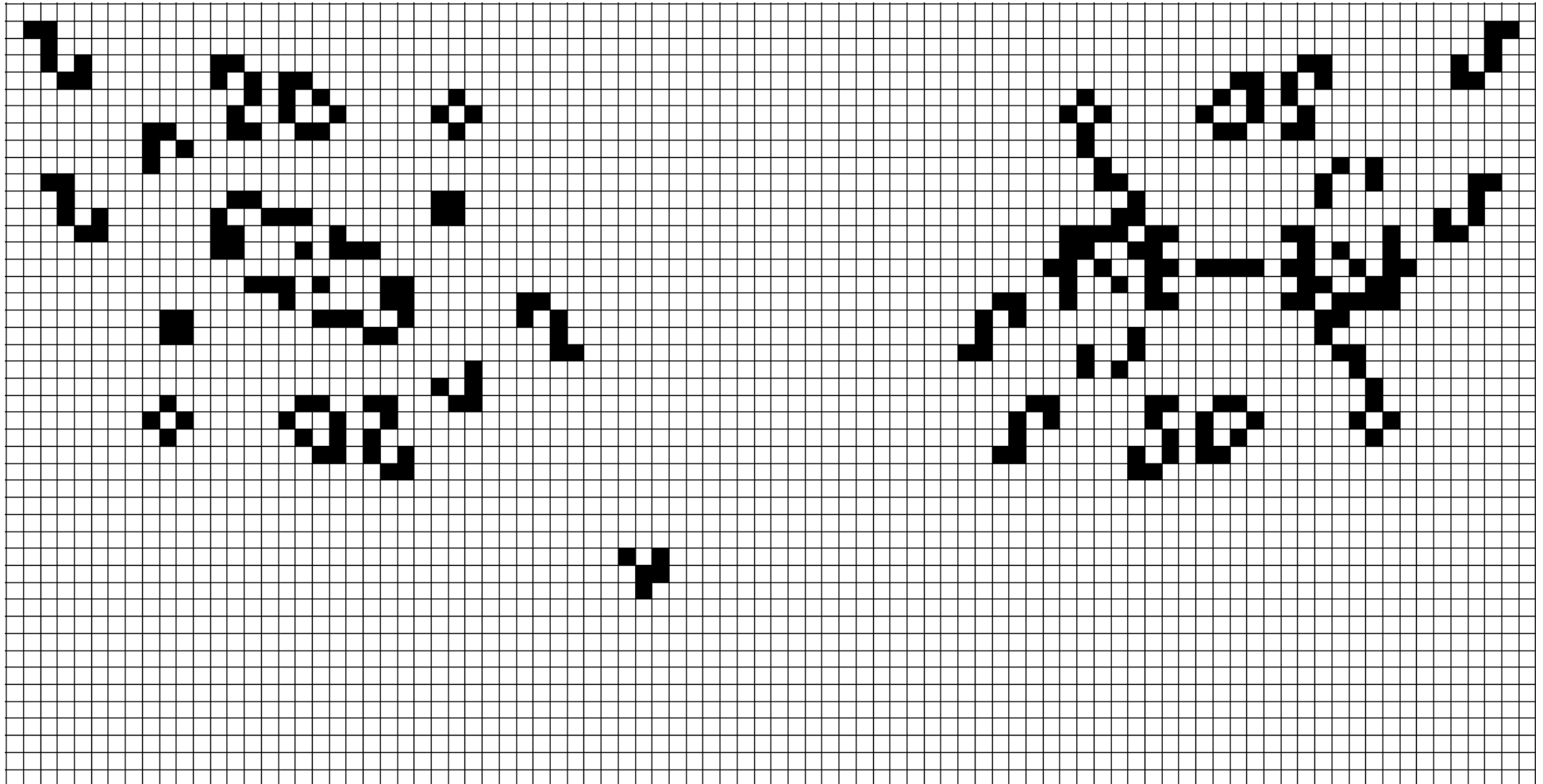
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

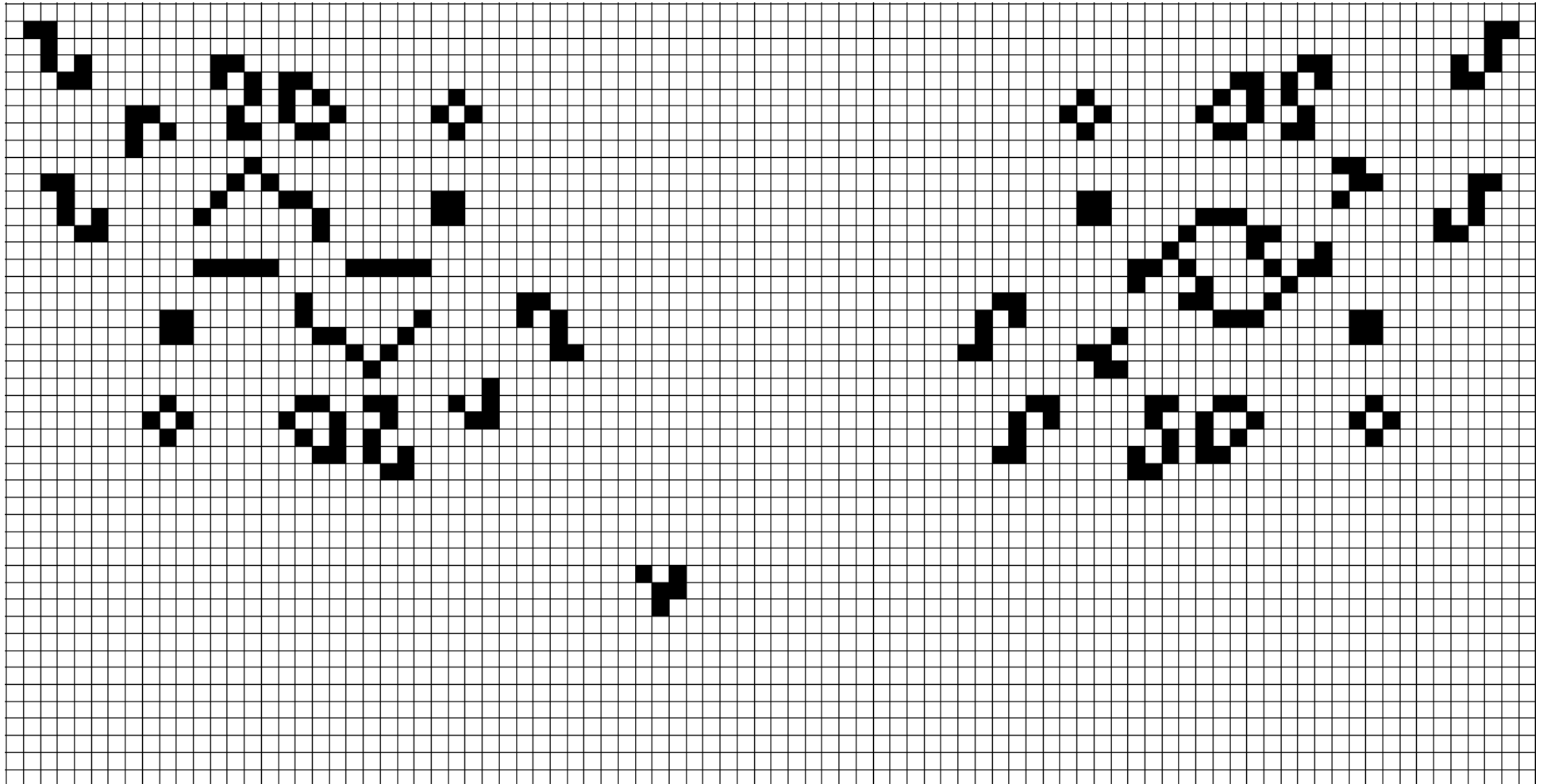
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

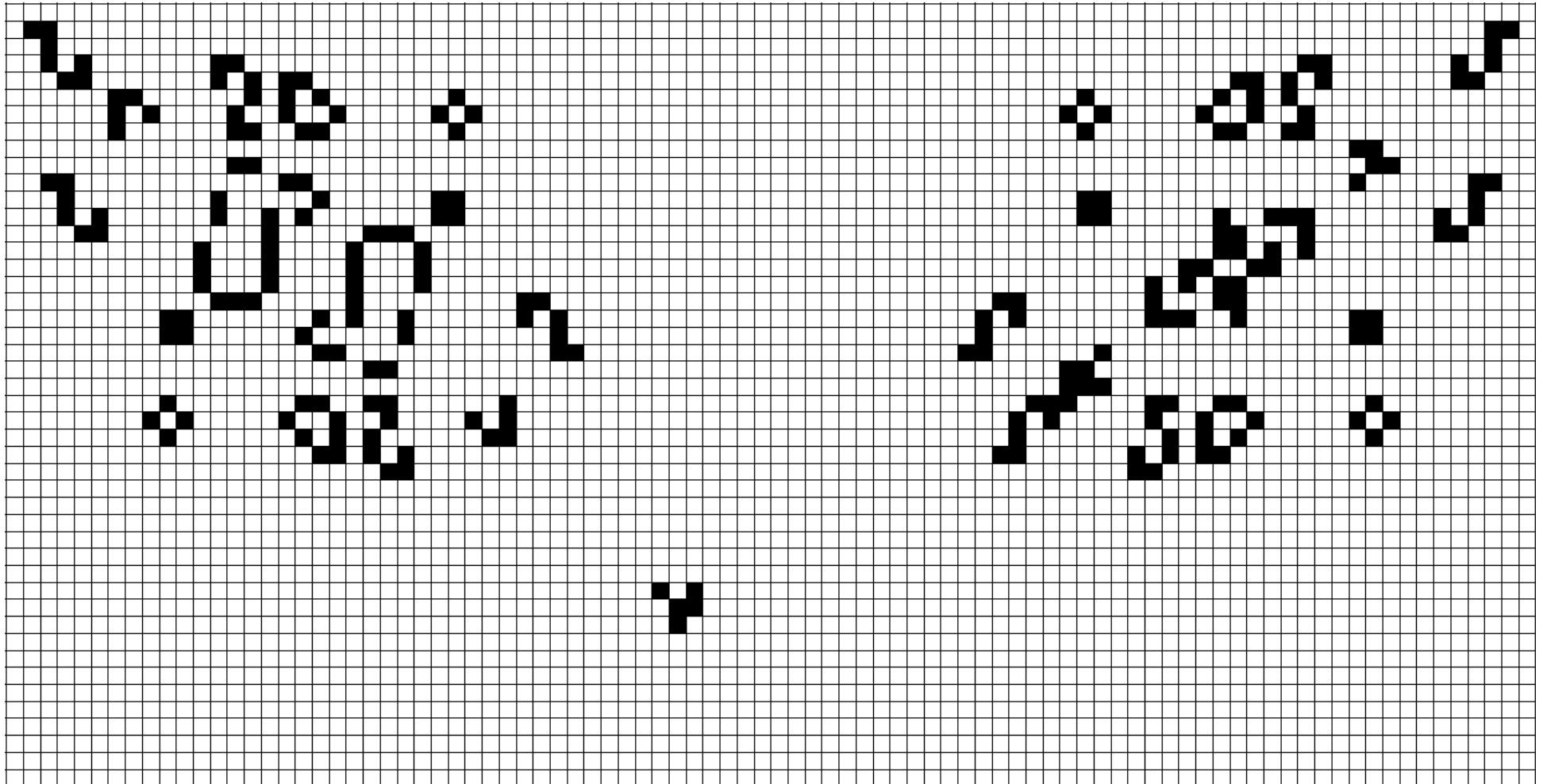
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

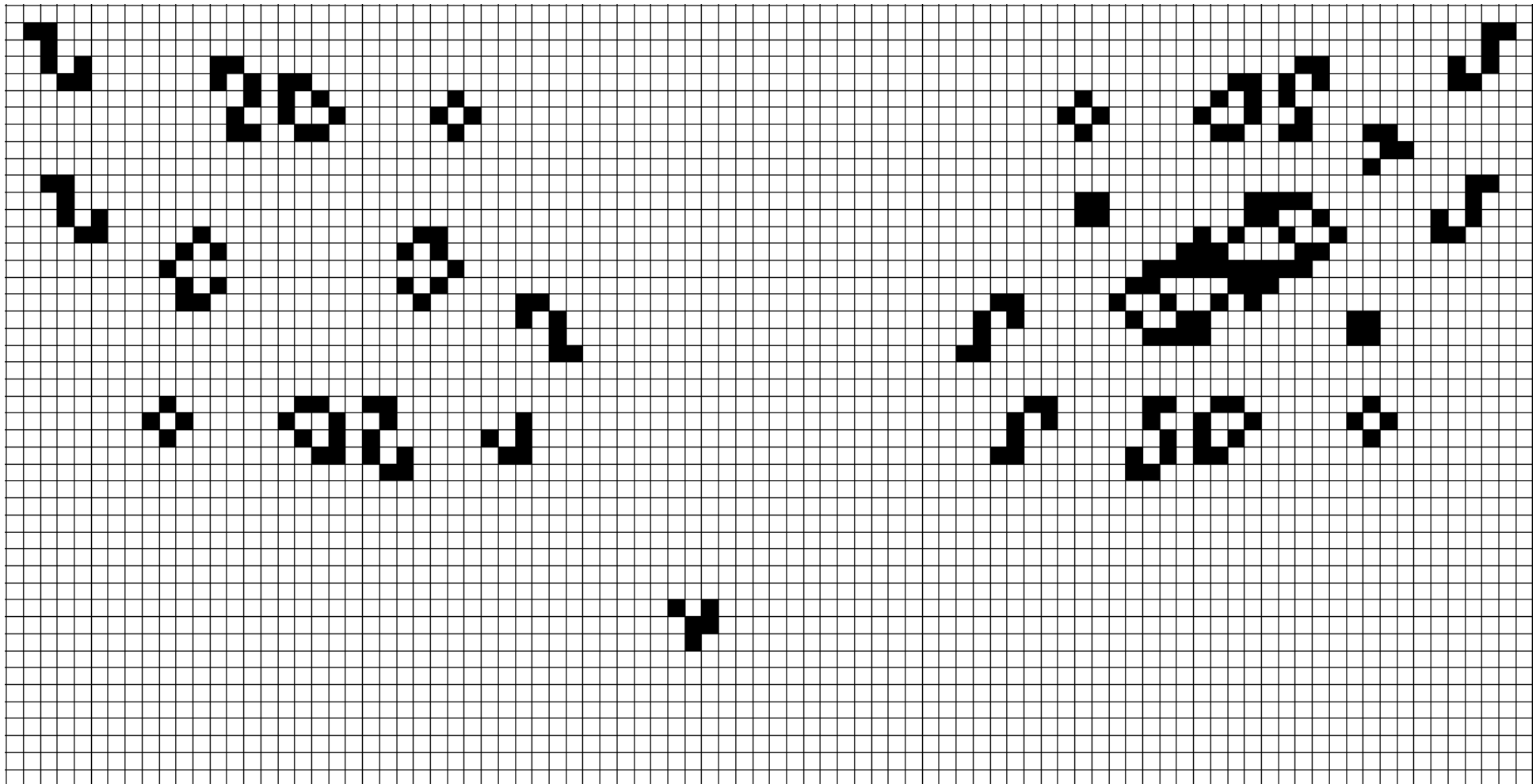
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

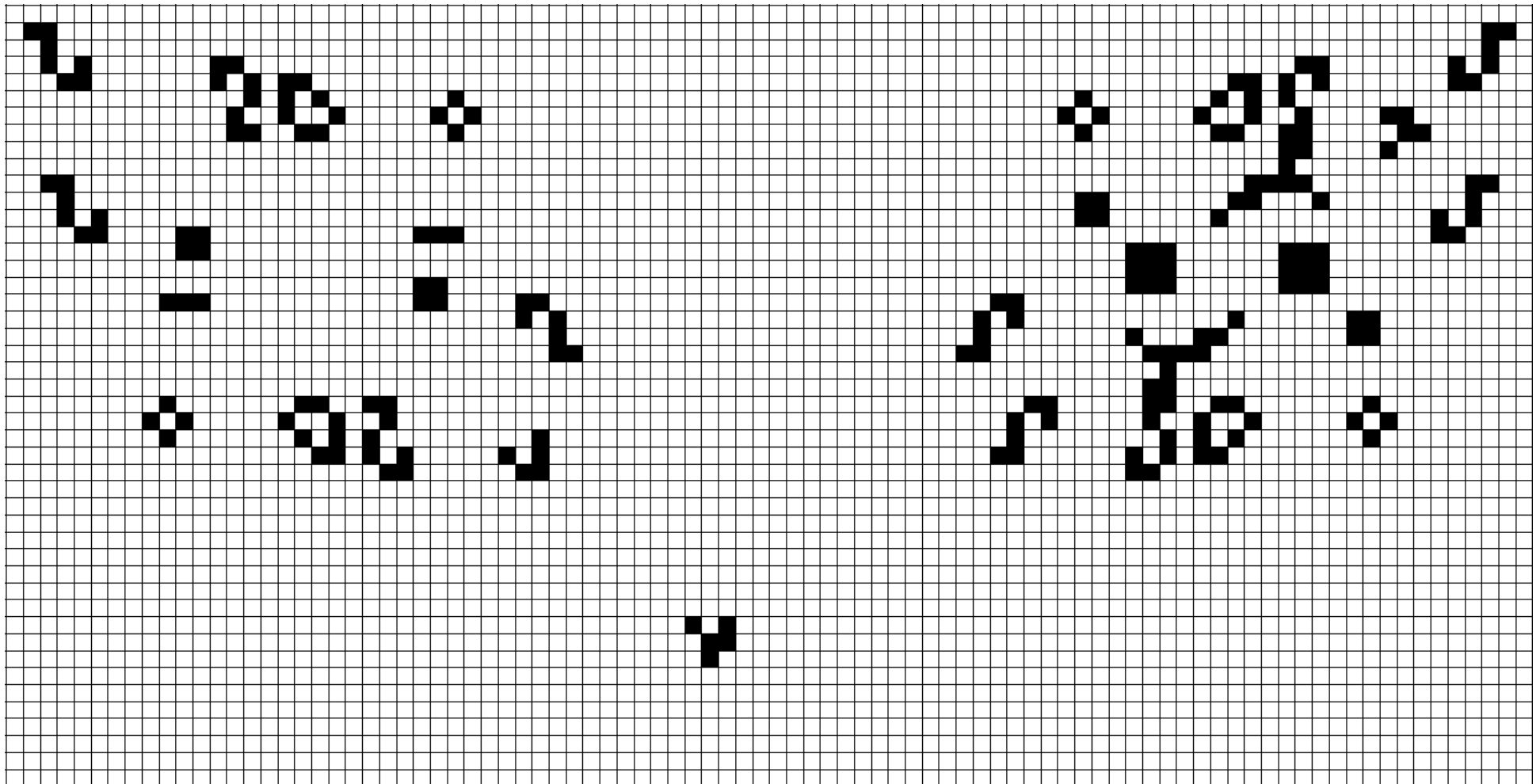
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

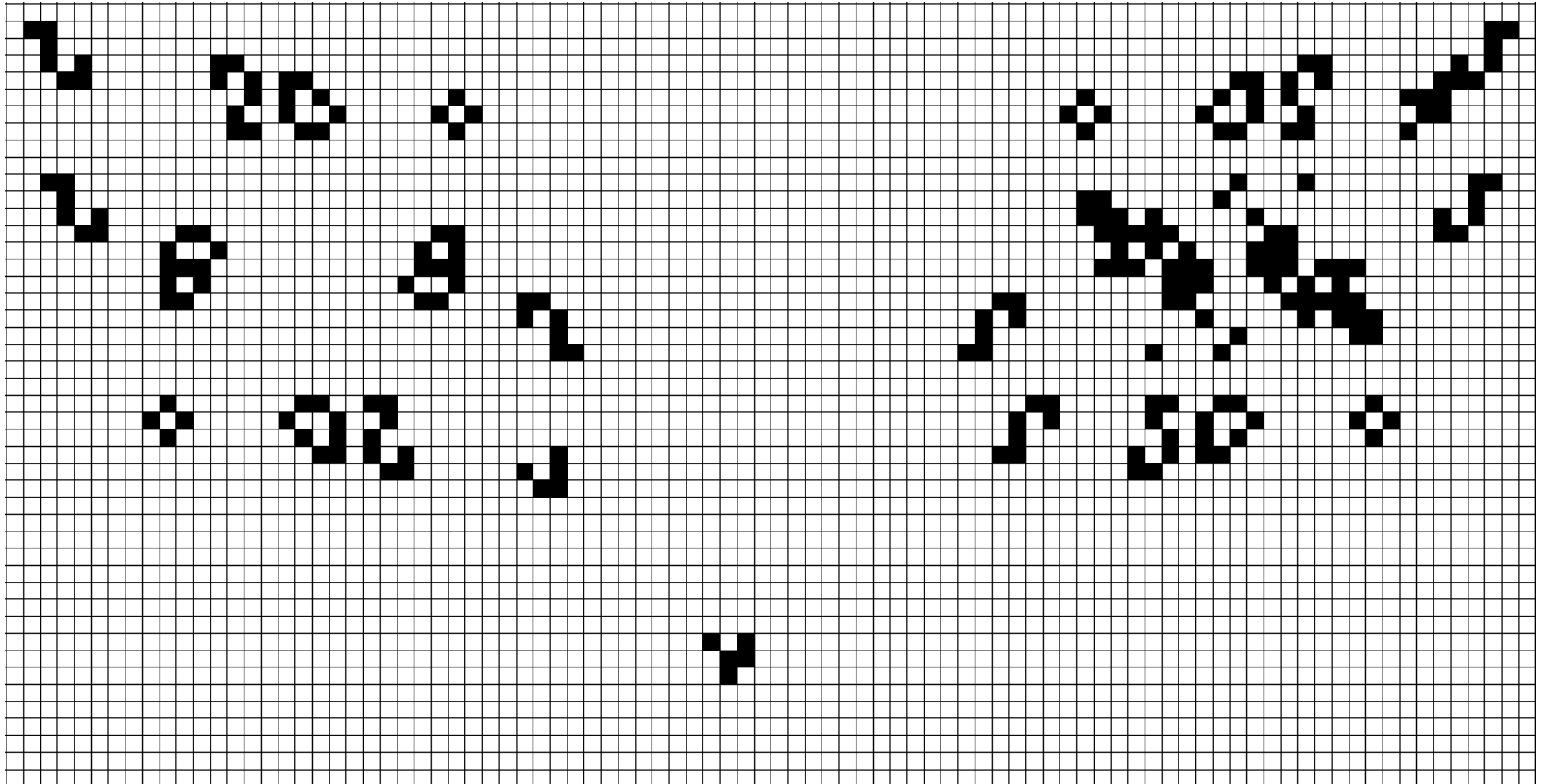
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

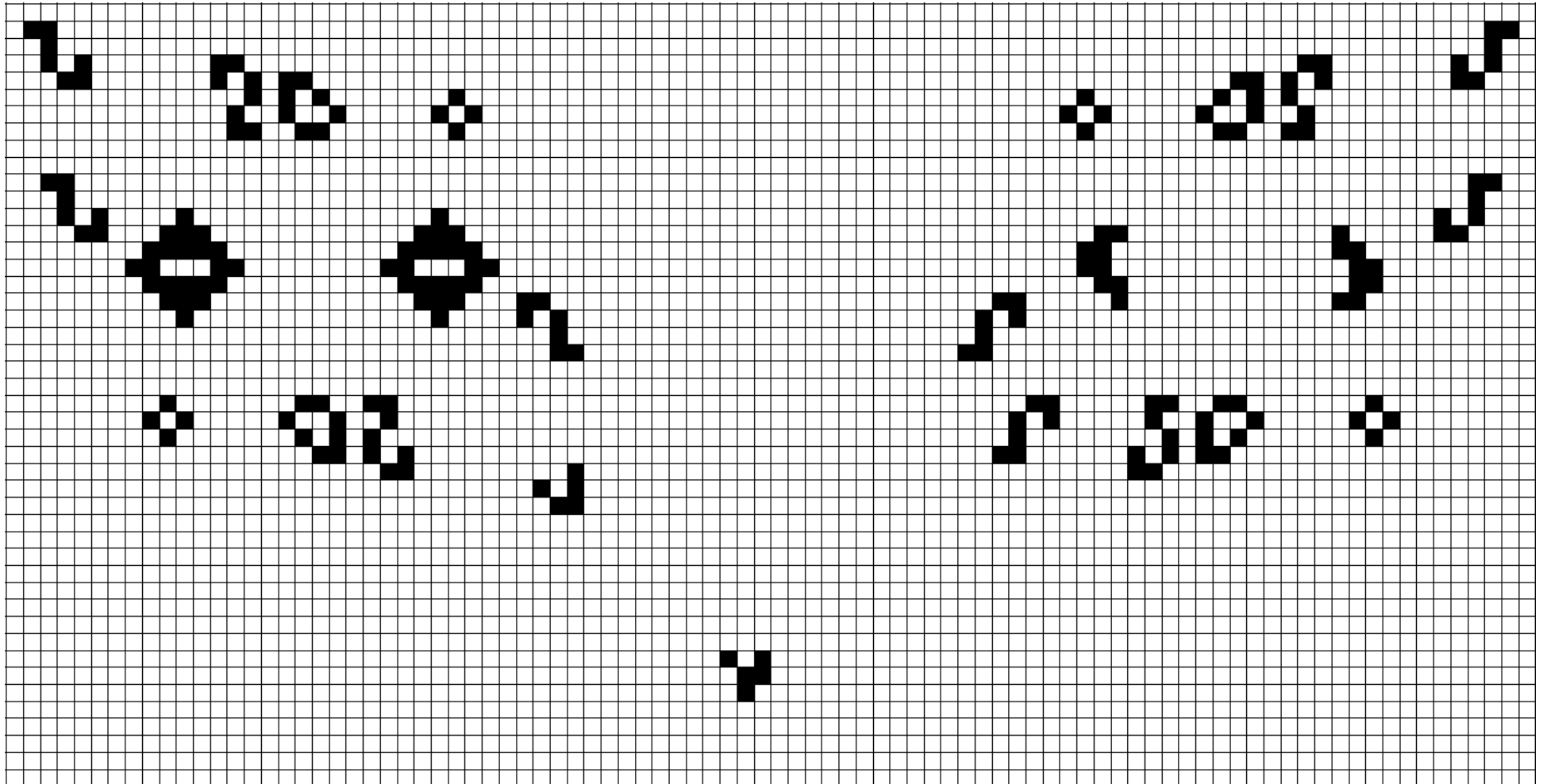
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

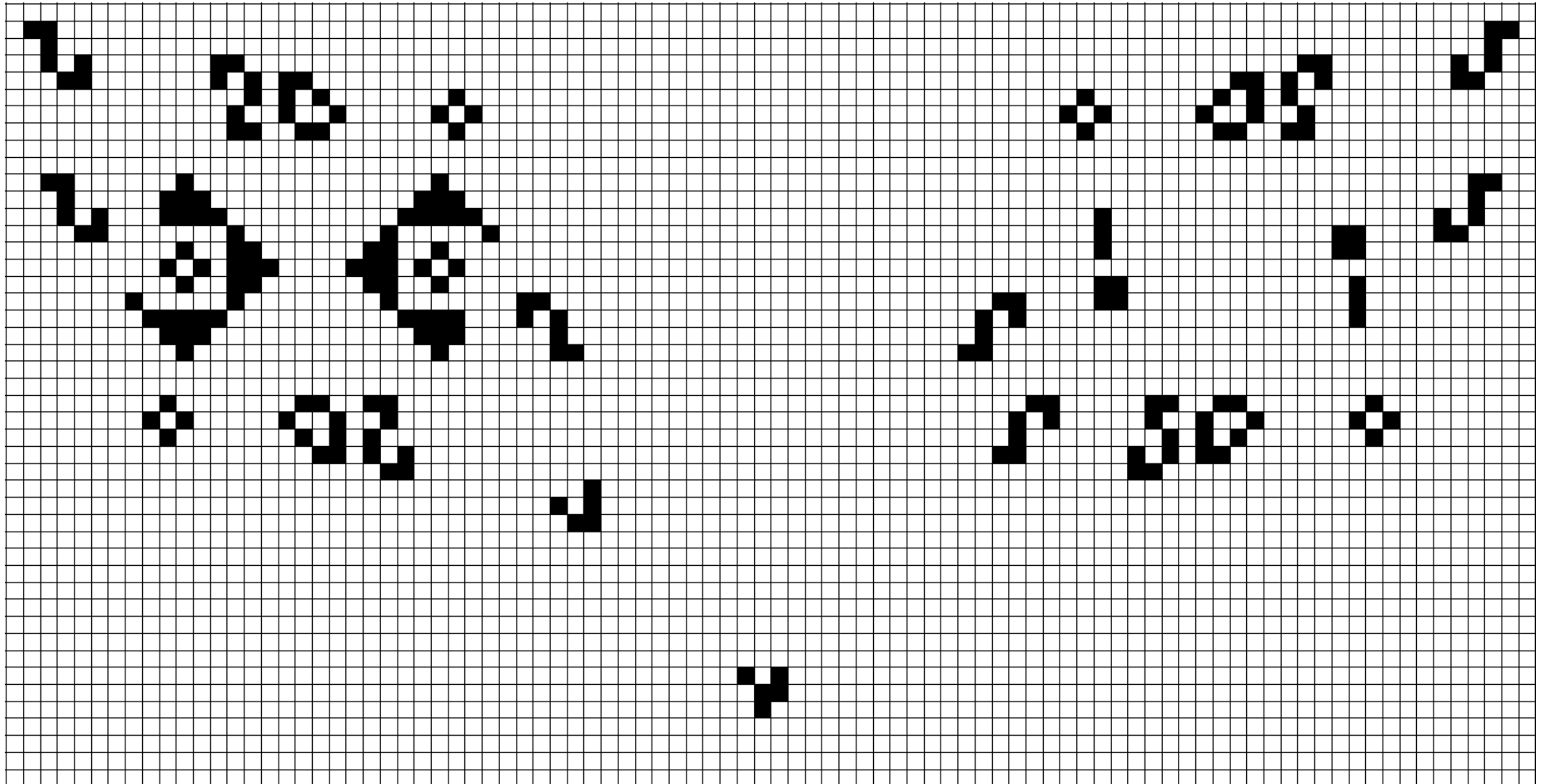
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

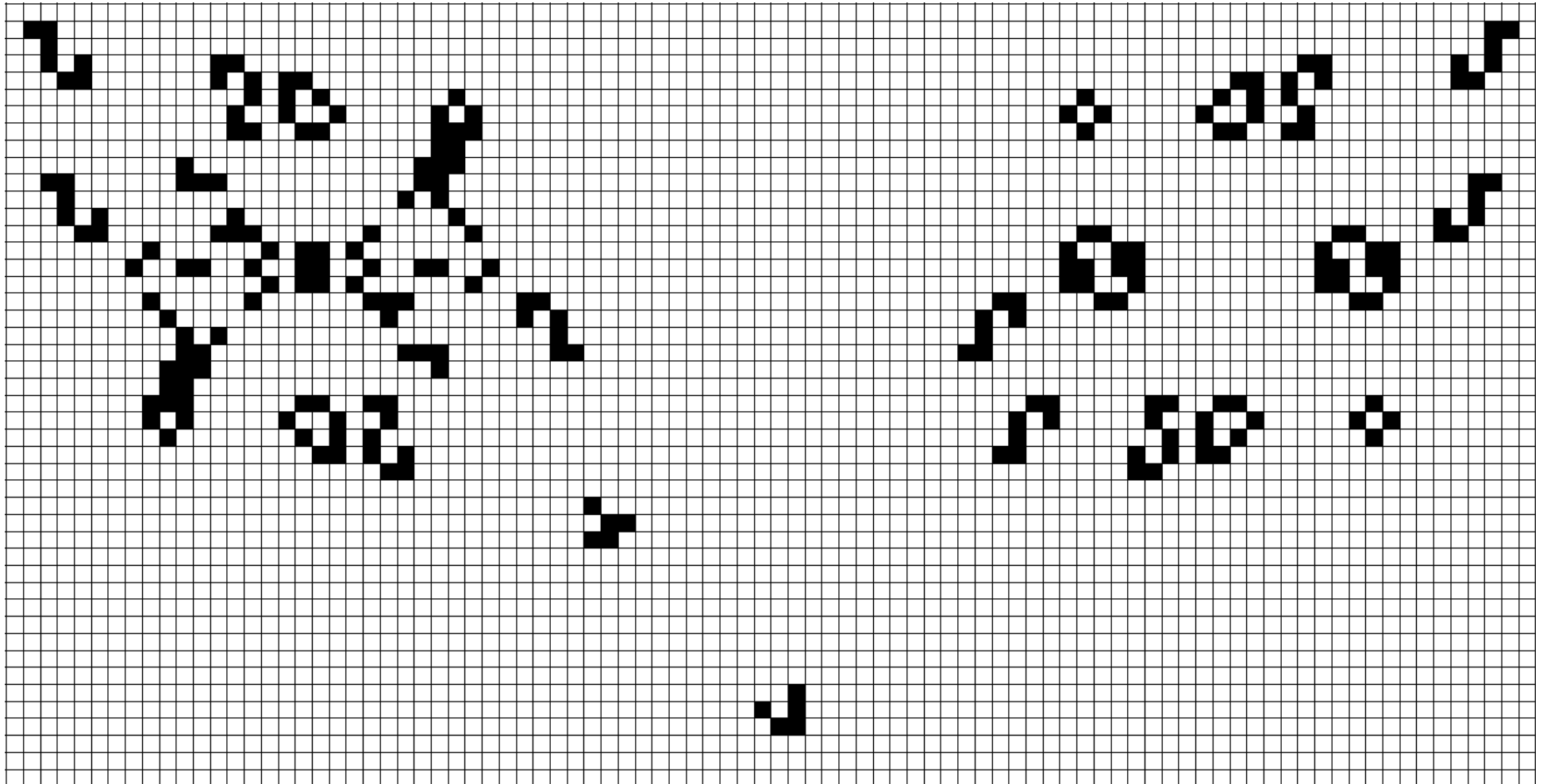
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

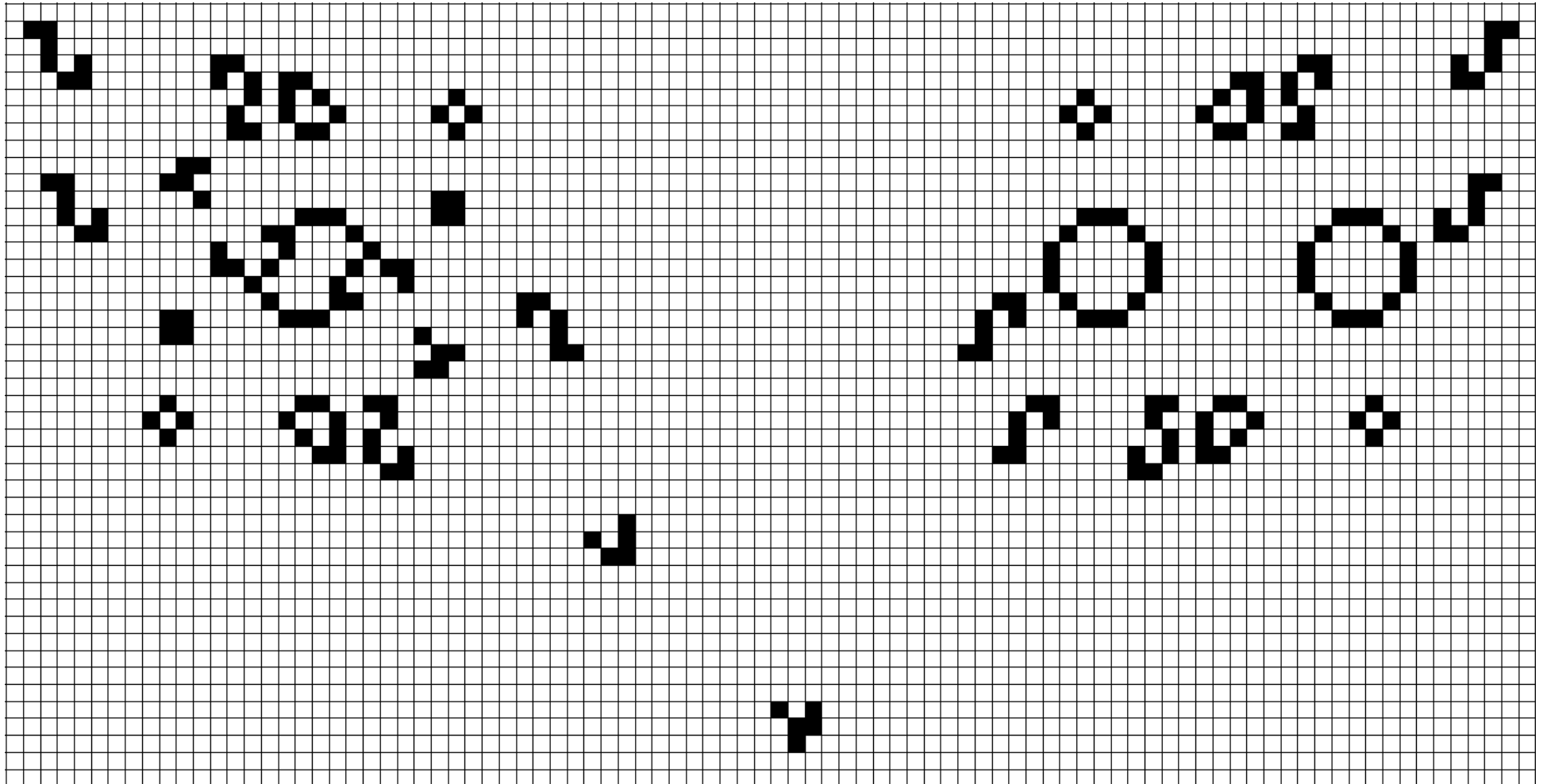
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

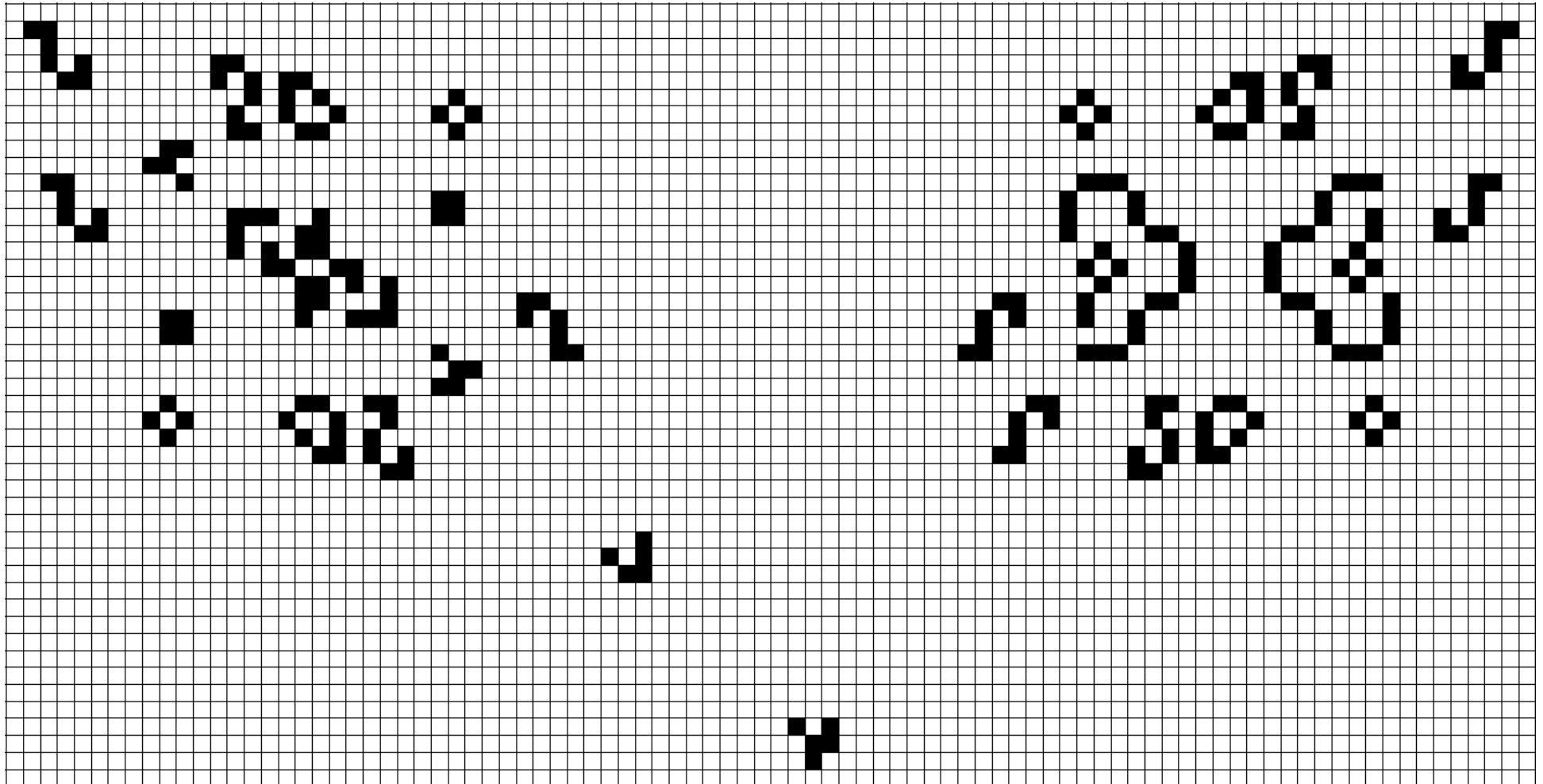
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

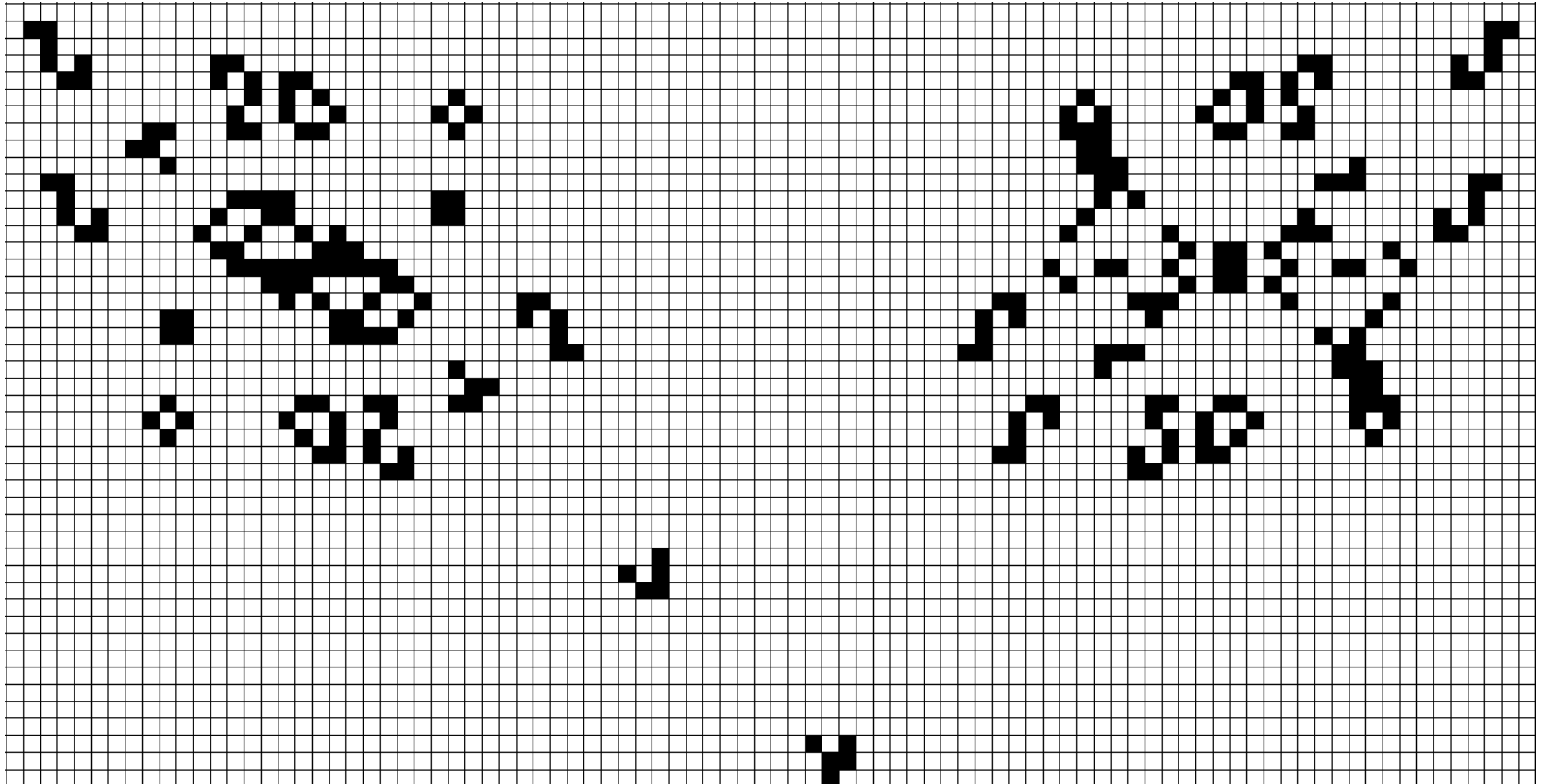
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

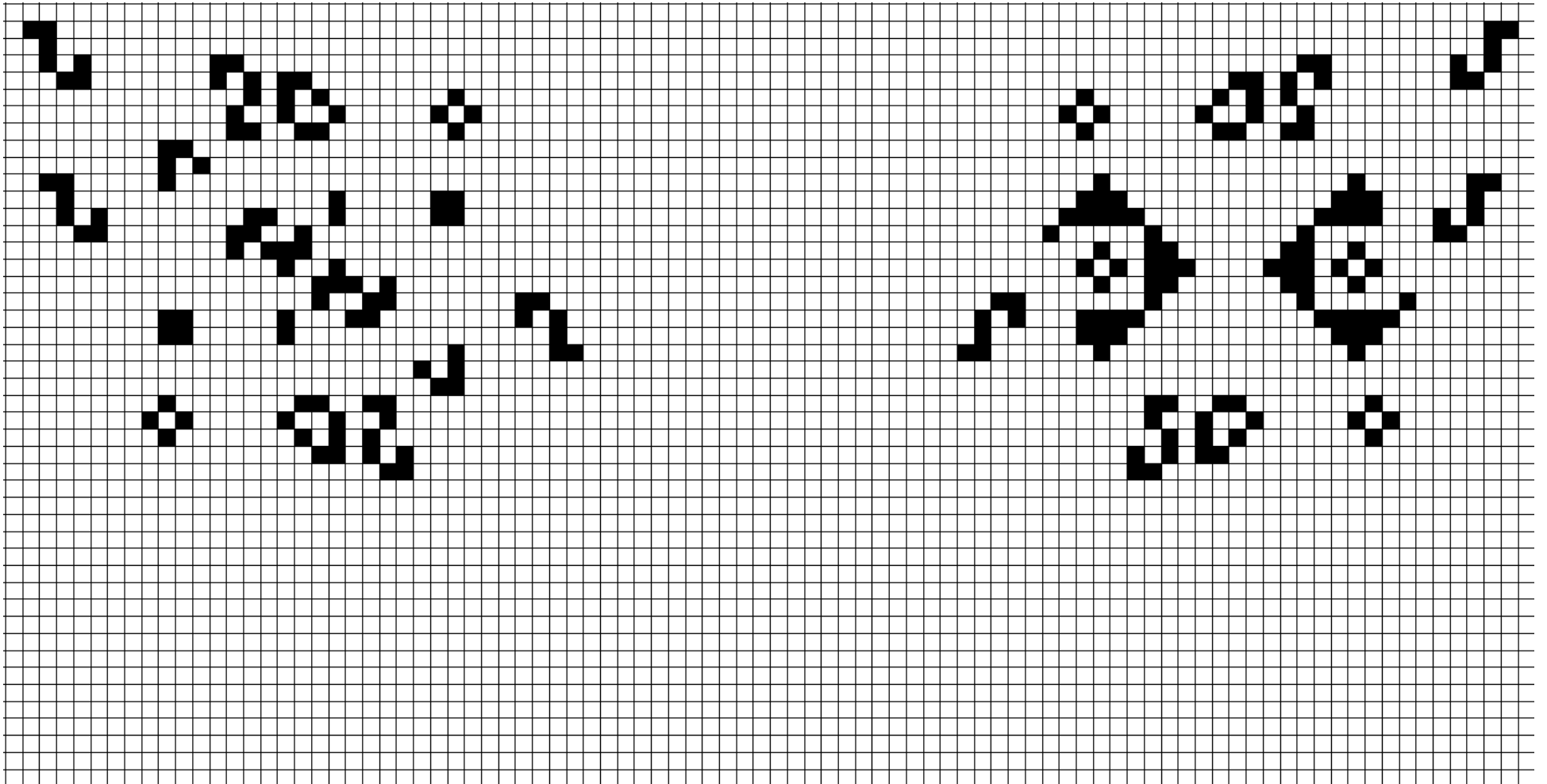
NO : Entrée 0



II. Outils nécessaires et composition du filtre

2. Portes logiques

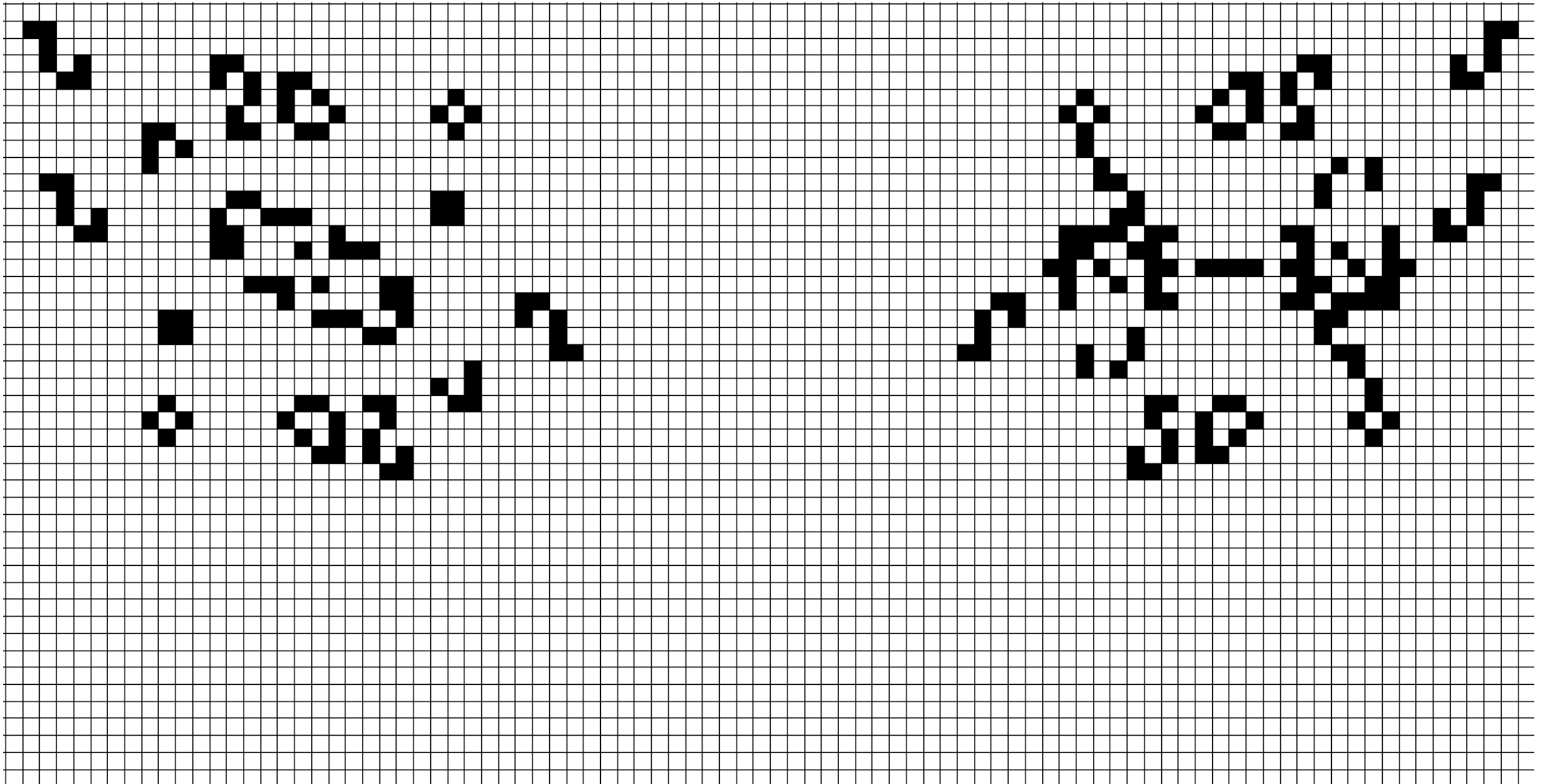
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

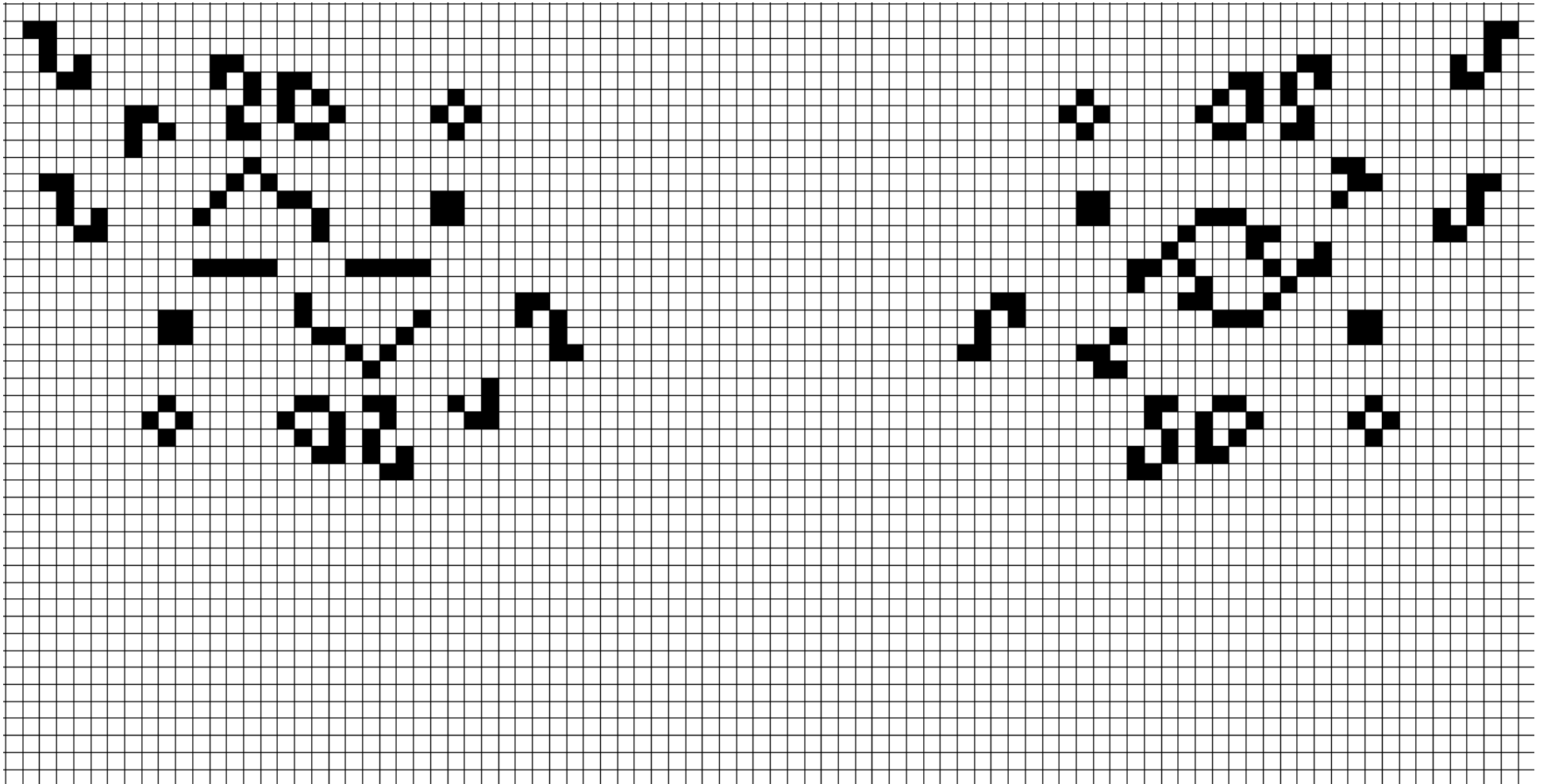
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

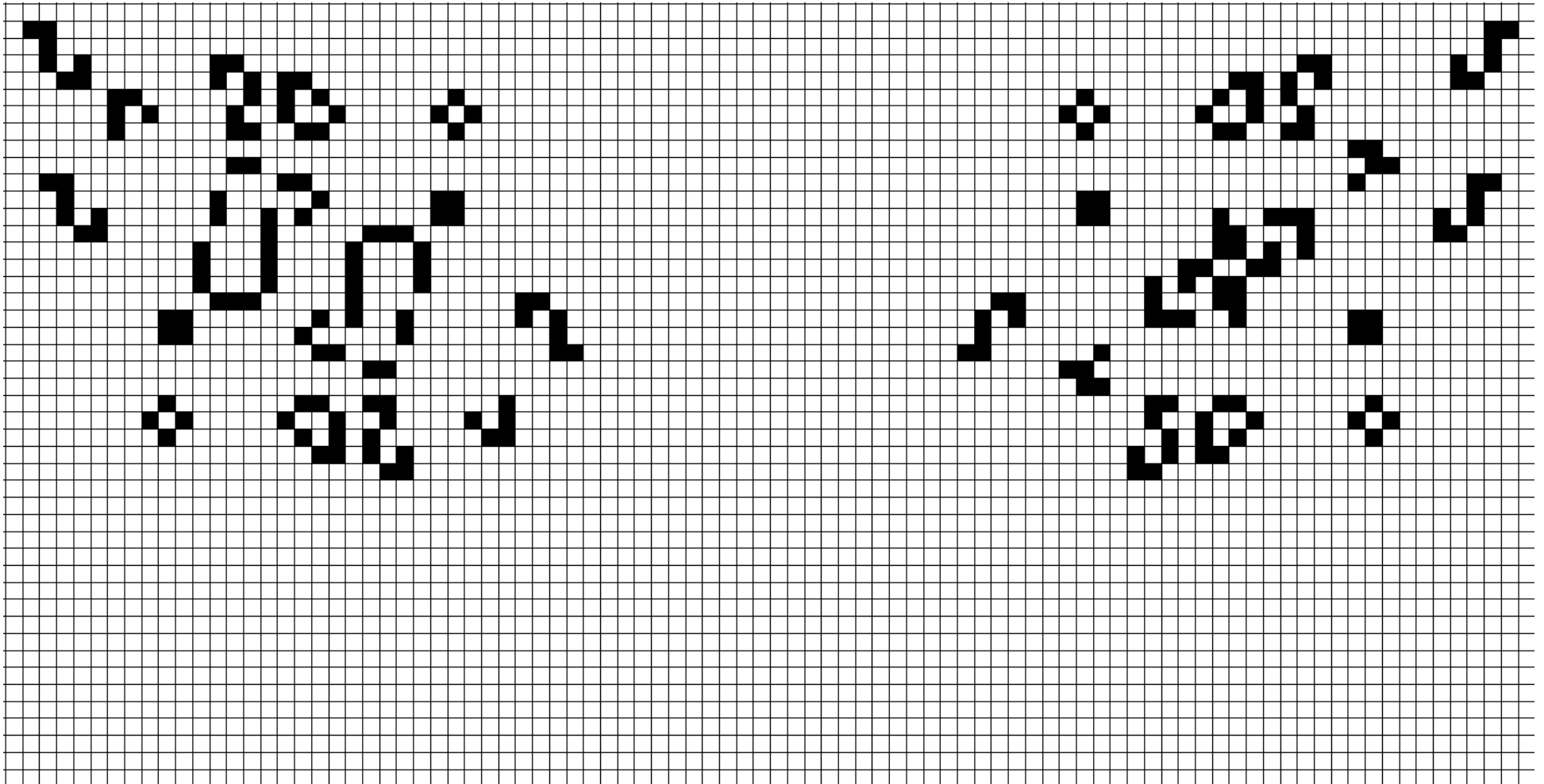
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

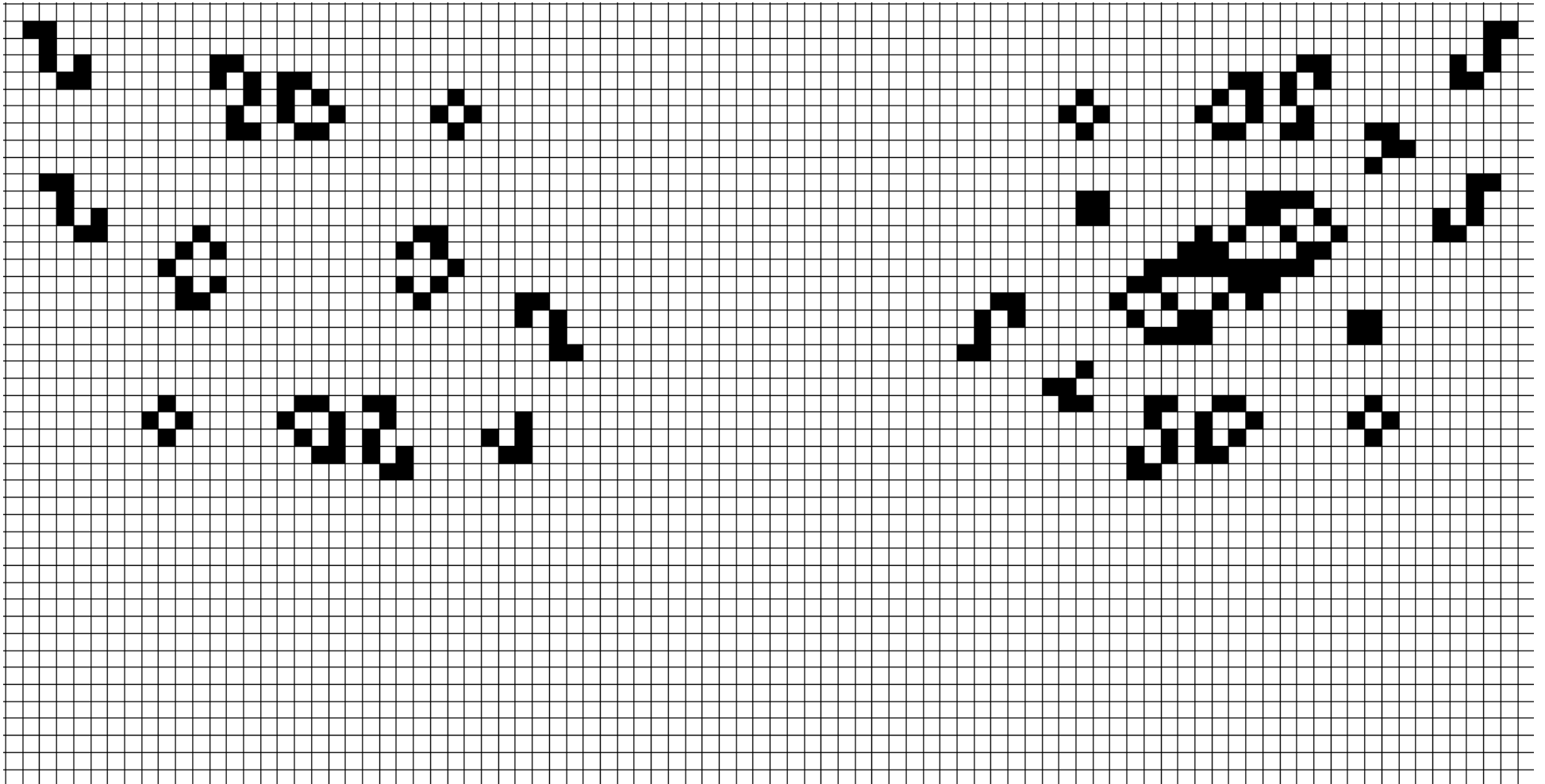
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

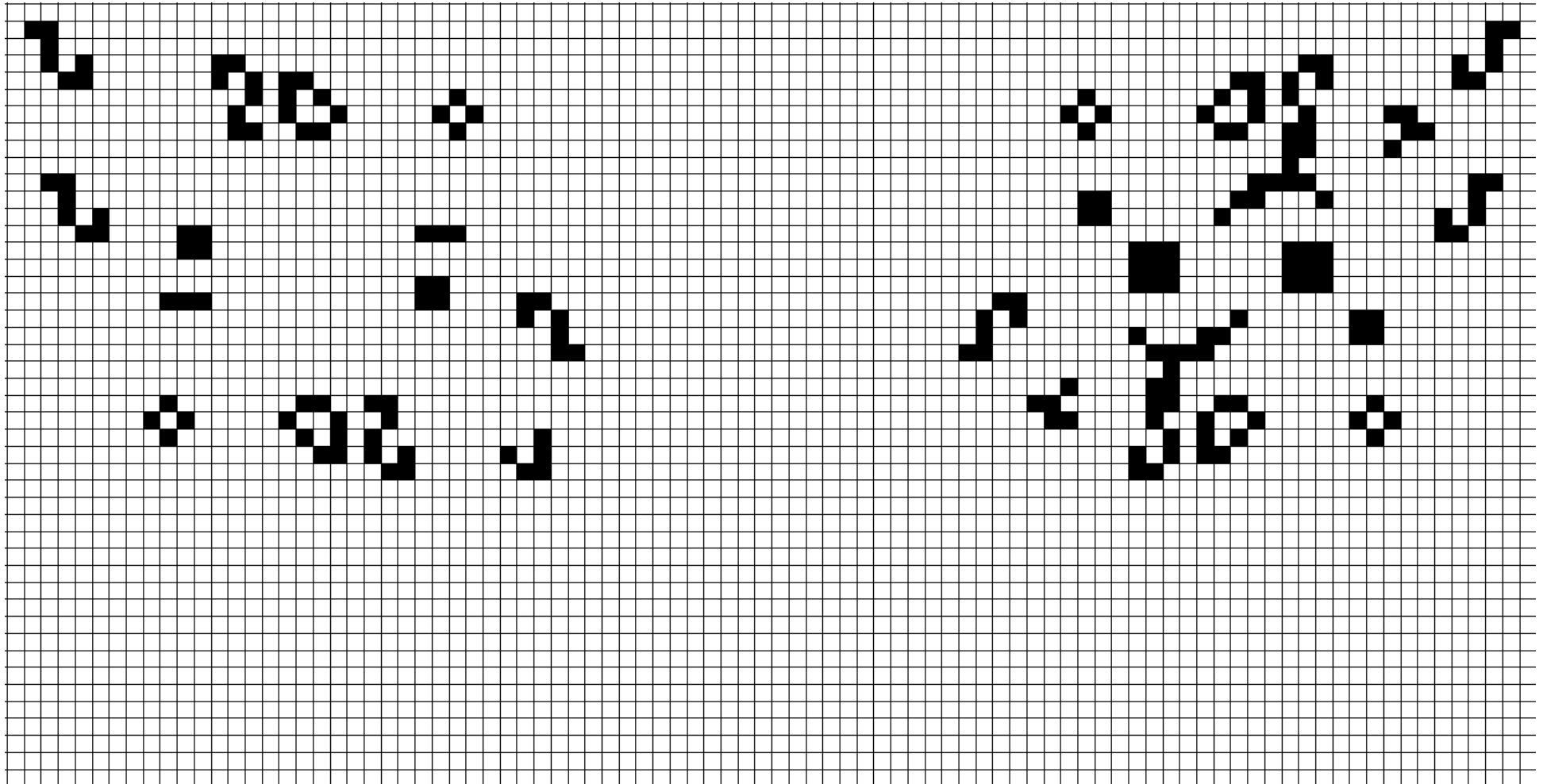
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

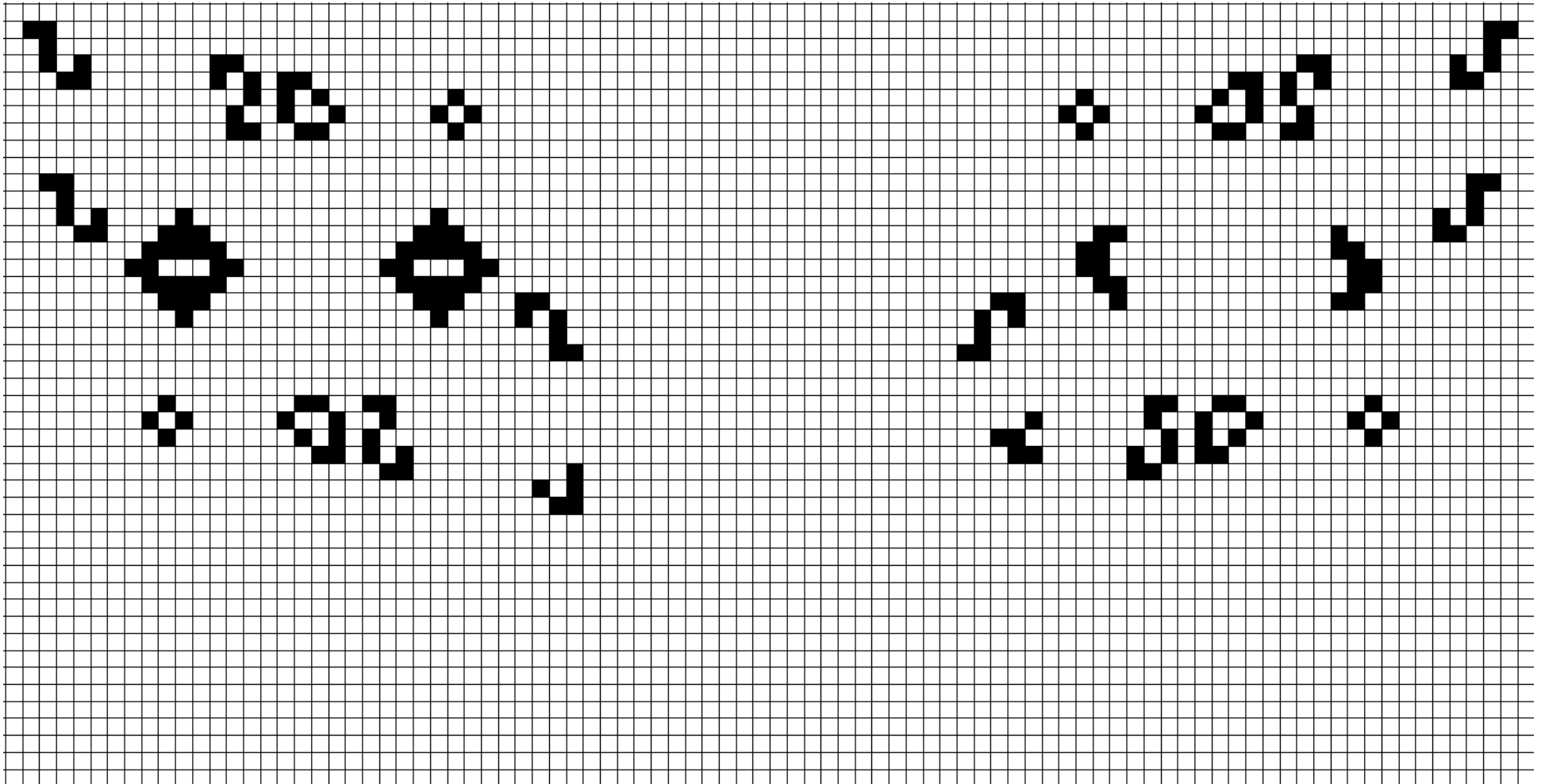
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

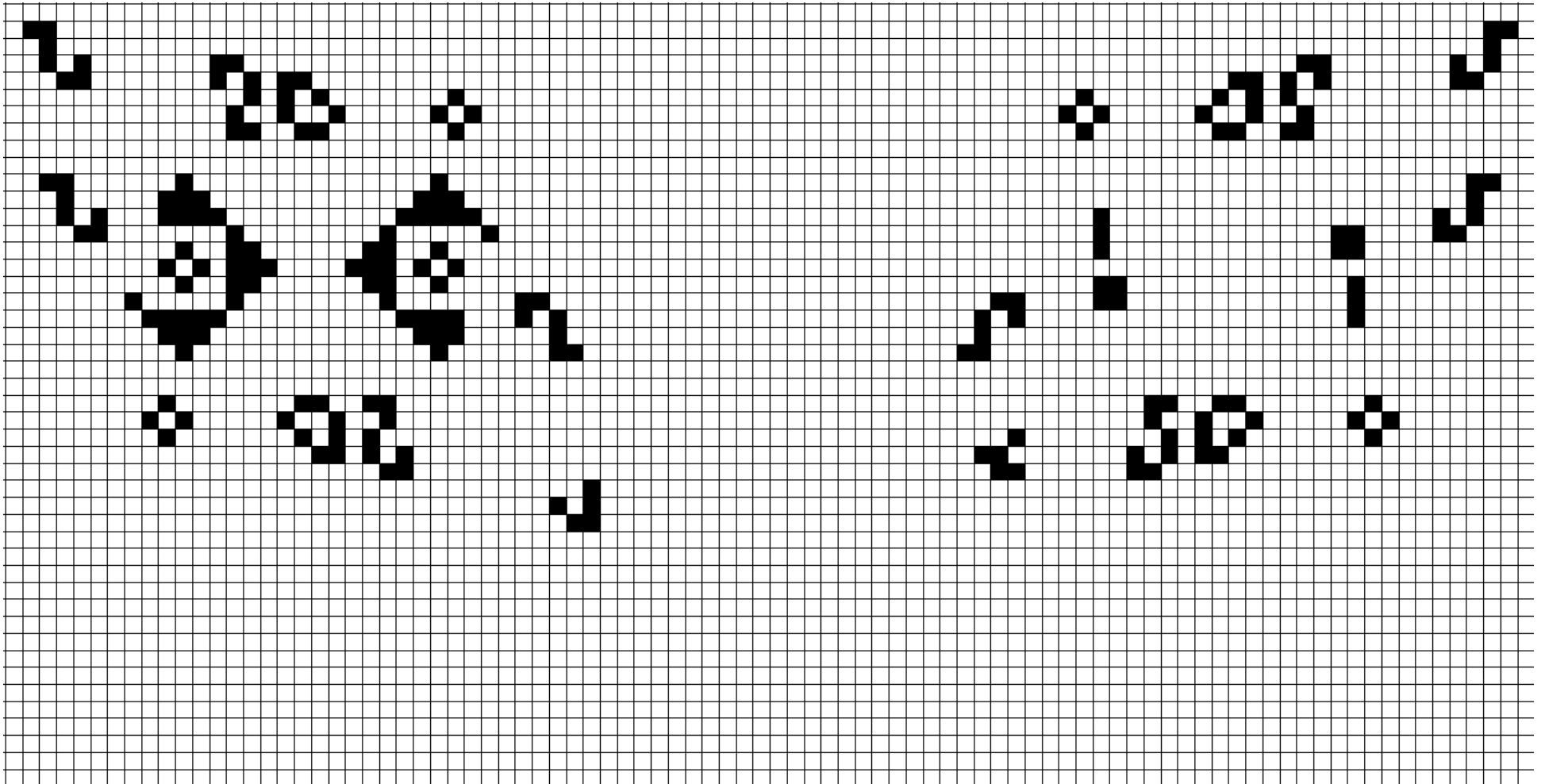
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

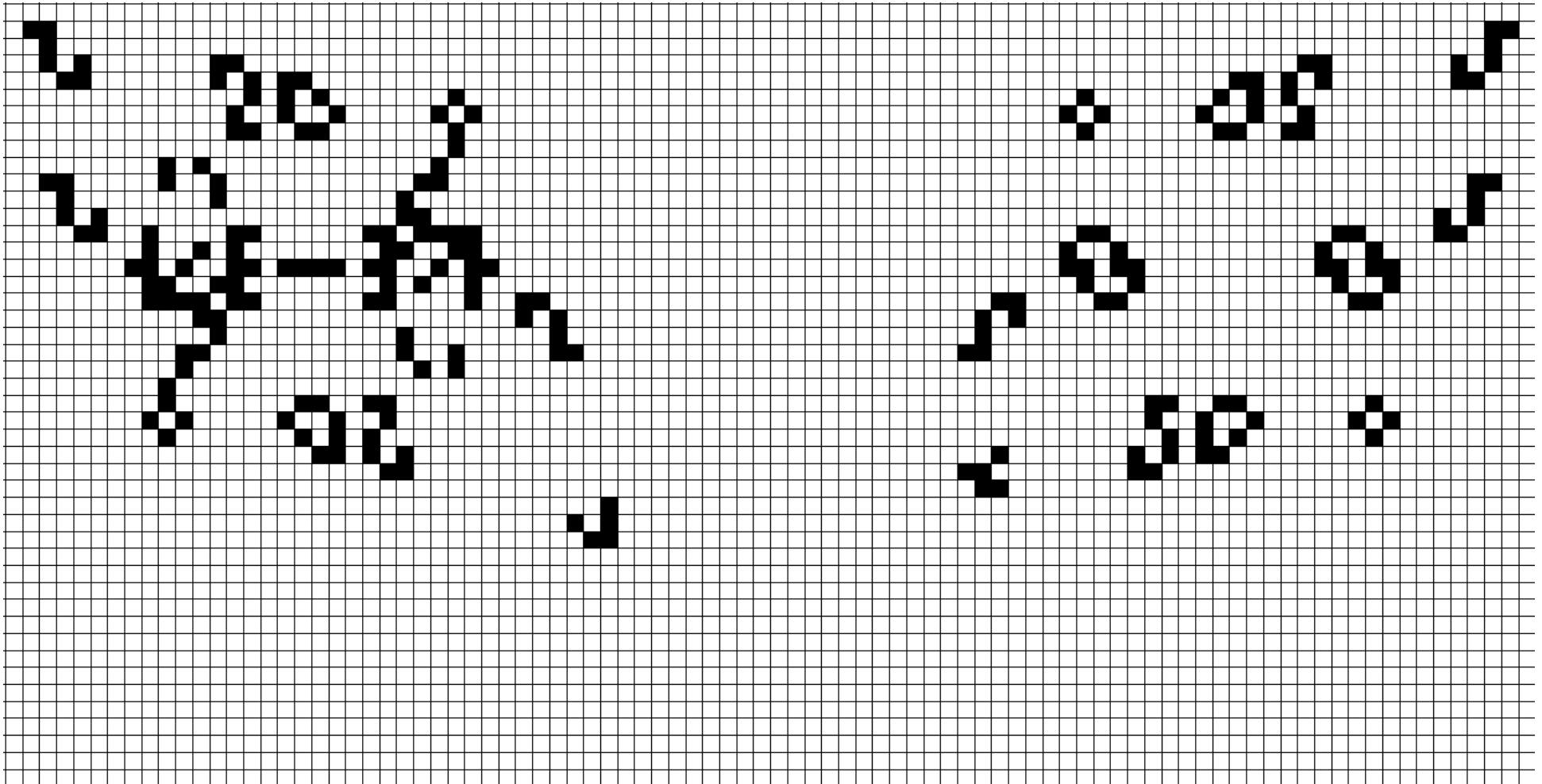
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

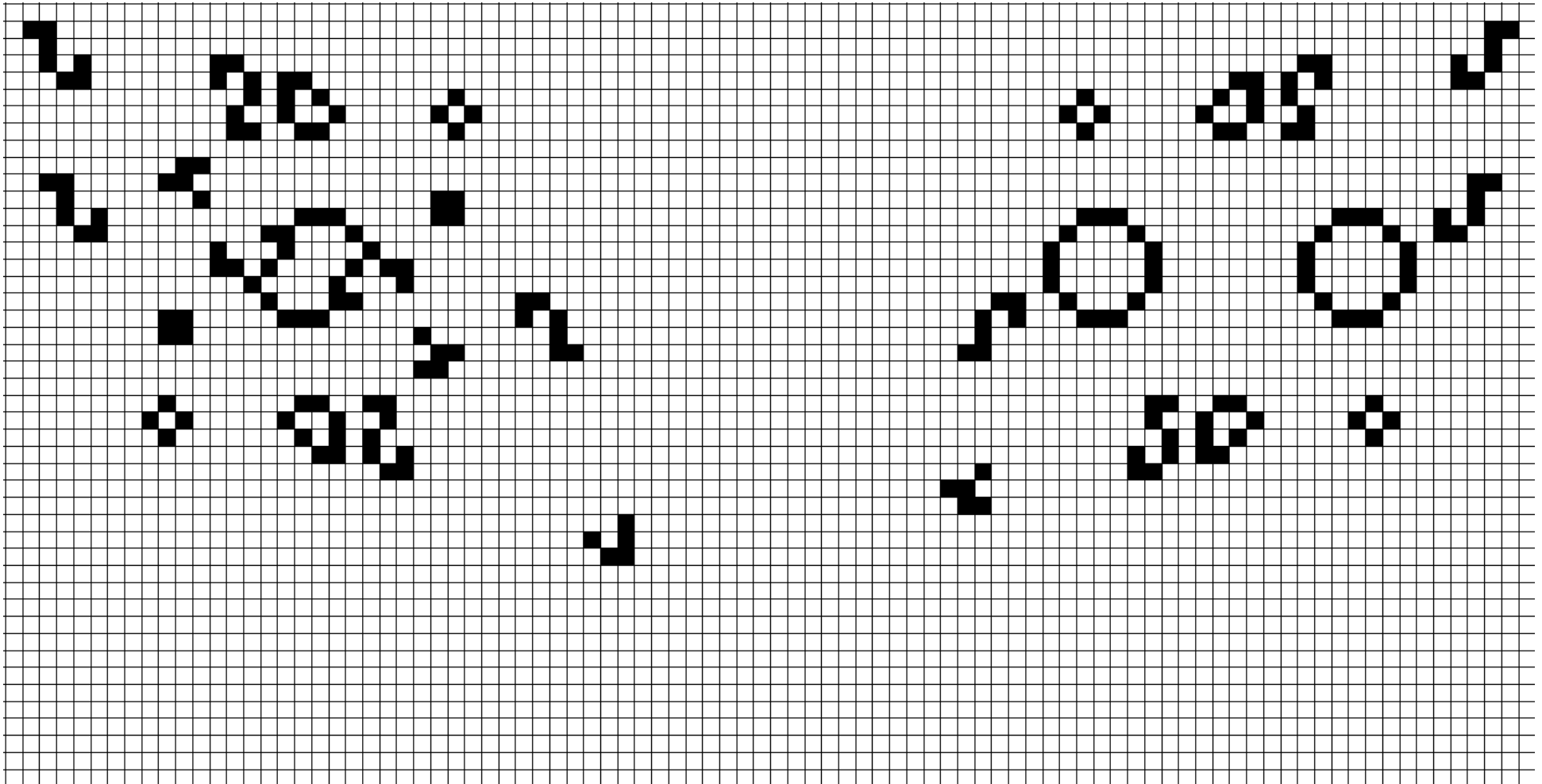
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

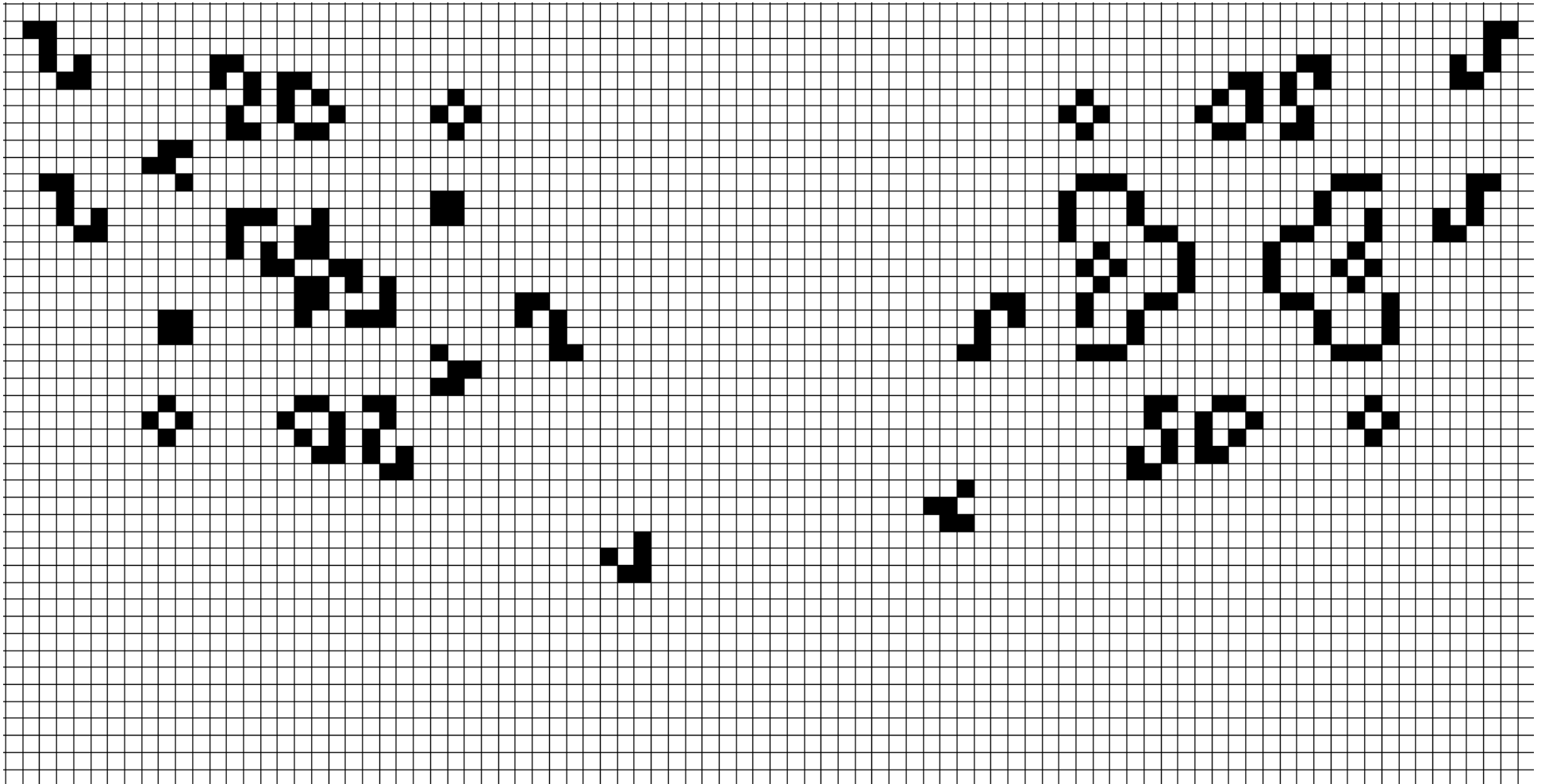
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

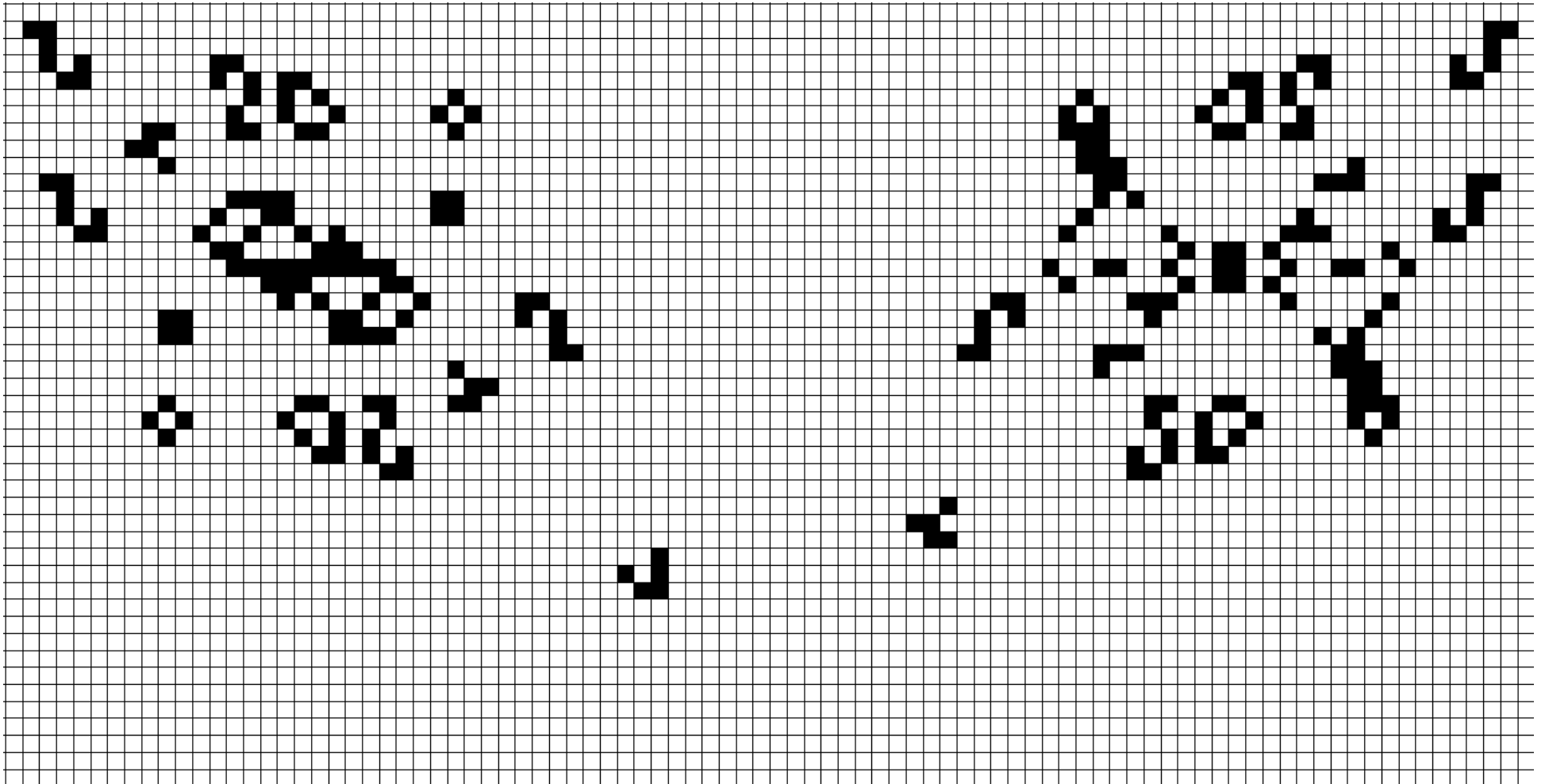
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

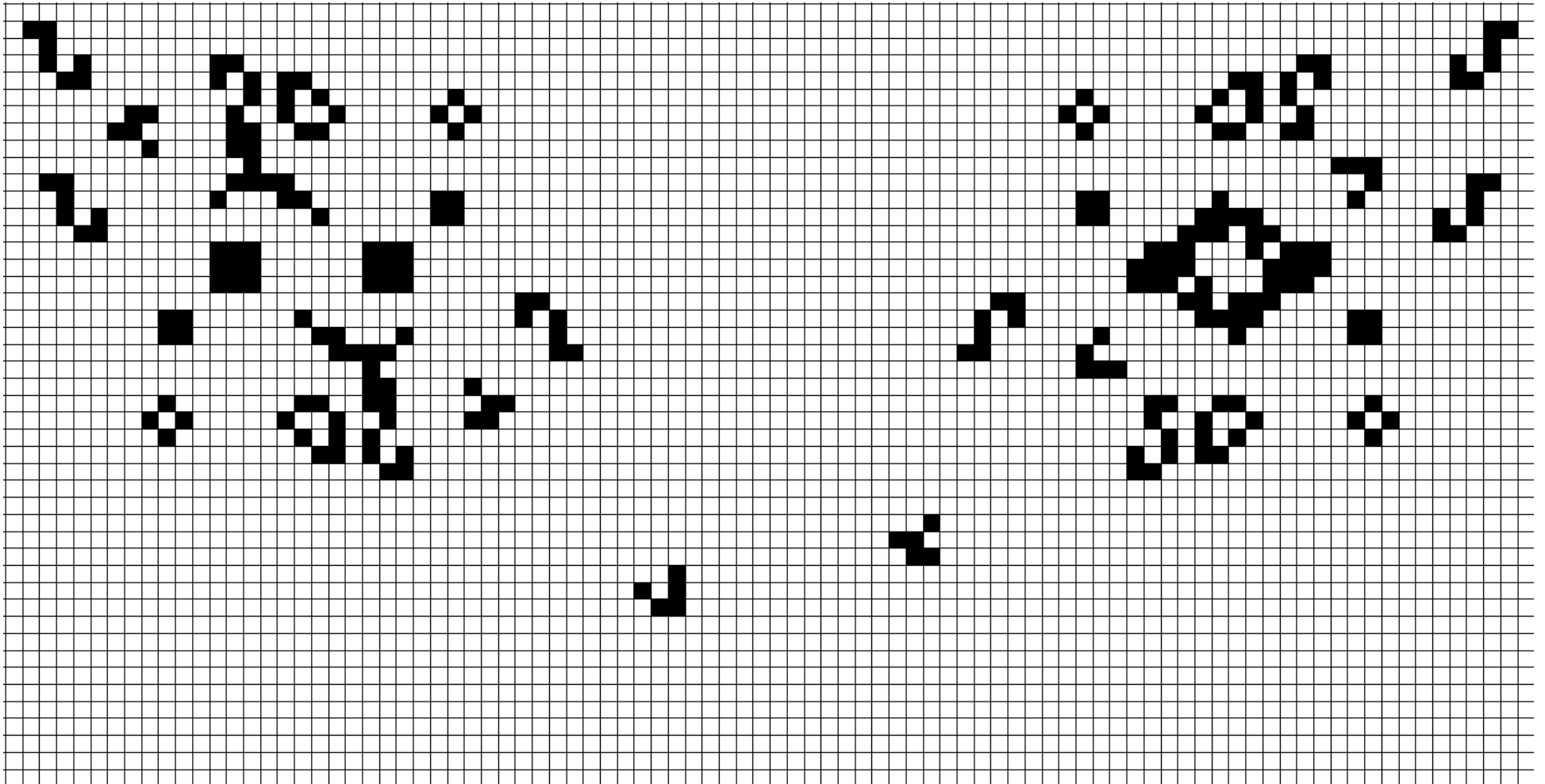
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

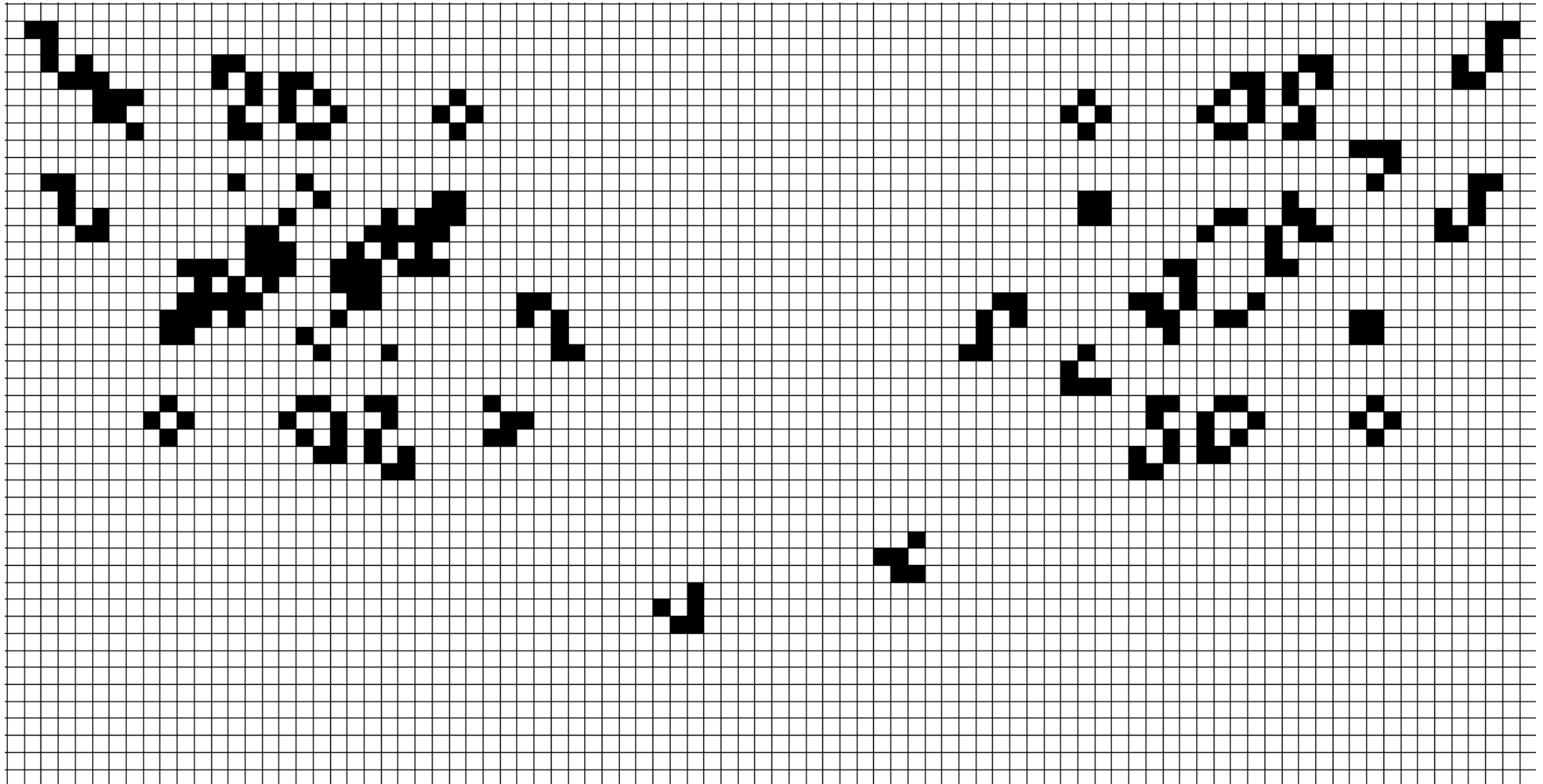
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

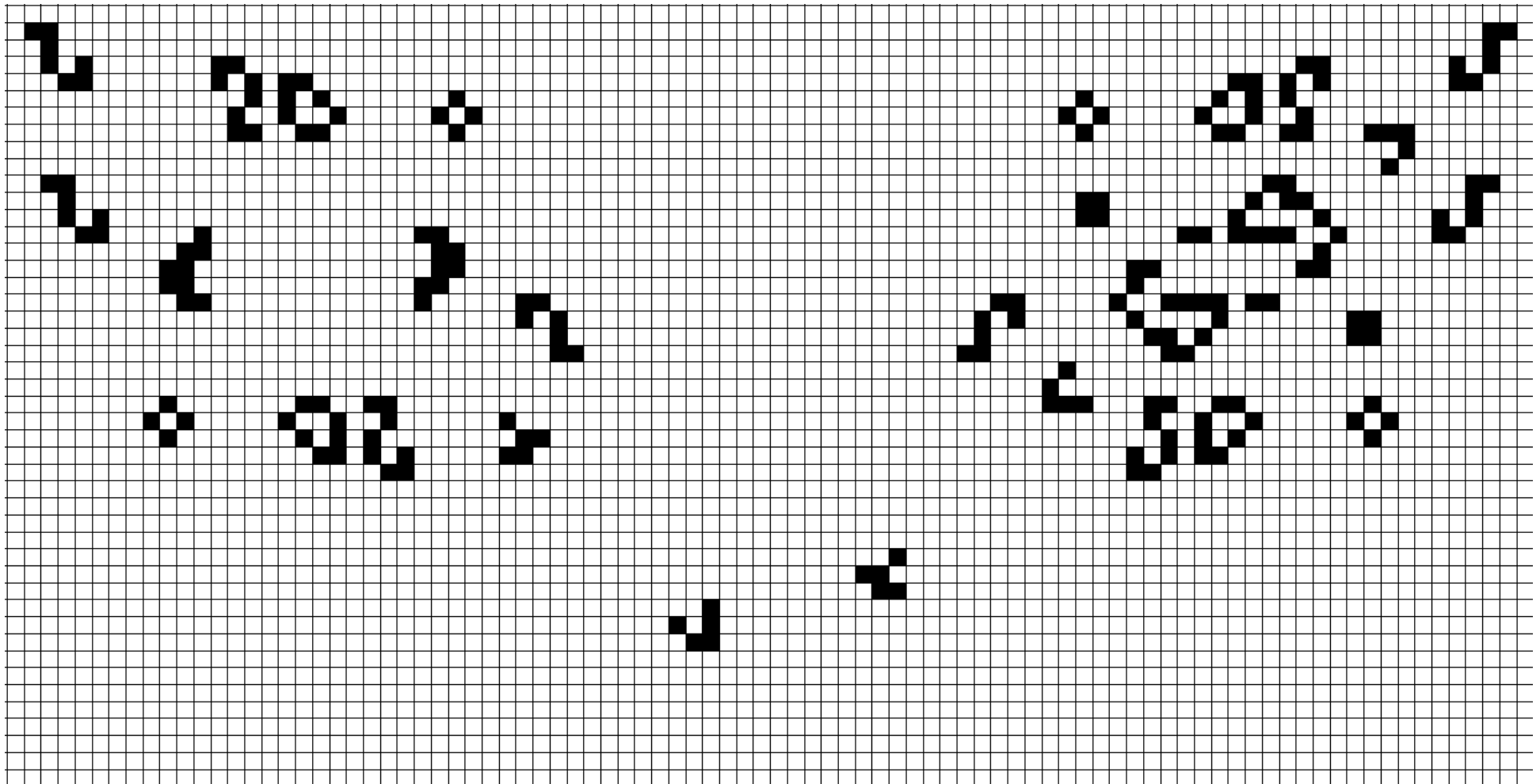
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

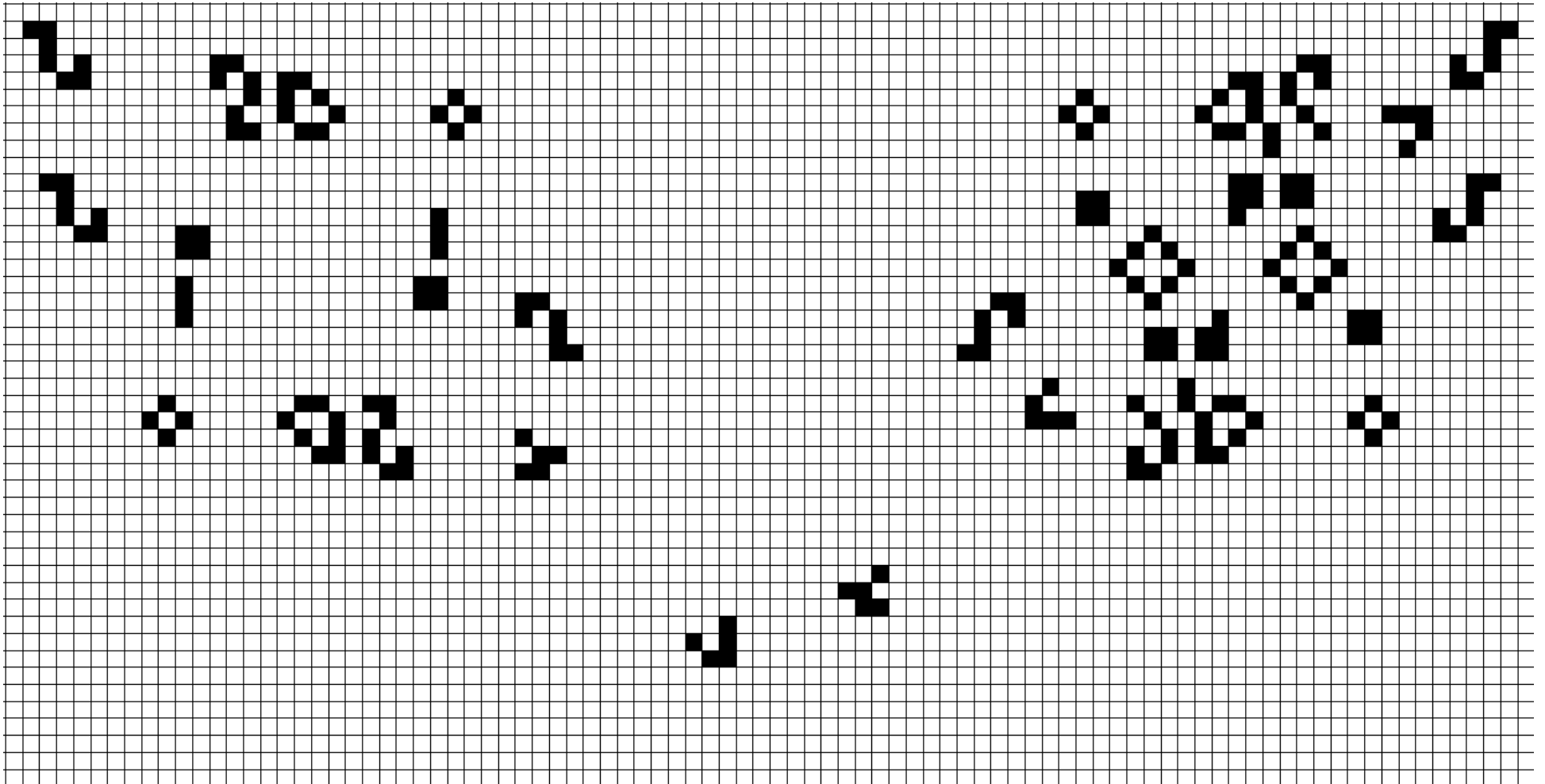
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

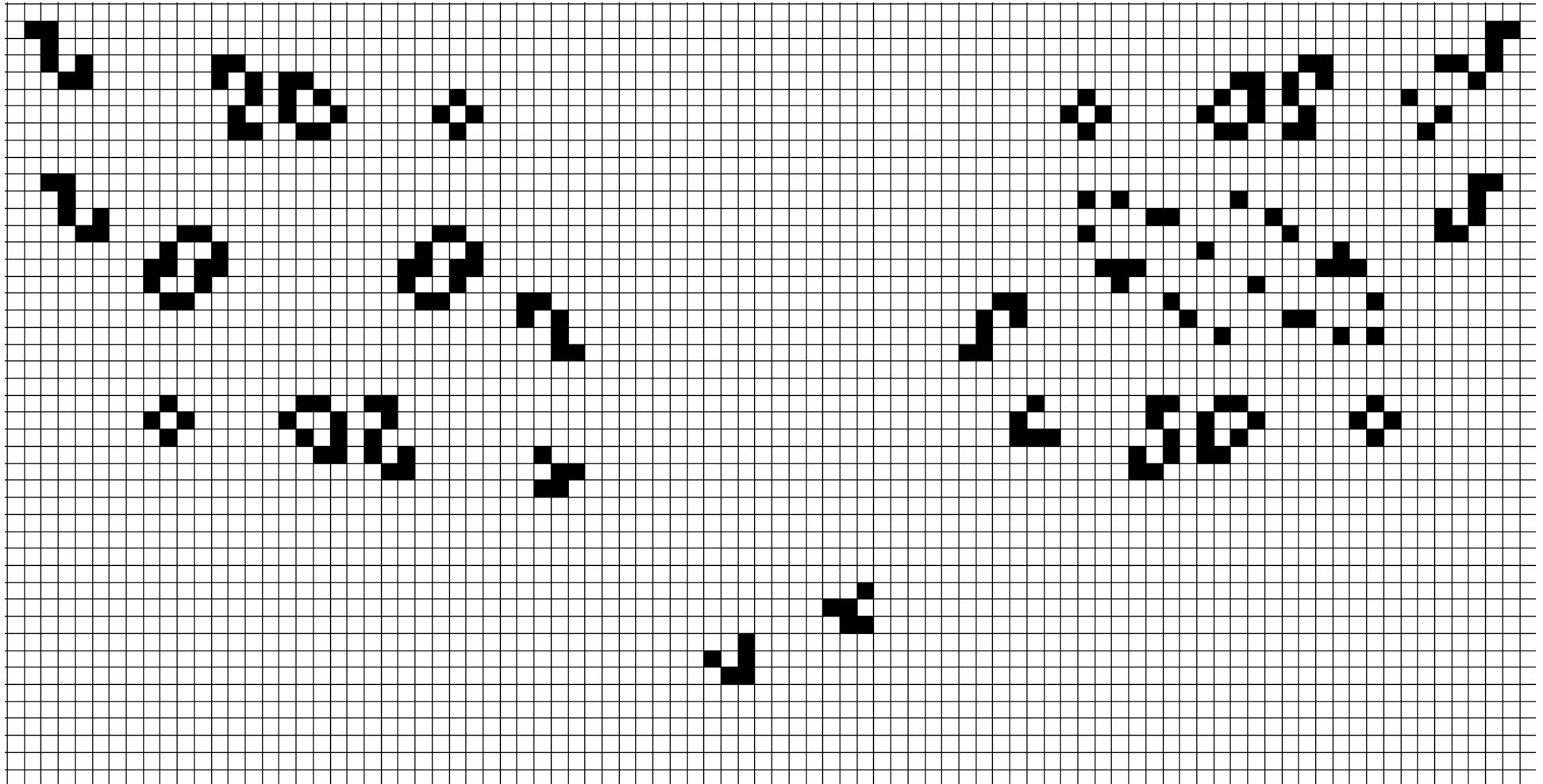
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

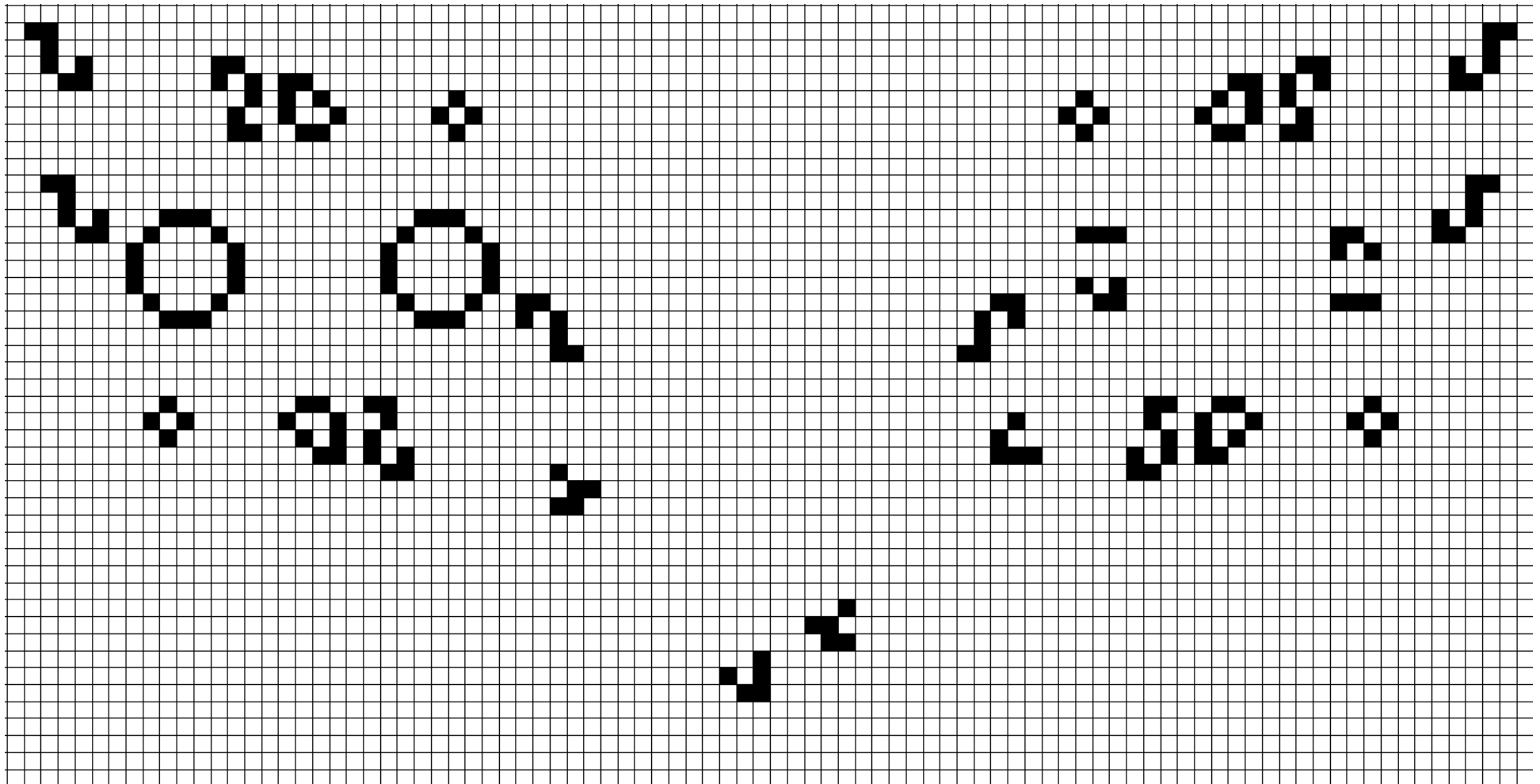
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

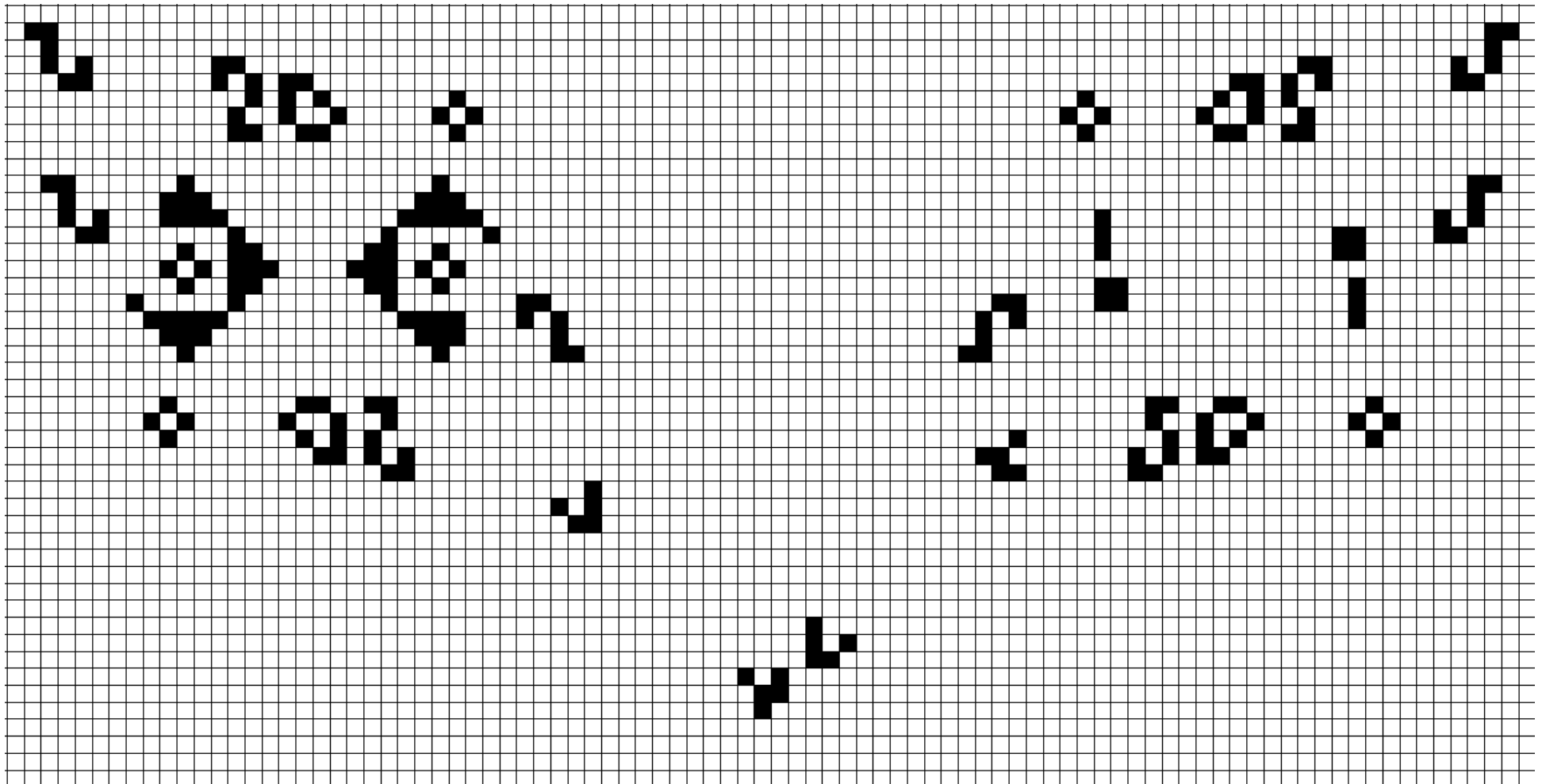
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

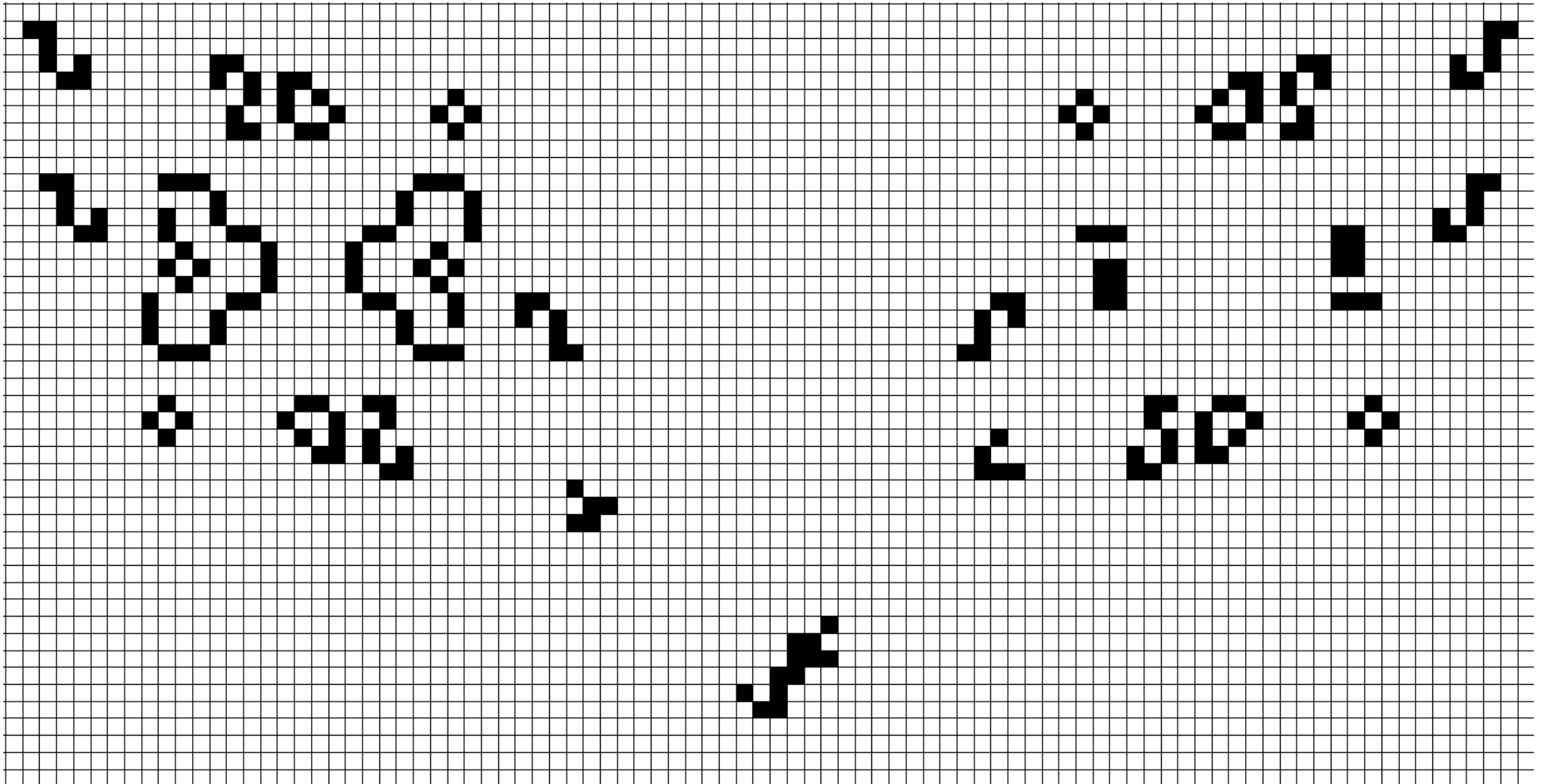
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

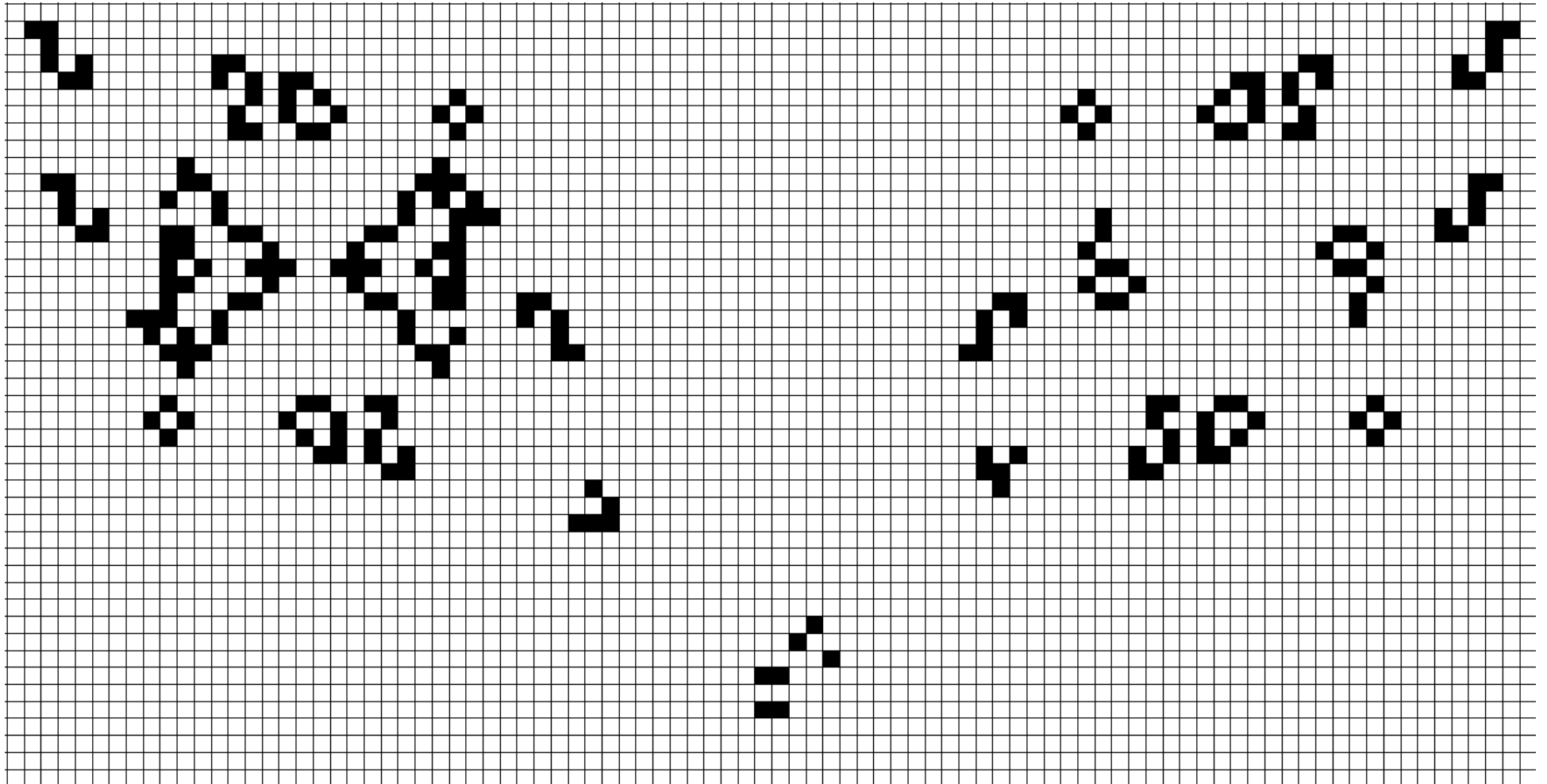
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

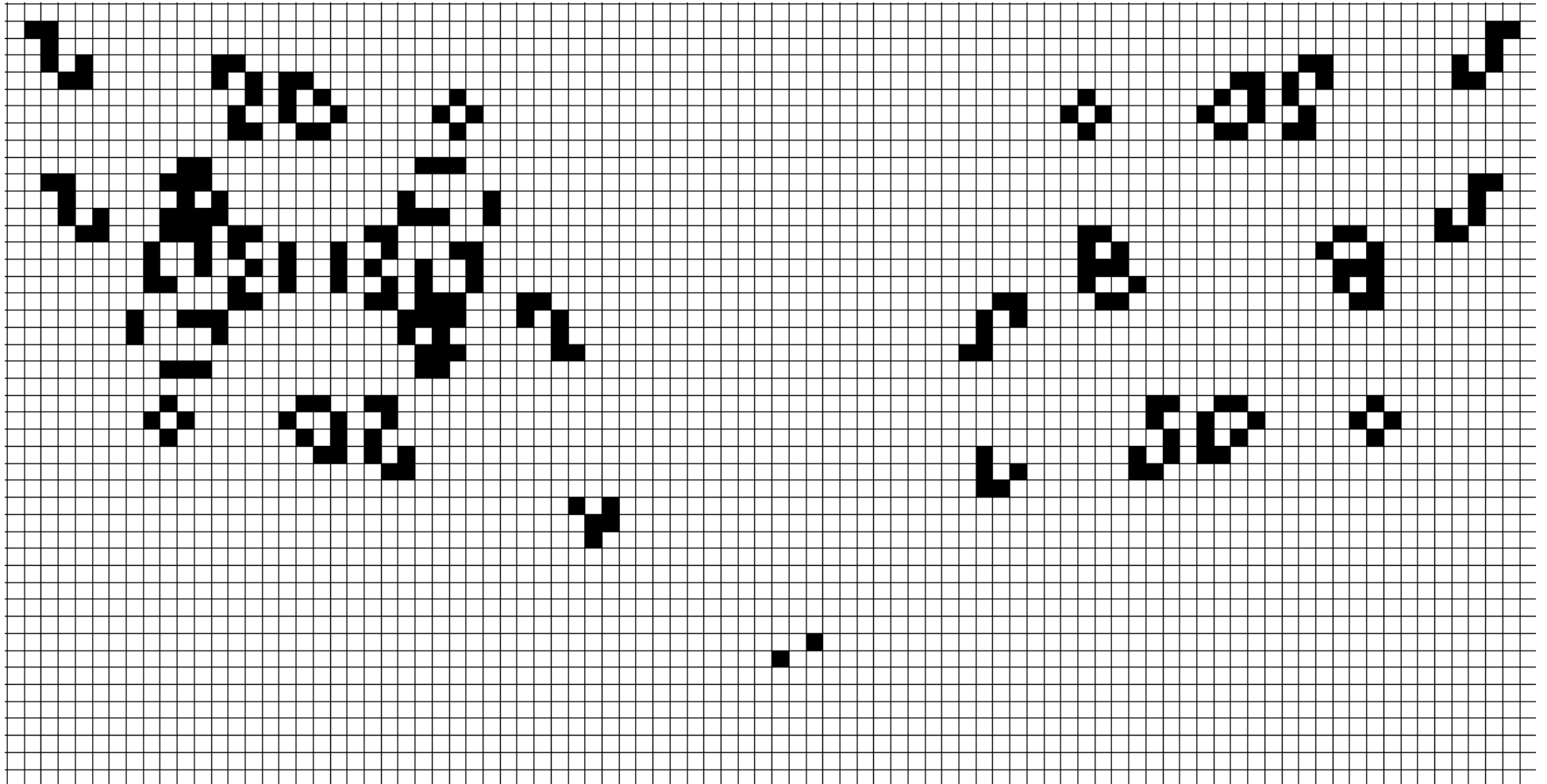
NO : Entrée 1



II. Outils nécessaires et composition du filtre

2. Portes logiques

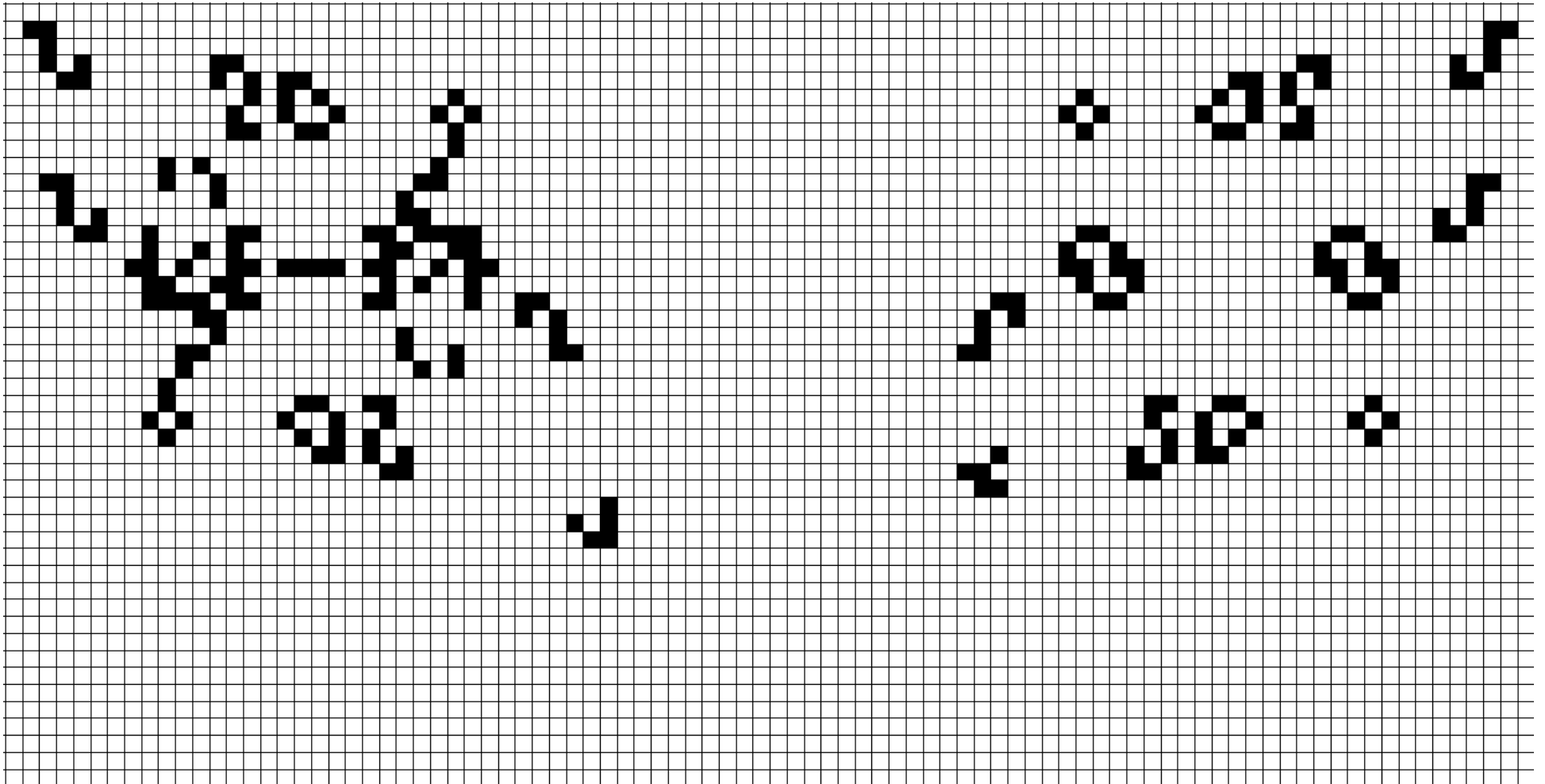
NO : Entrée 1



II. Outils nécessaires et composition du filtre

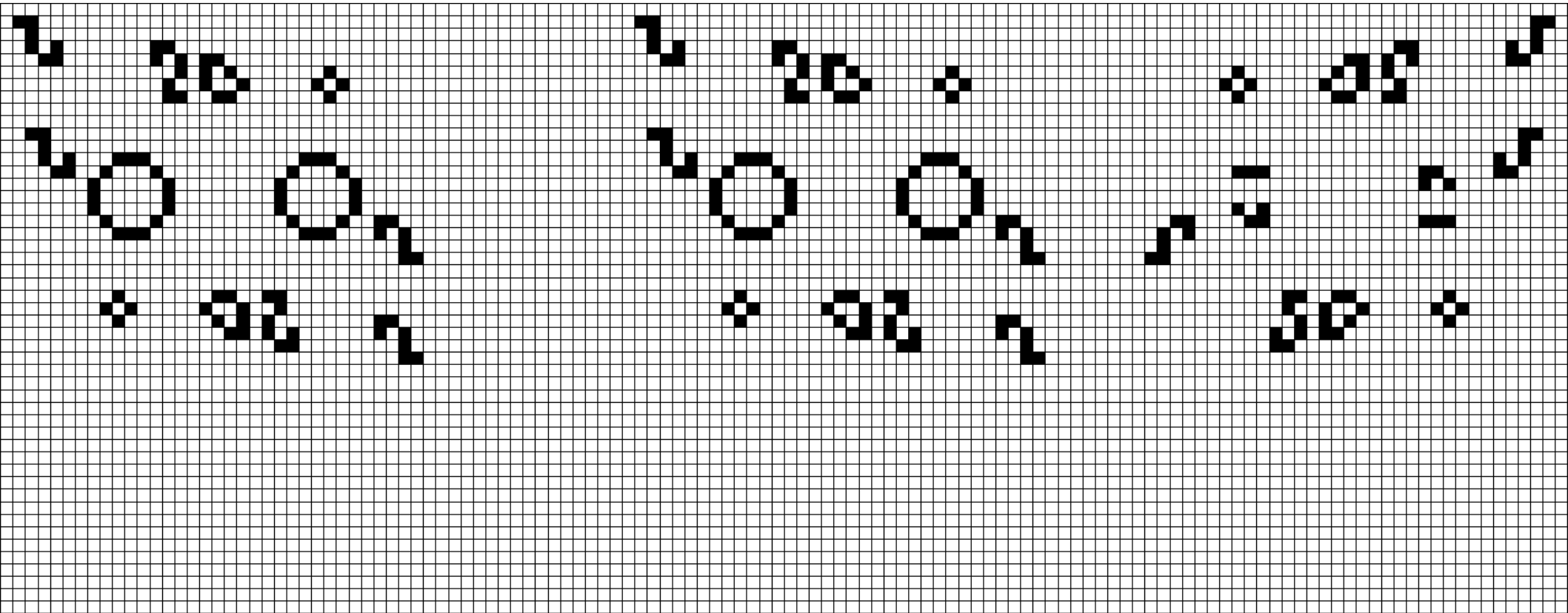
2. Portes logiques

NO : Entrée 1



II. Outils nécessaires et composition du filtre

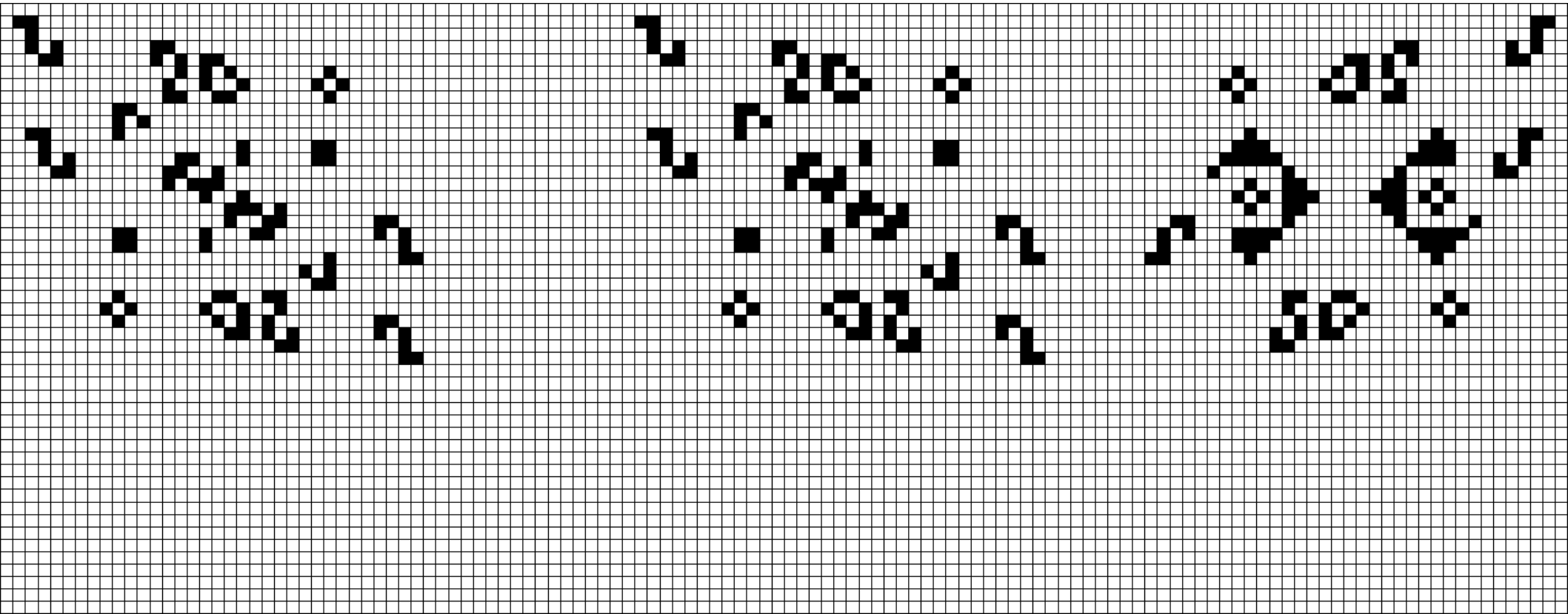
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

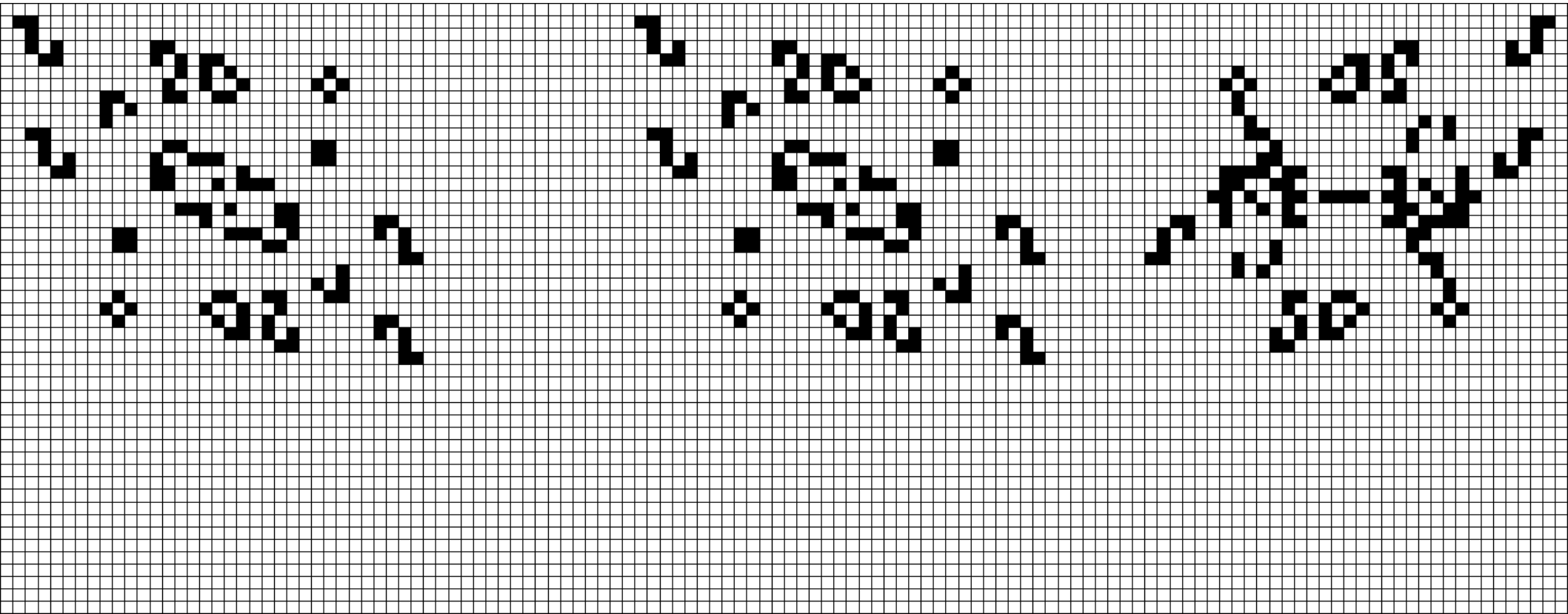
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

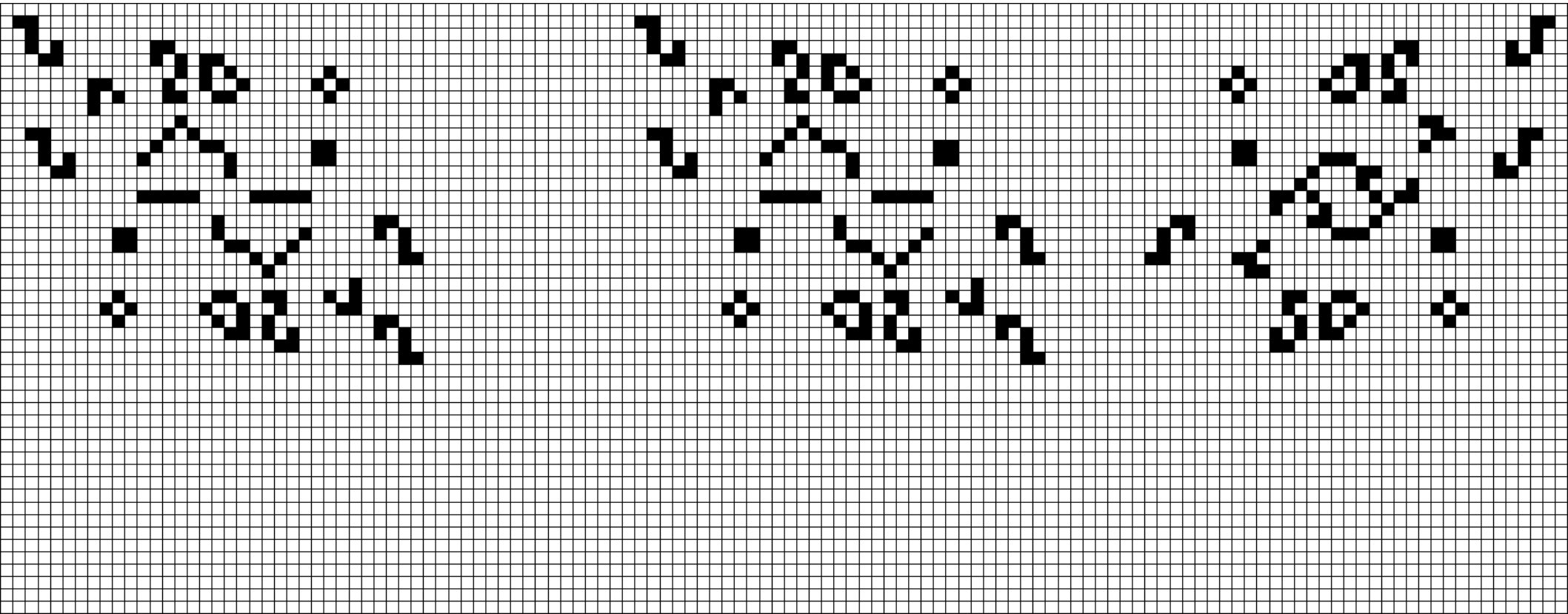
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

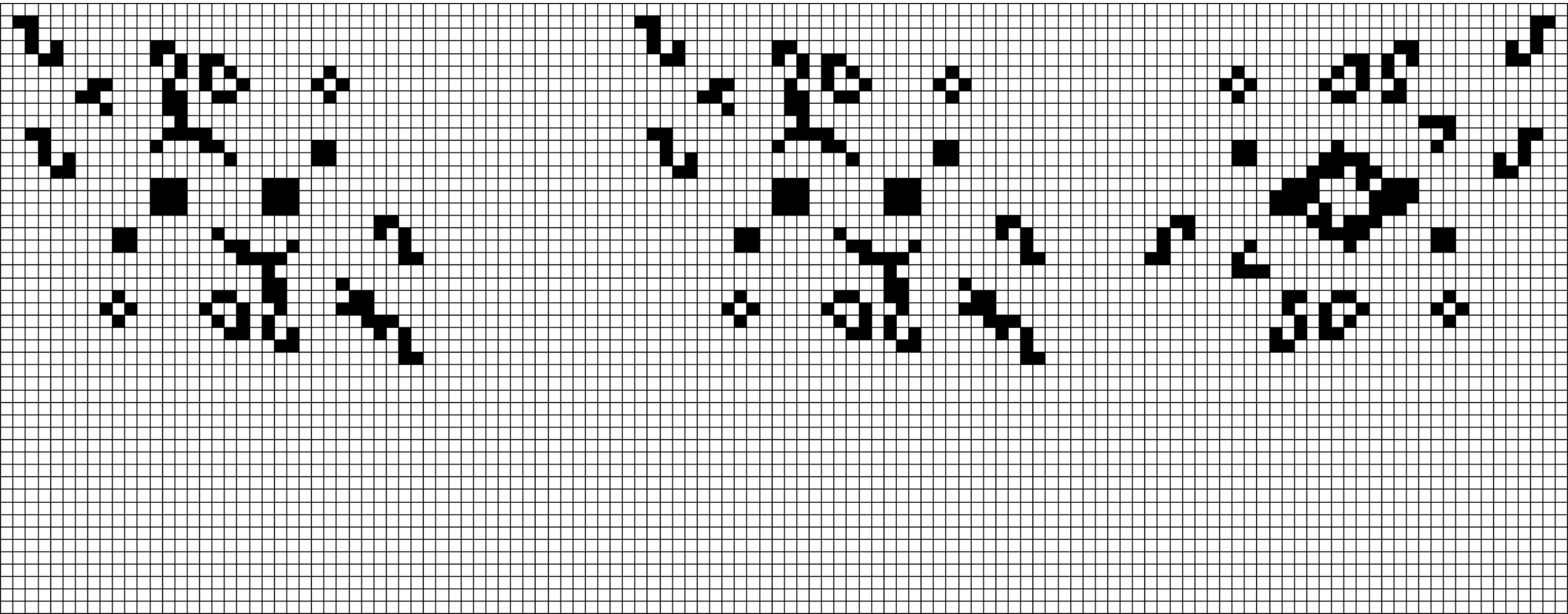
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

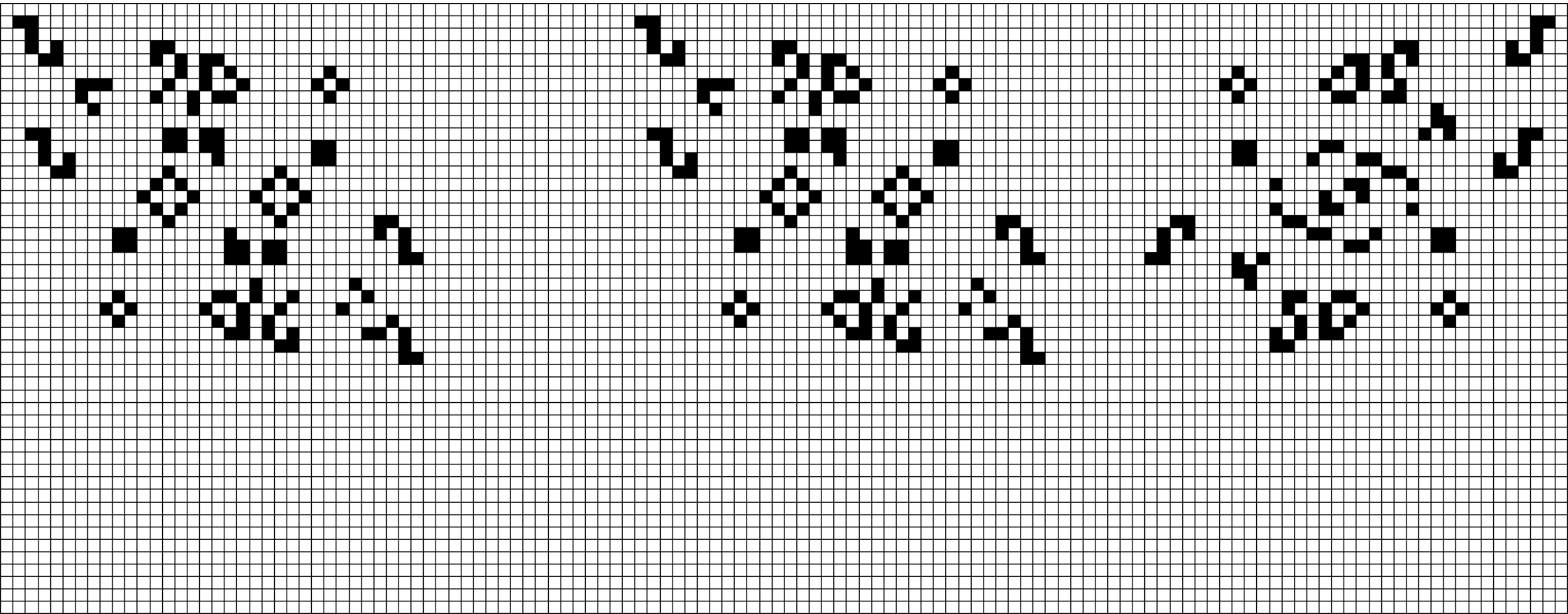
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

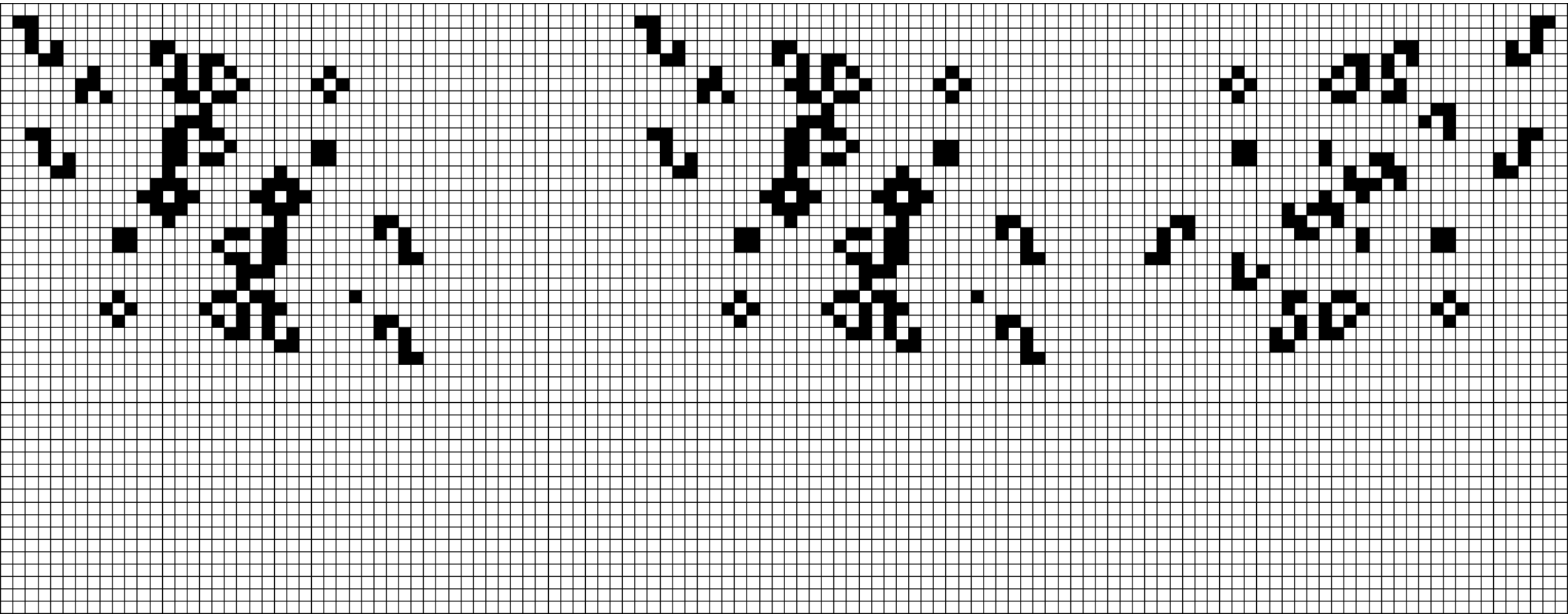
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

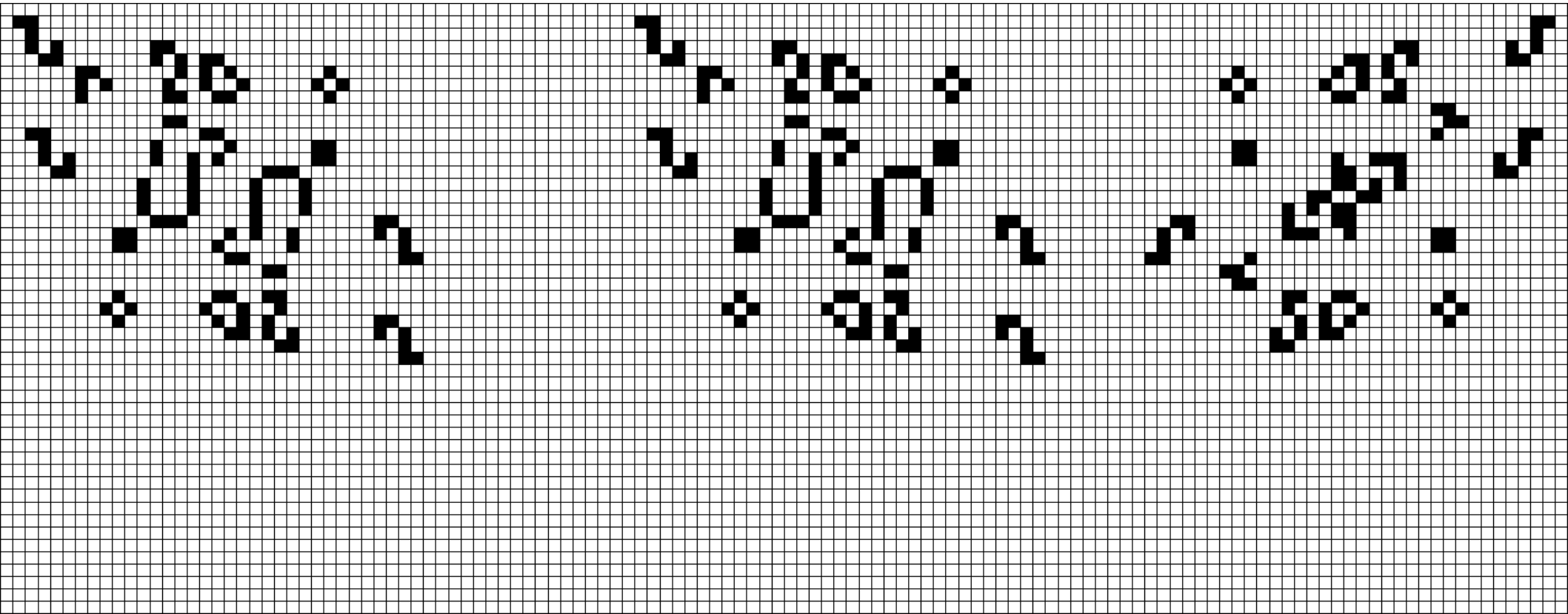
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

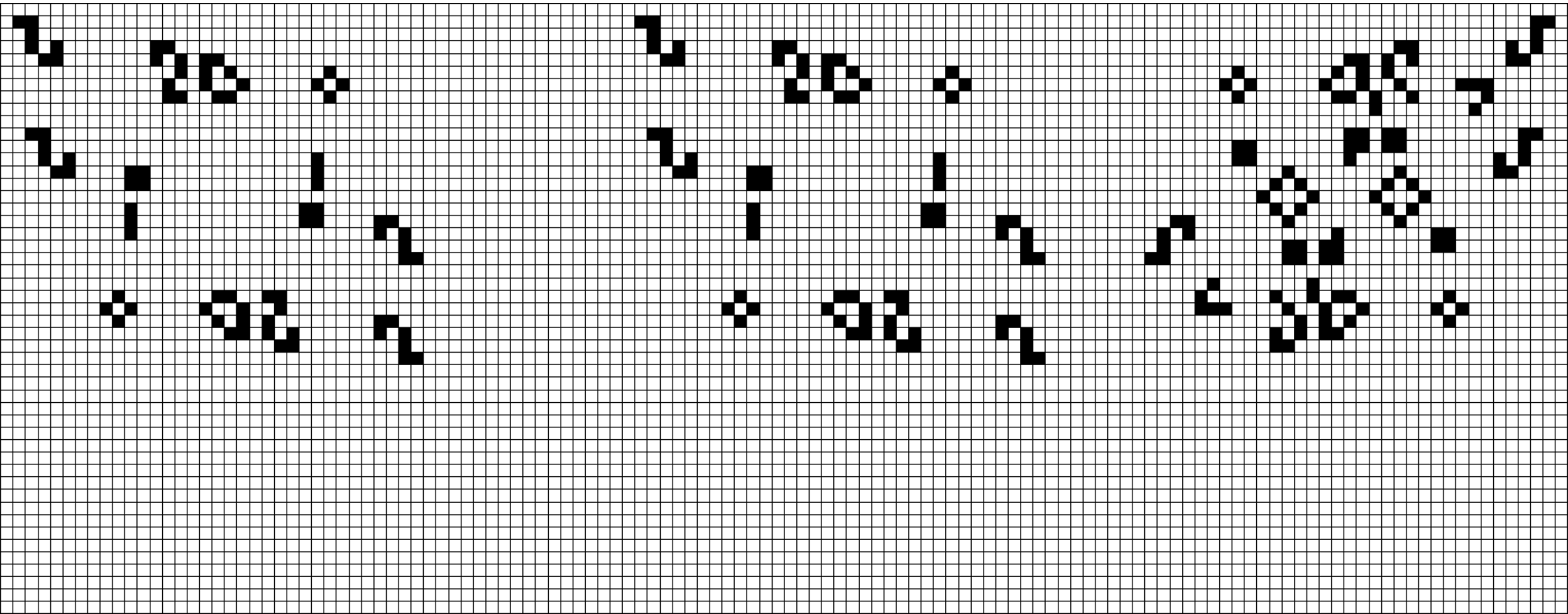
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

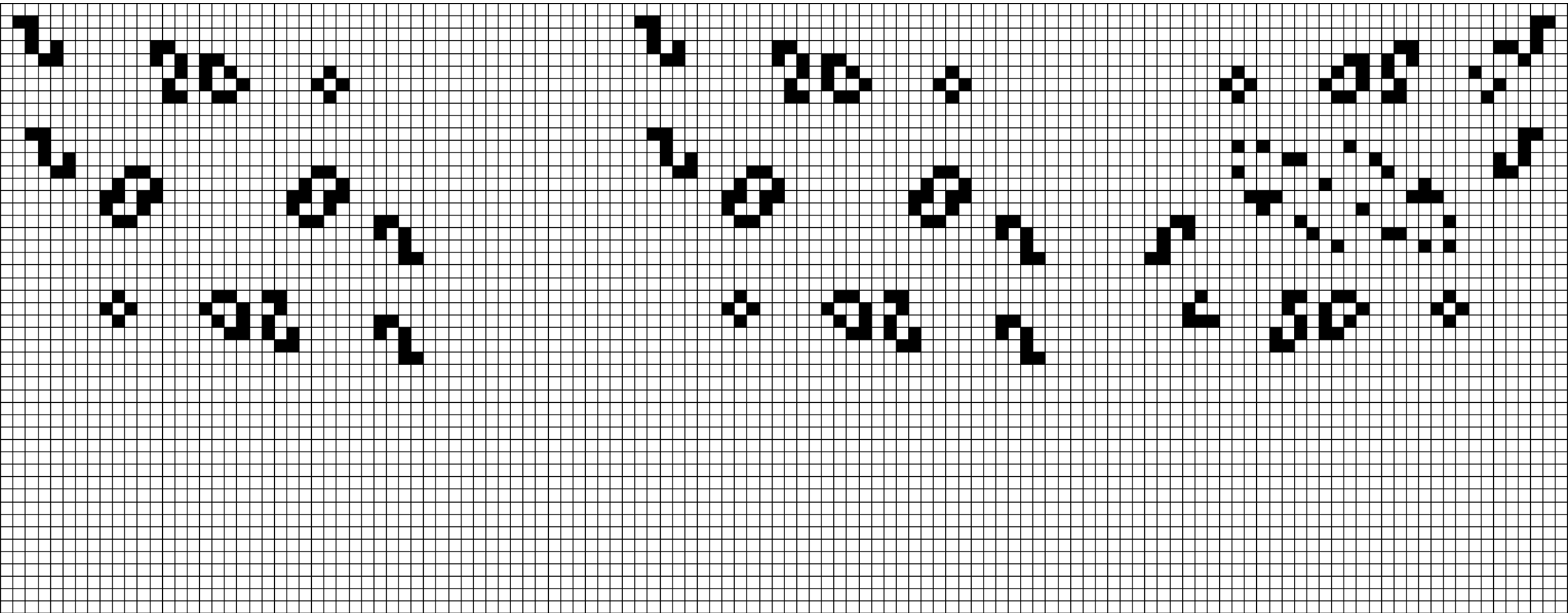
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

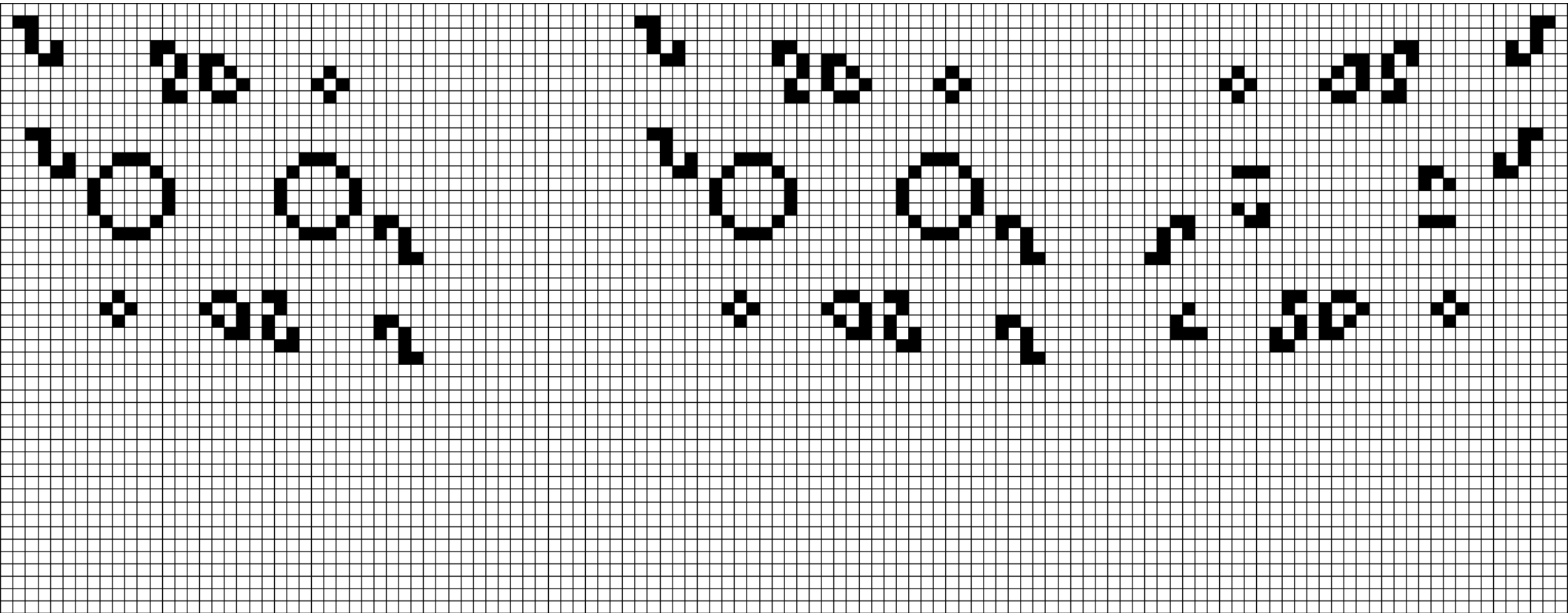
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

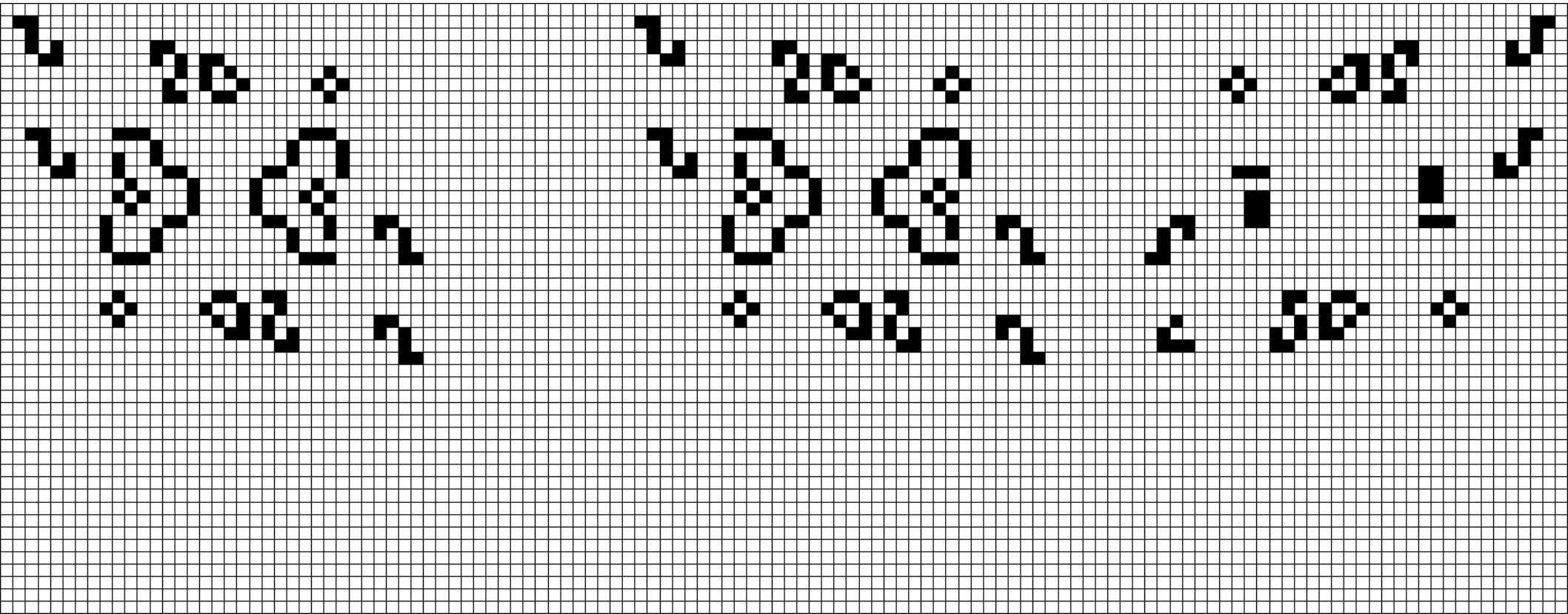
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

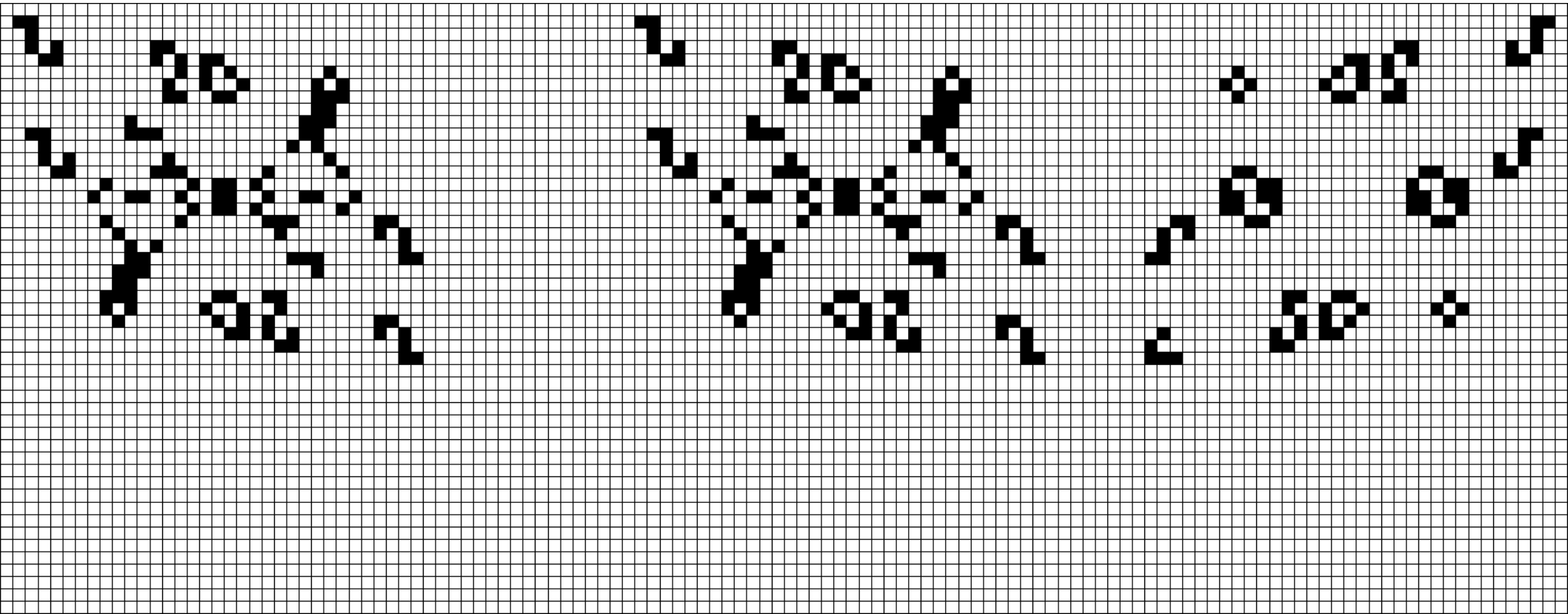
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

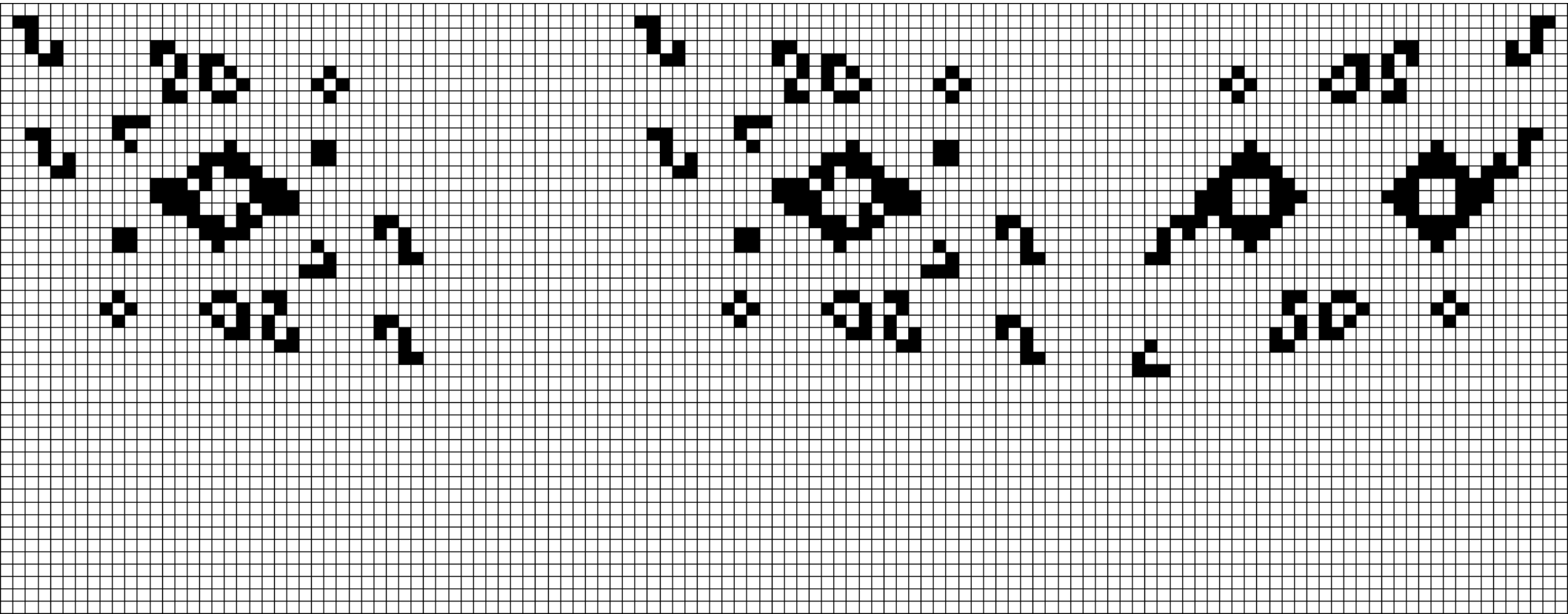
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

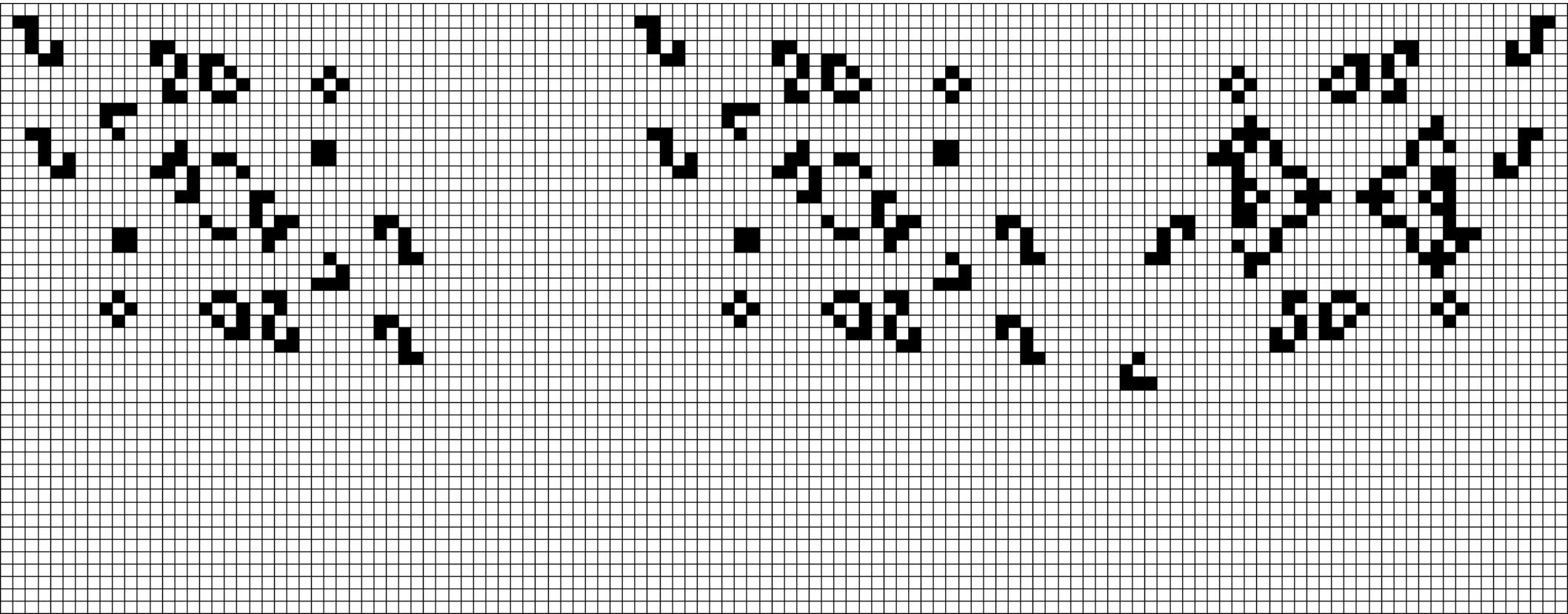
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

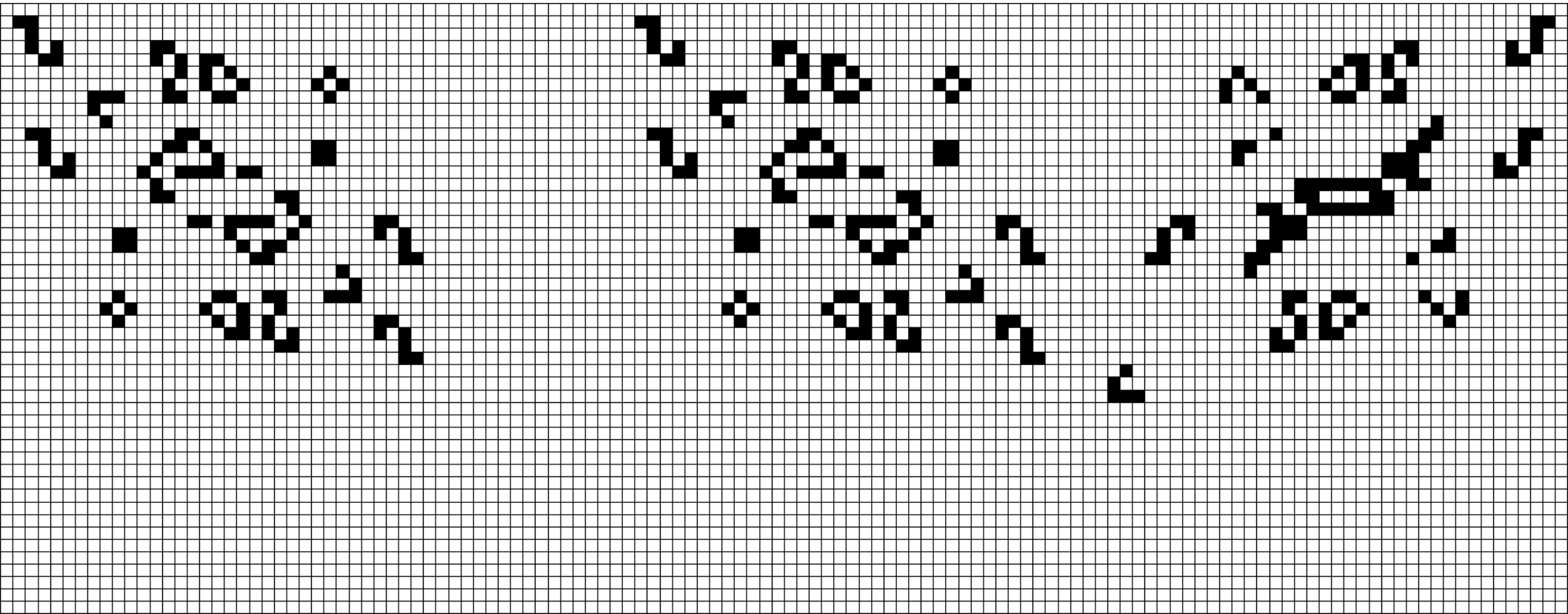
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

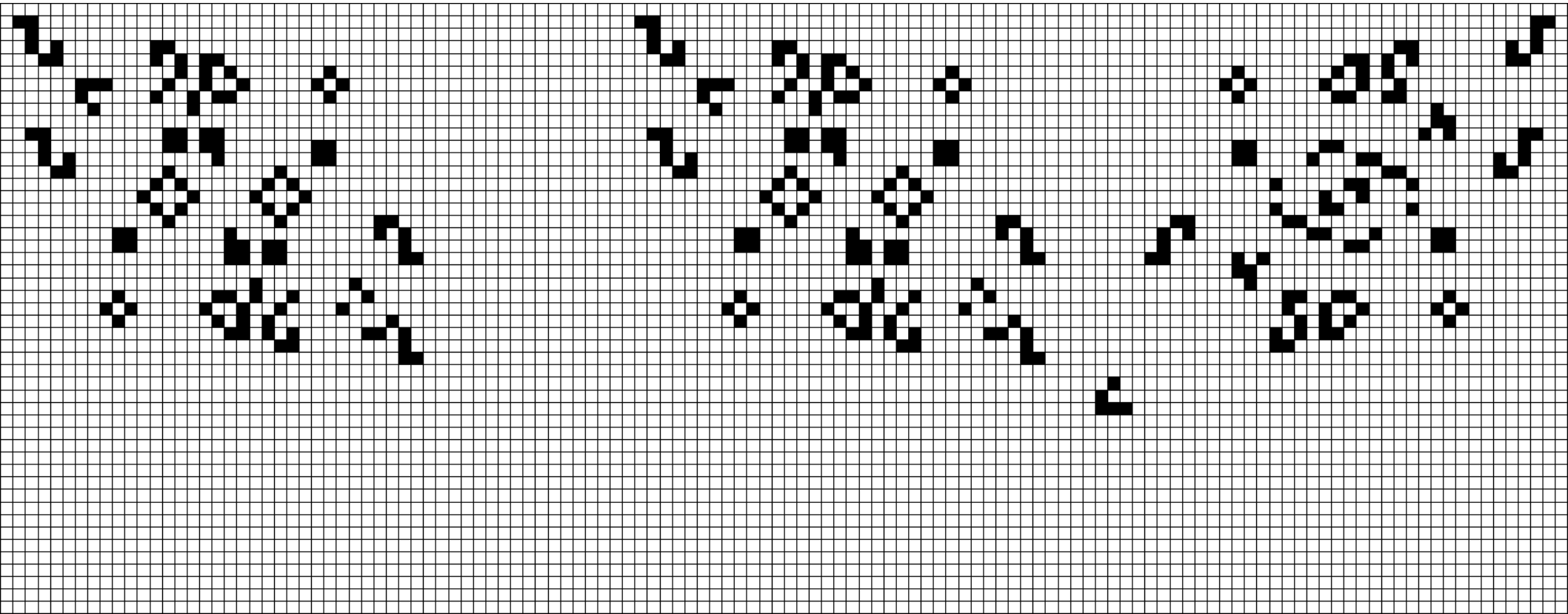
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

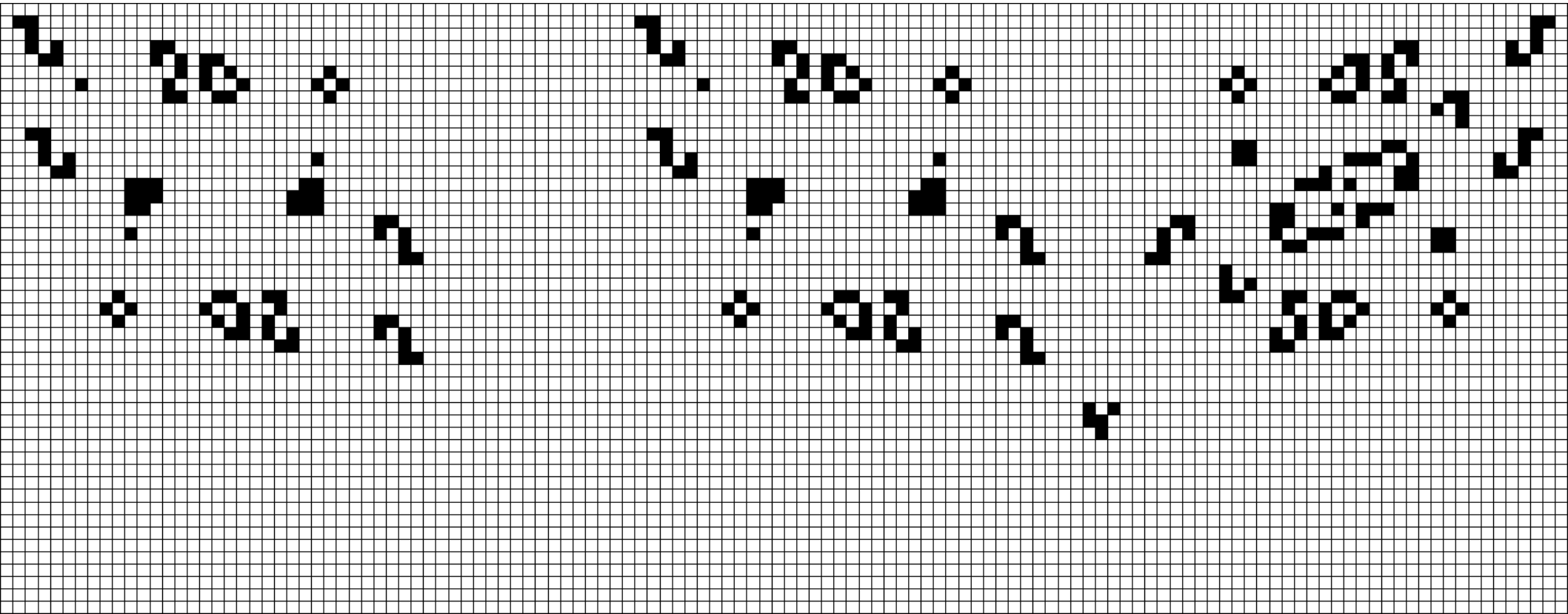
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

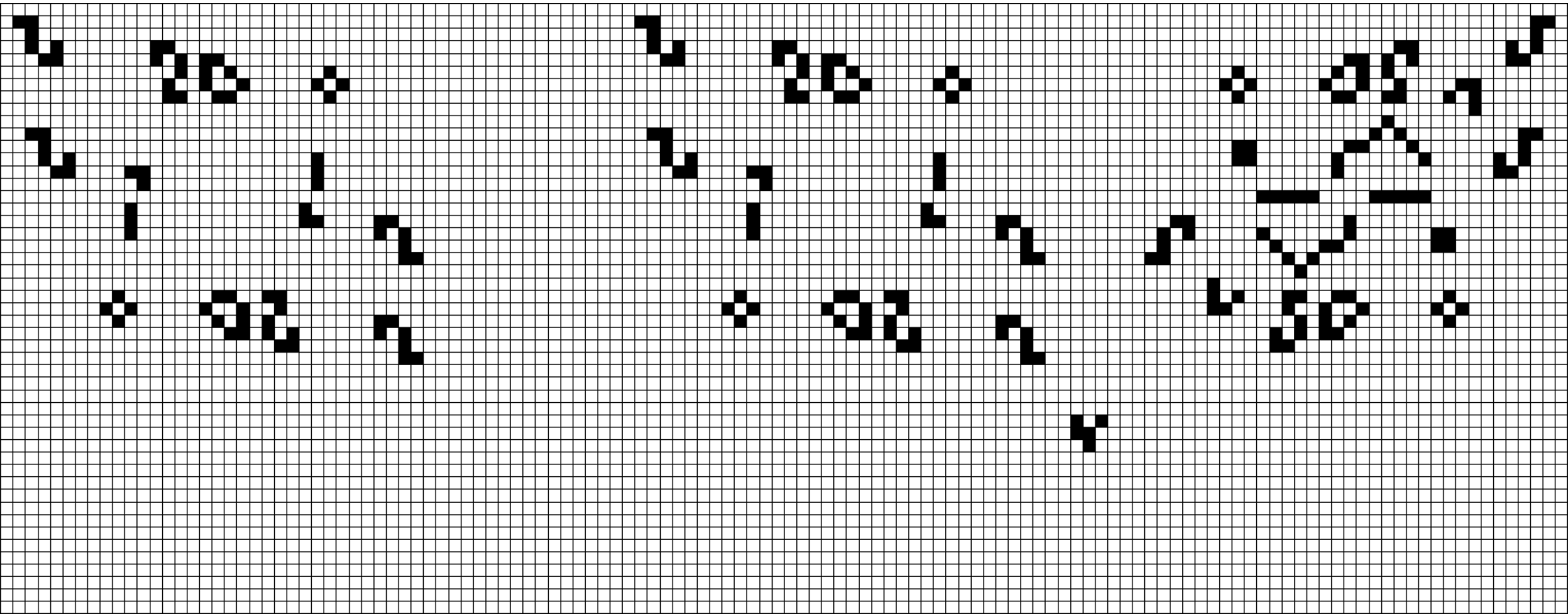
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

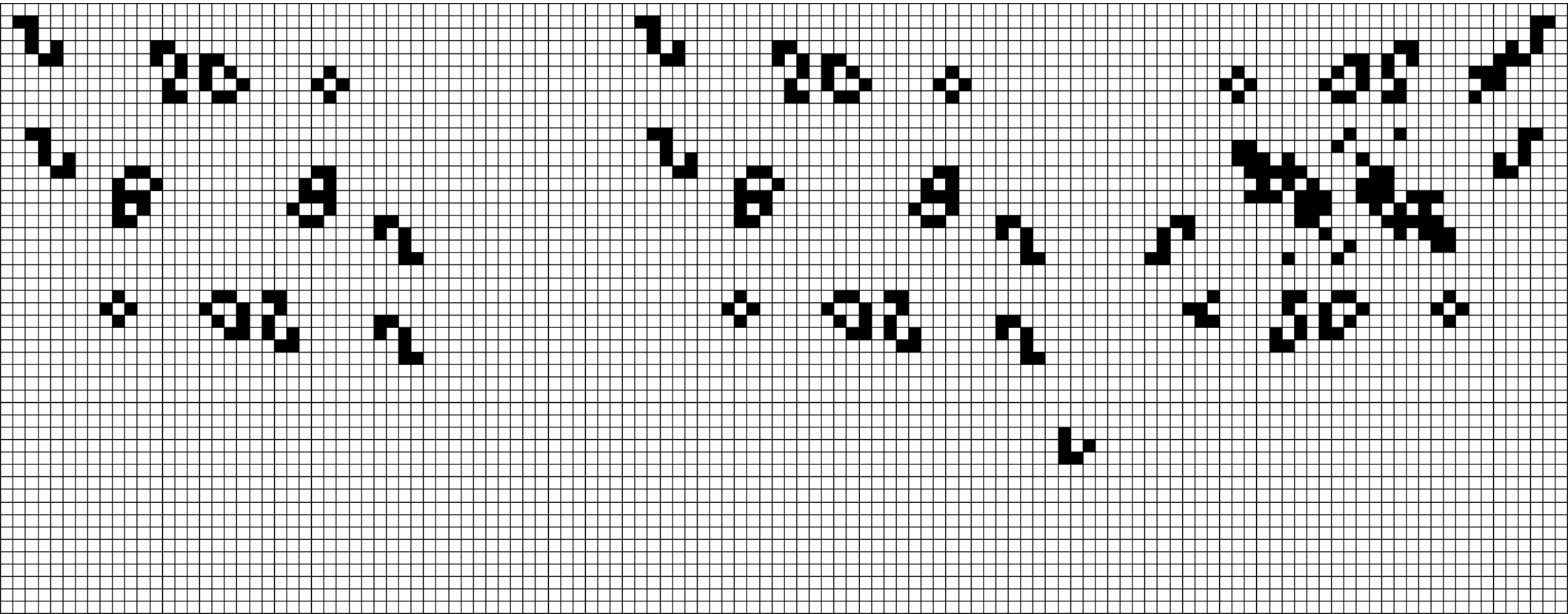
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

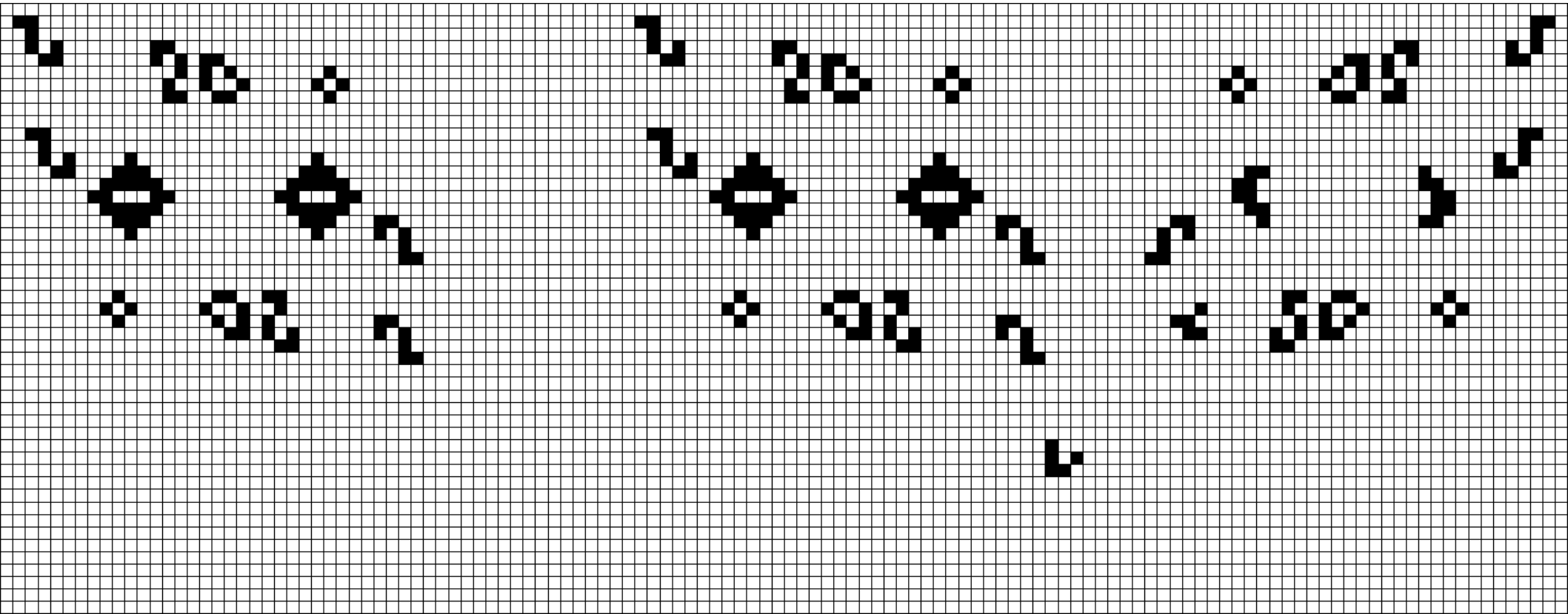
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

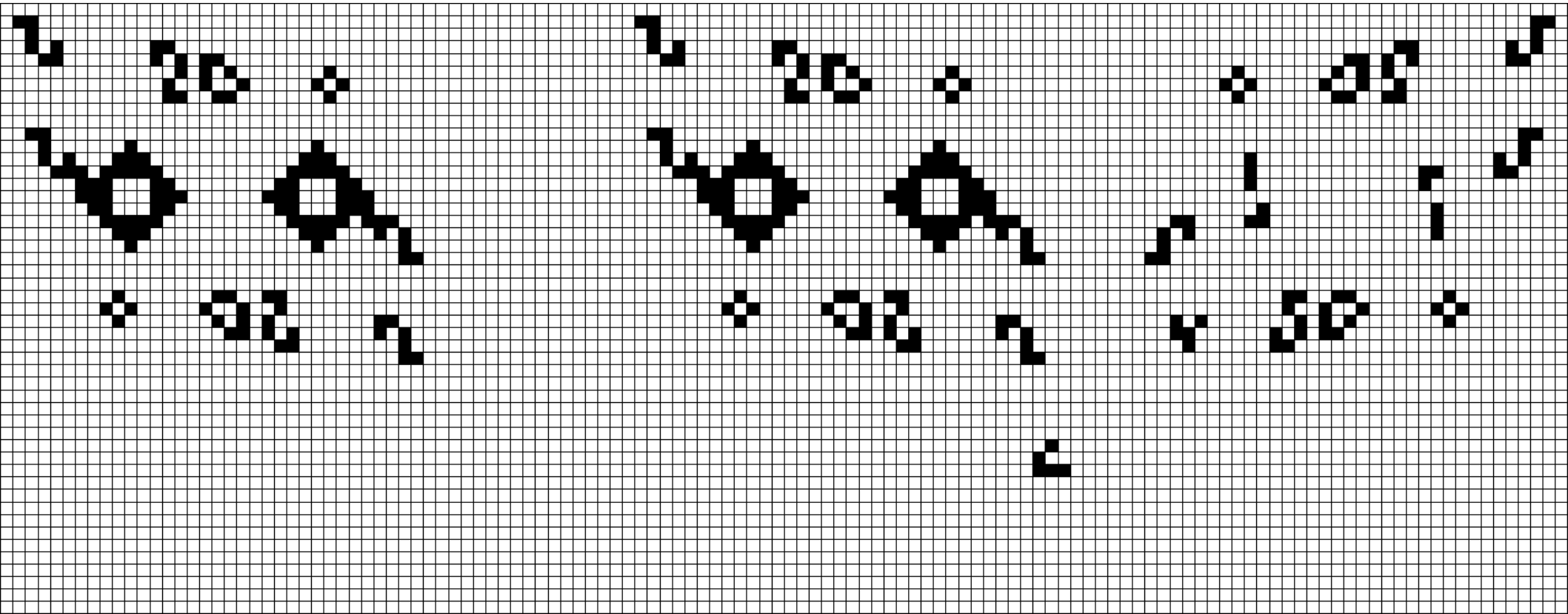
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

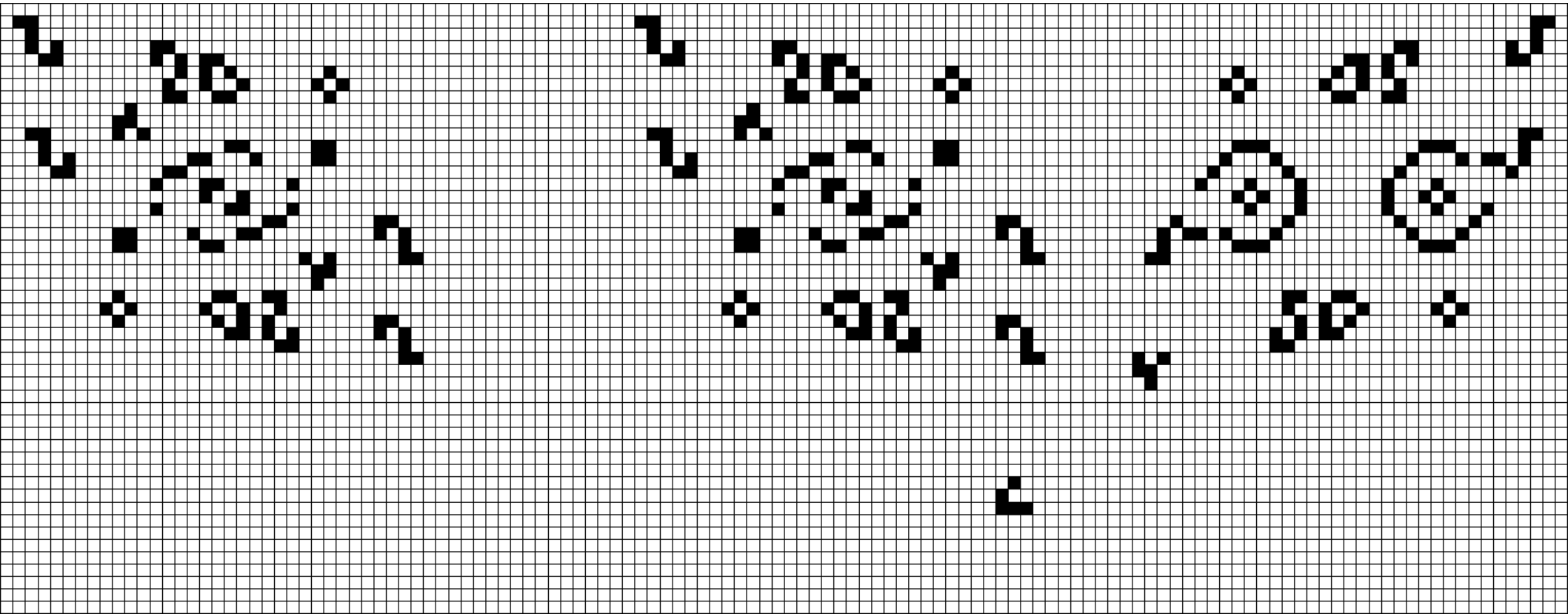
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

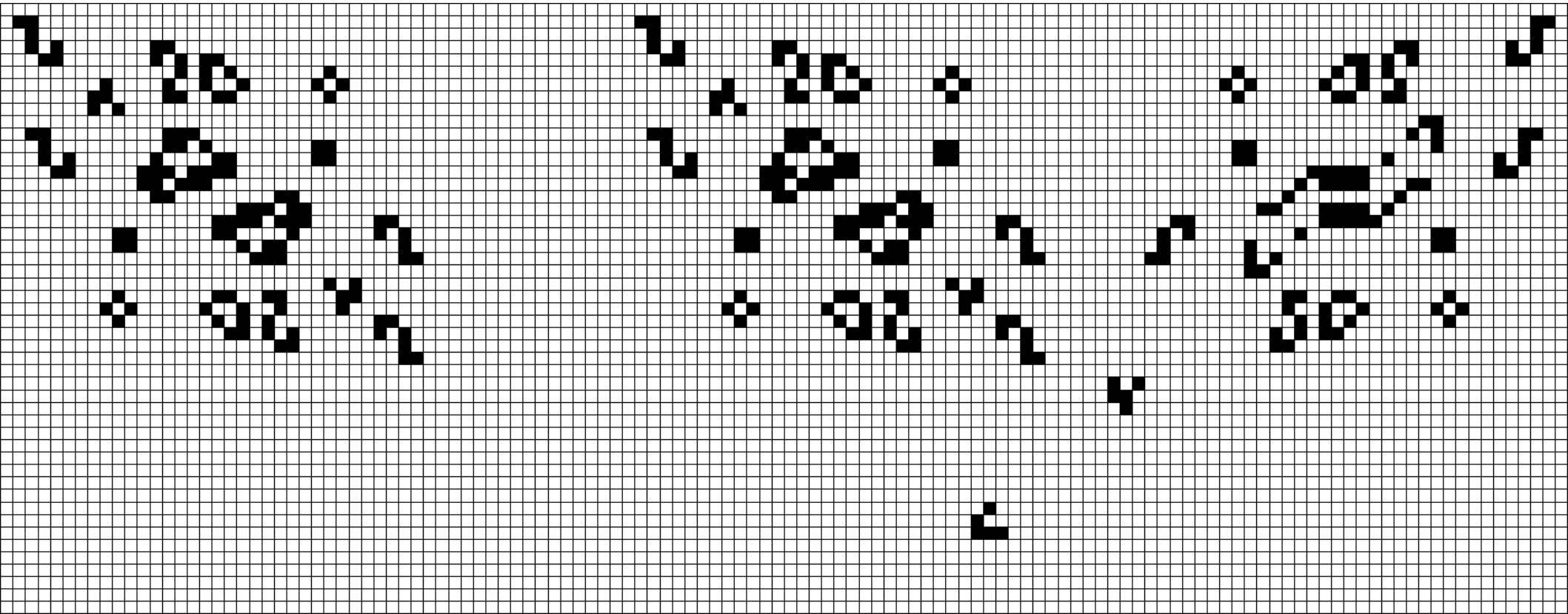
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

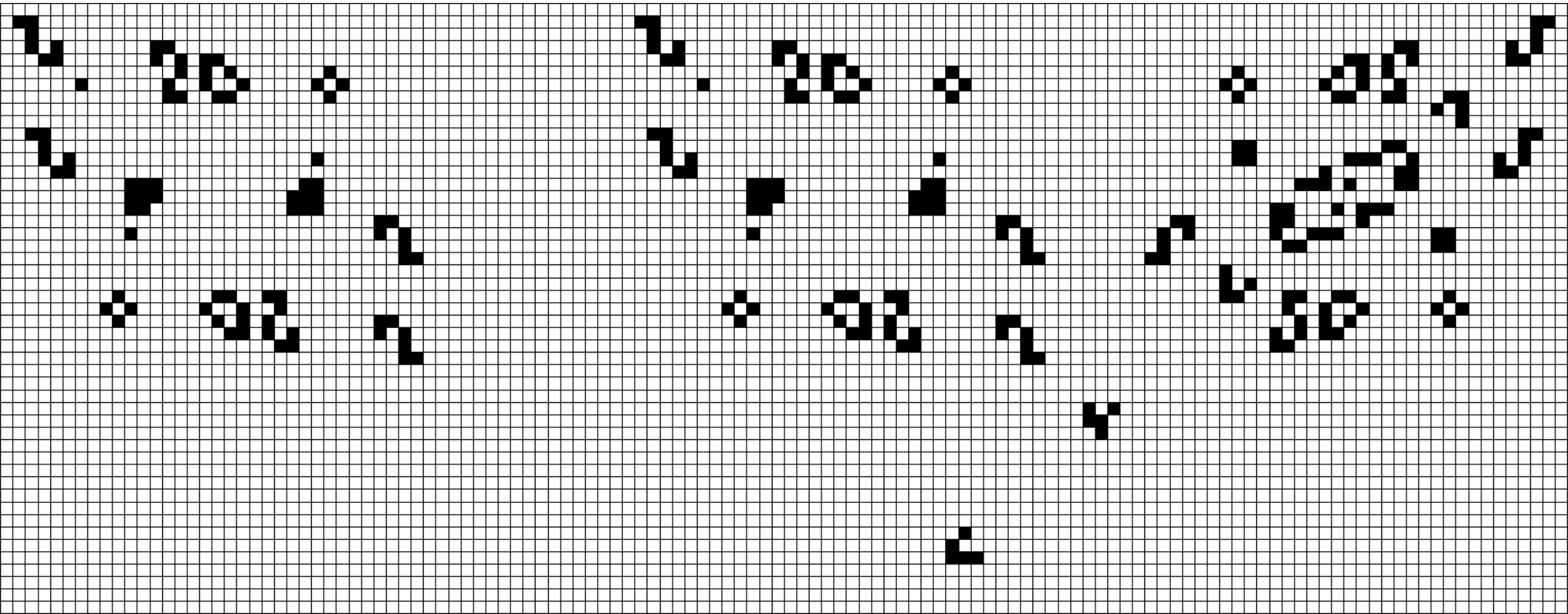
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

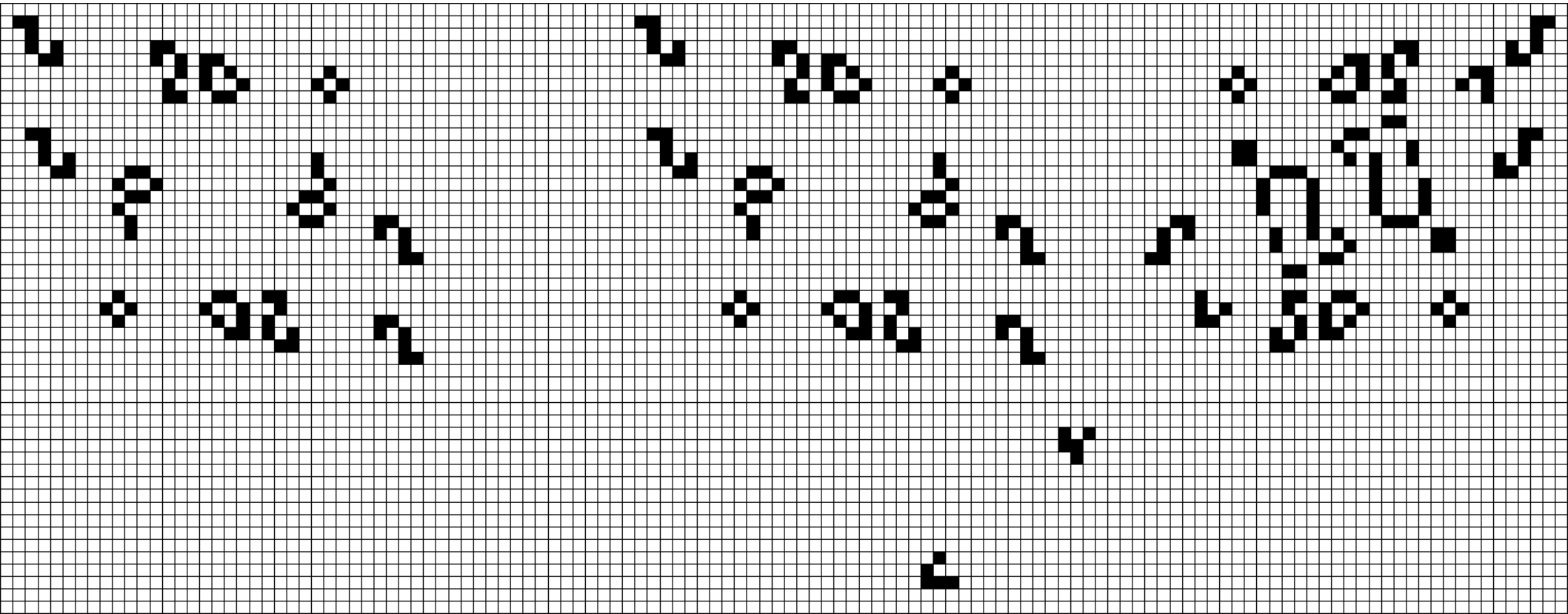
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

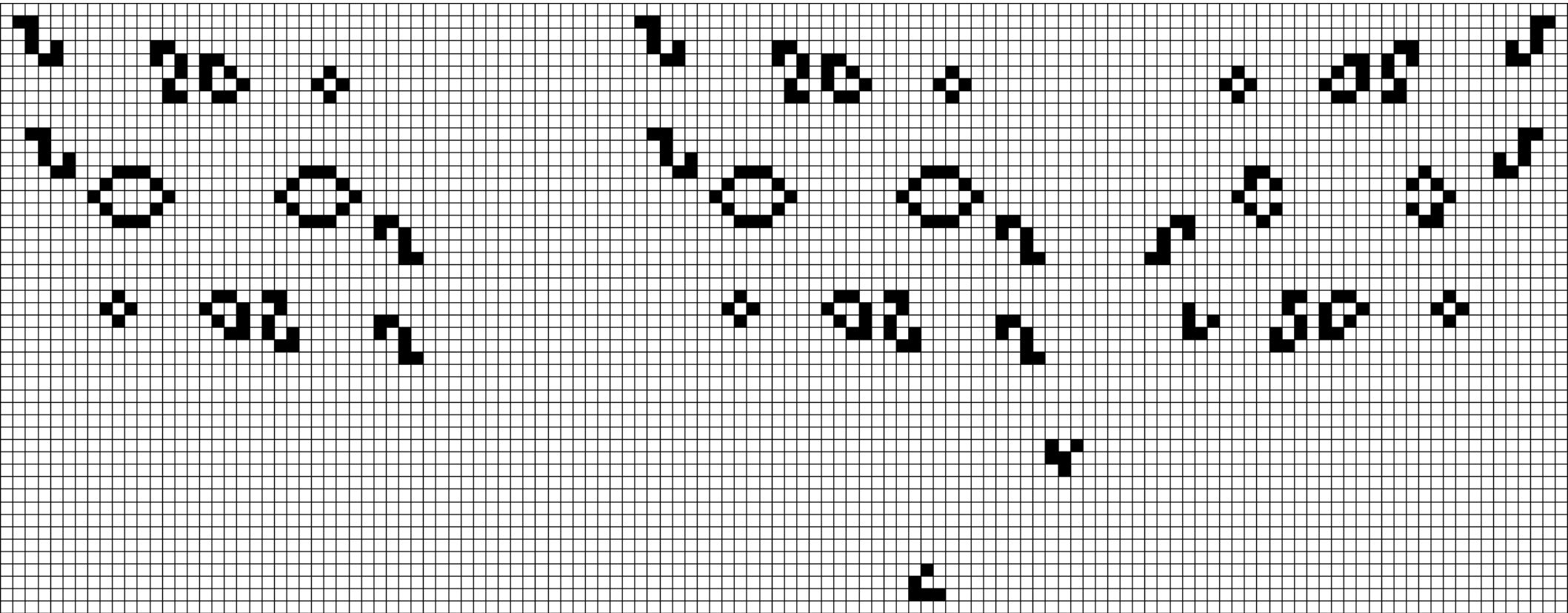
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

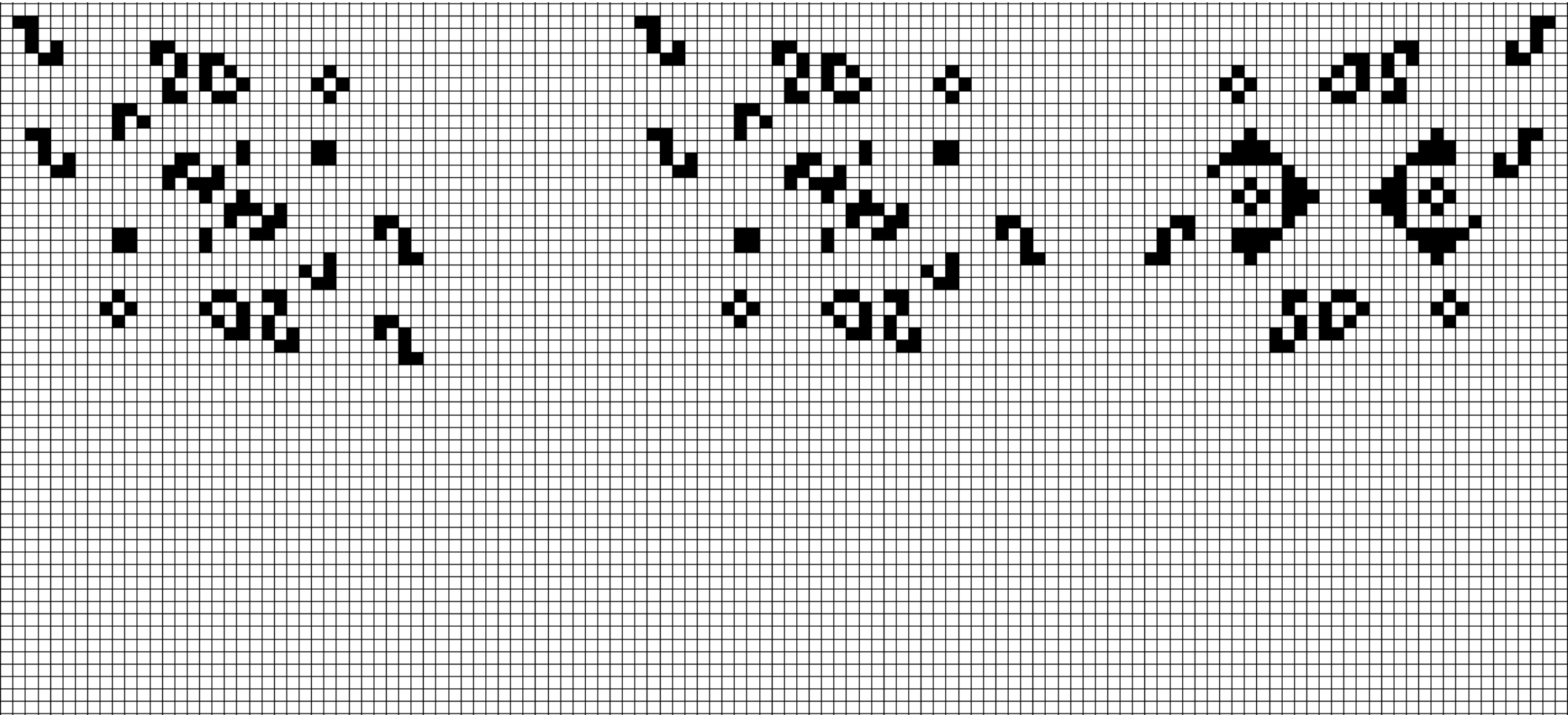
2. Portes logiques



AND : Entrée 0 0

II. Outils nécessaires et composition du filtre

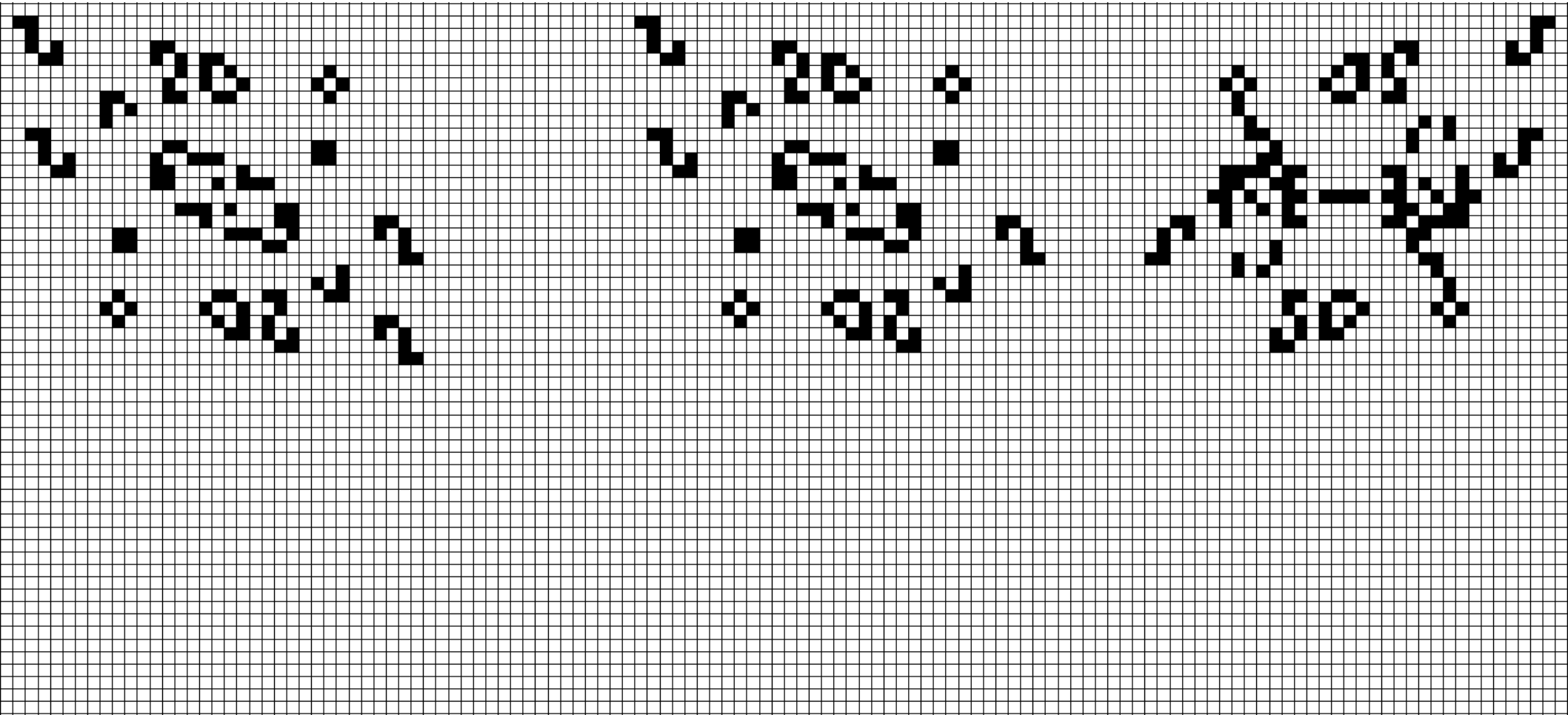
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

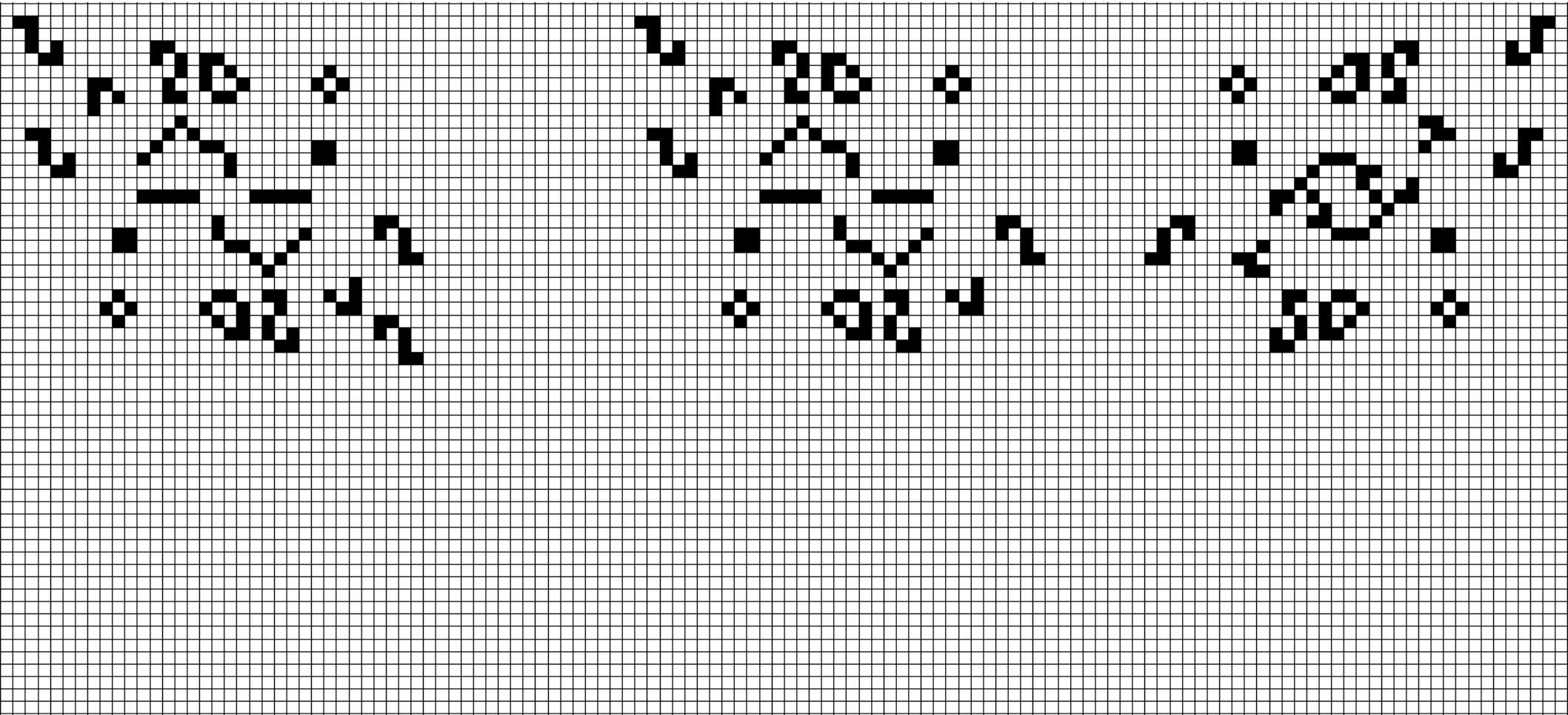
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

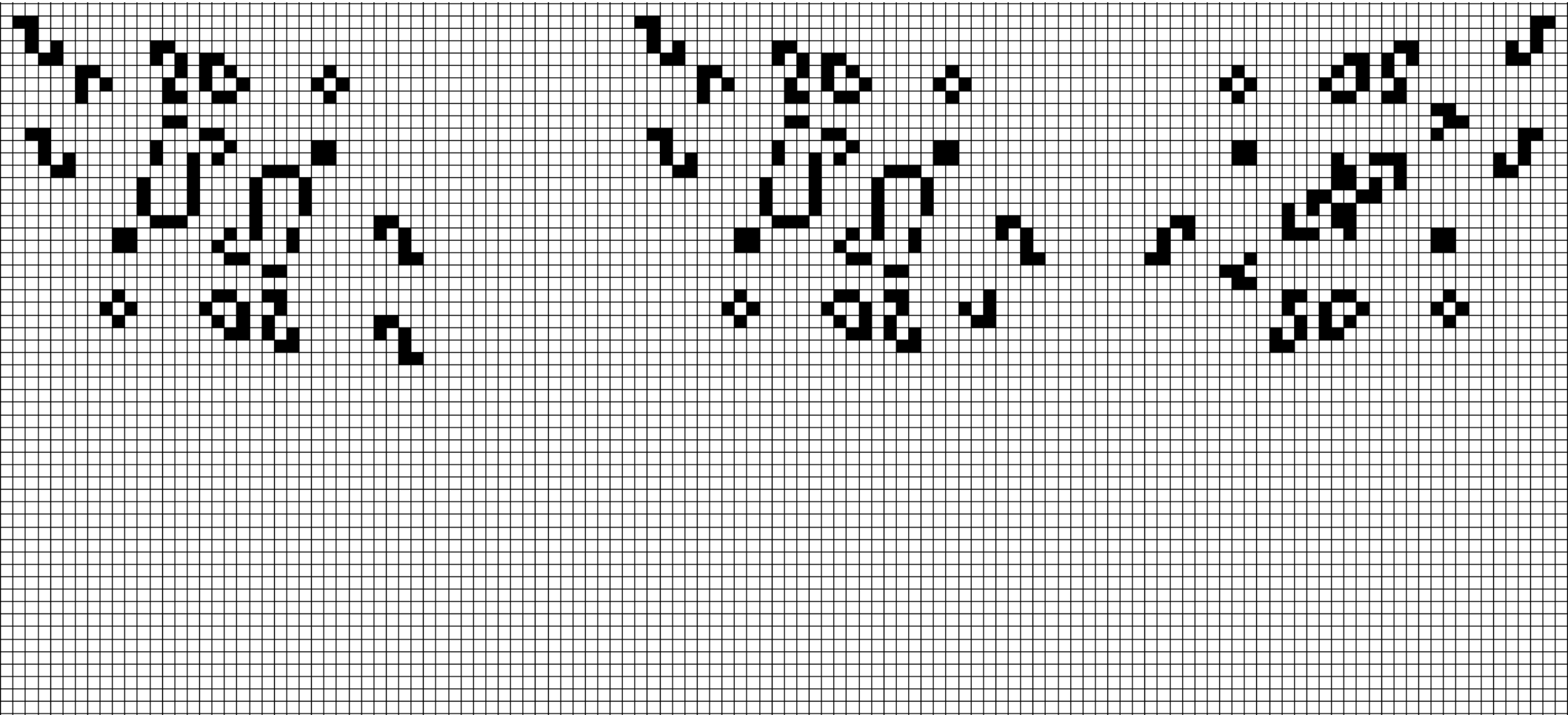
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

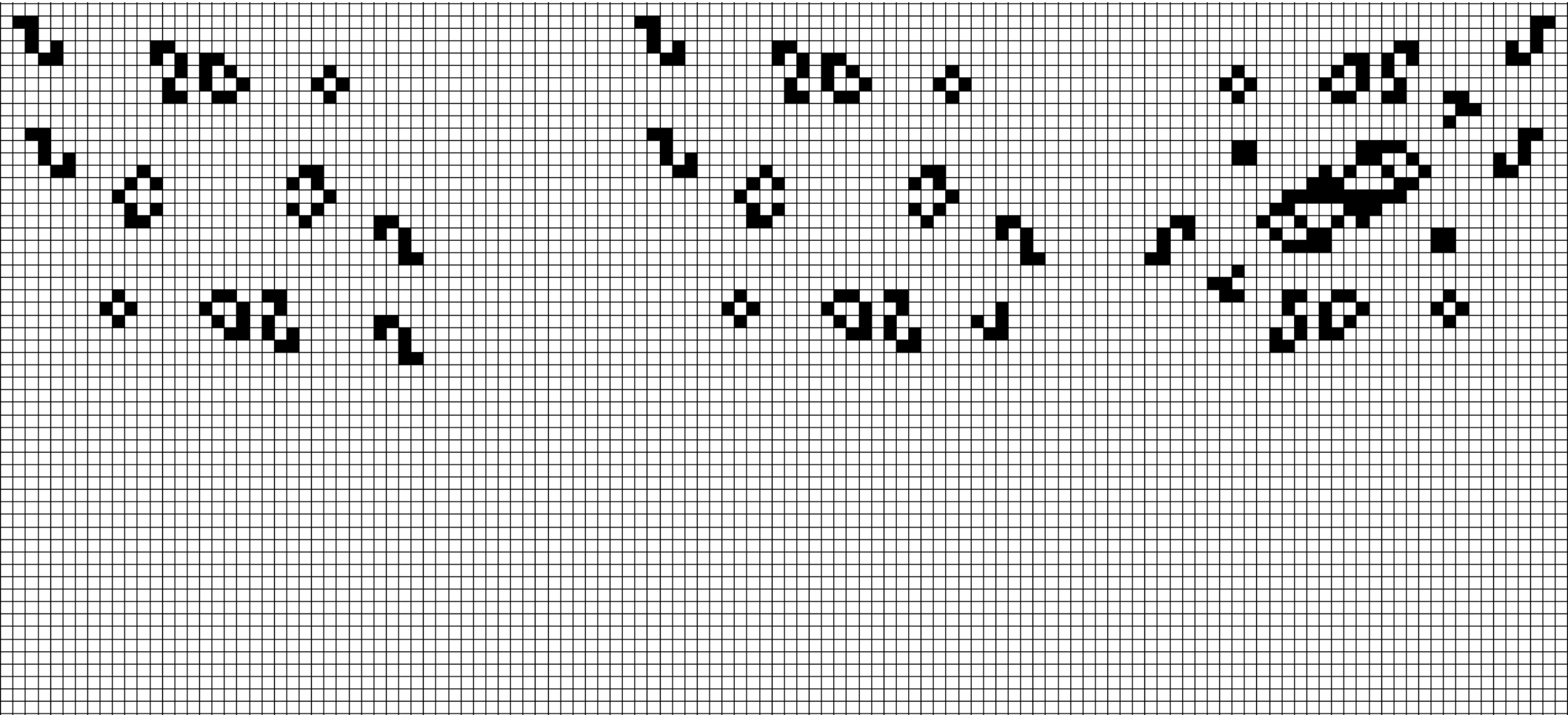
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

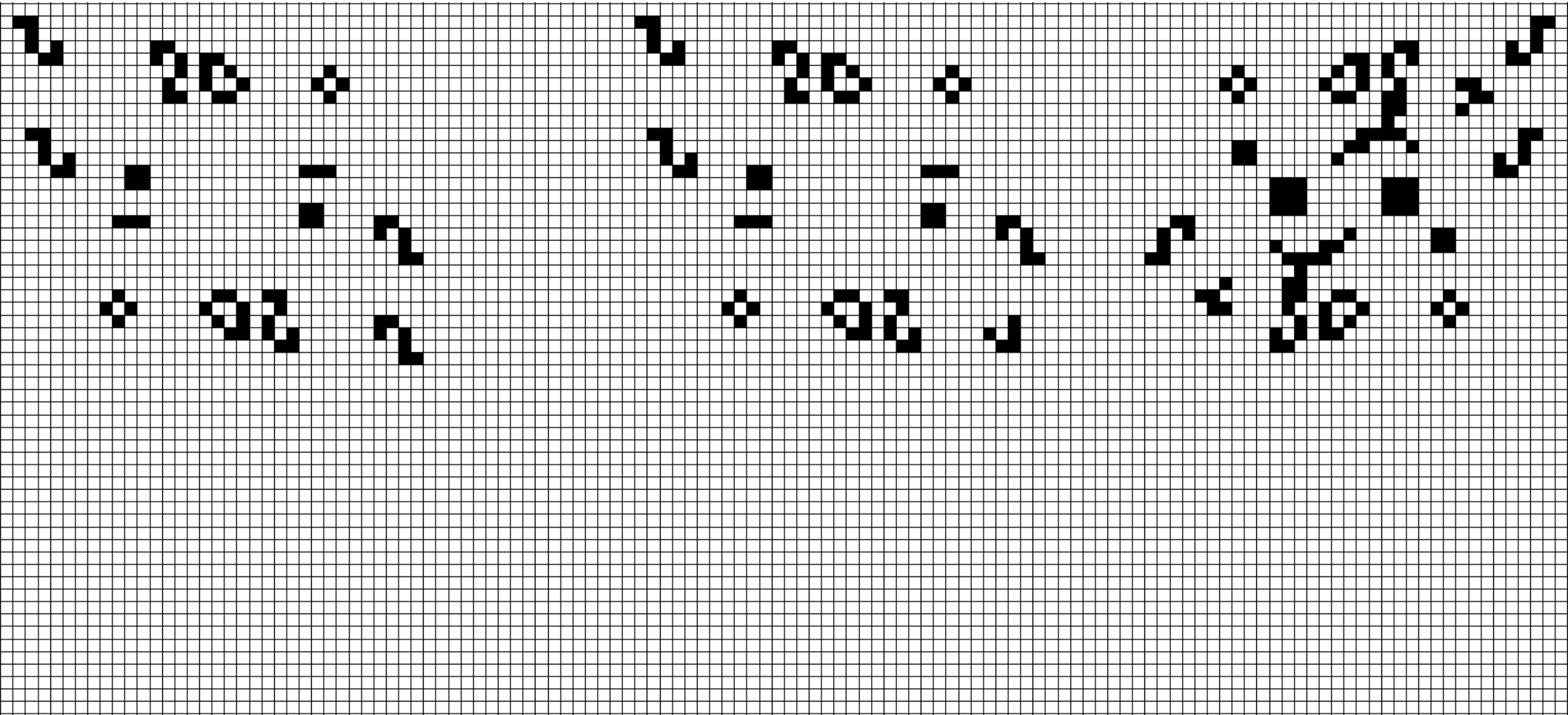
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

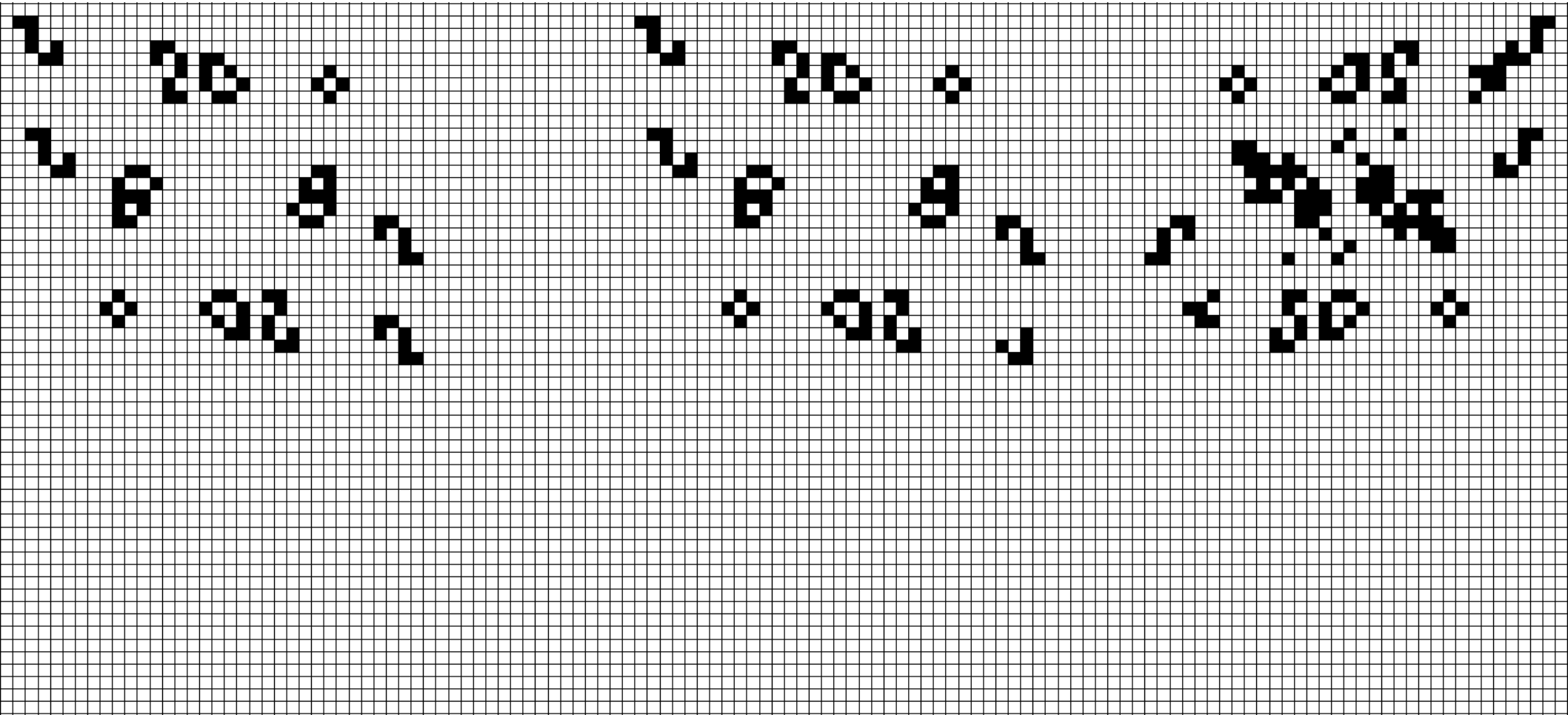
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

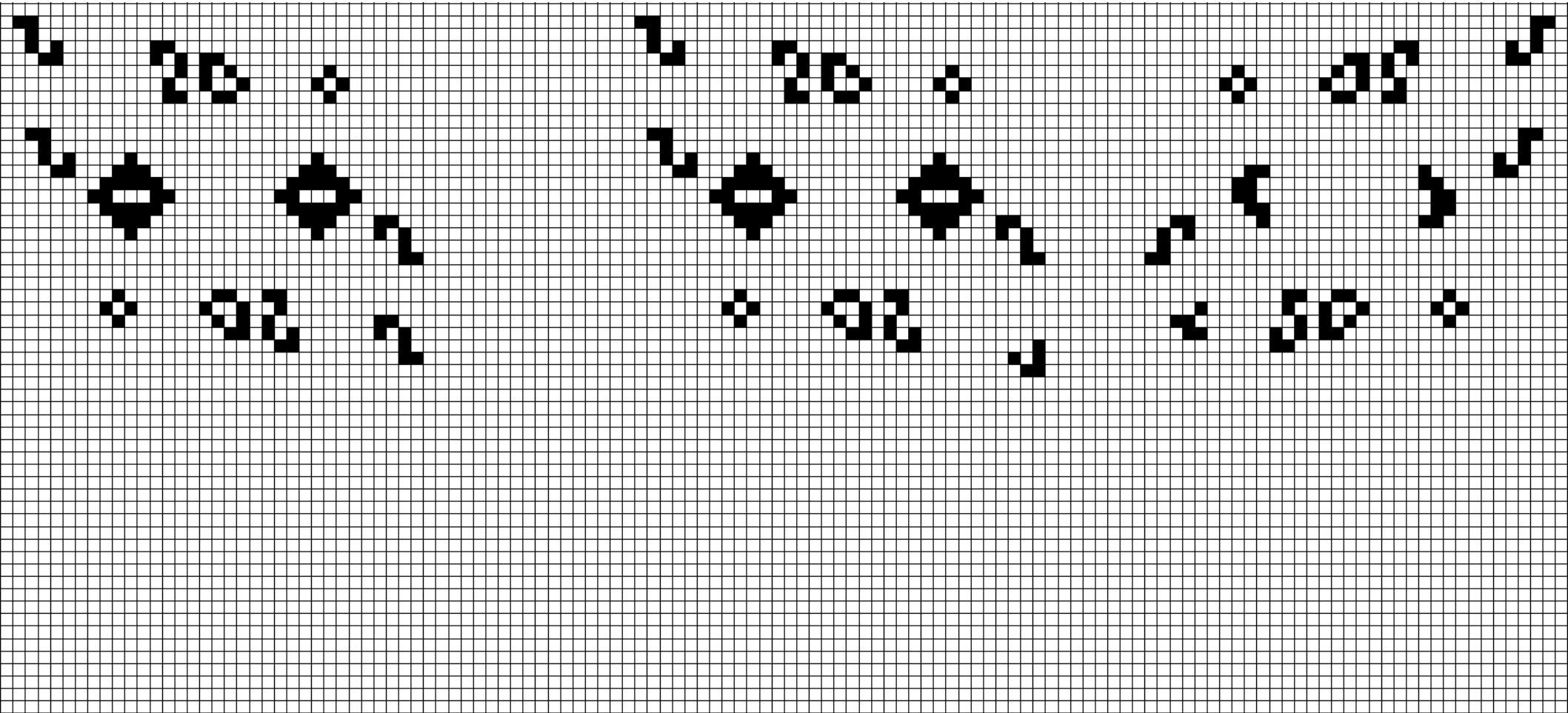
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

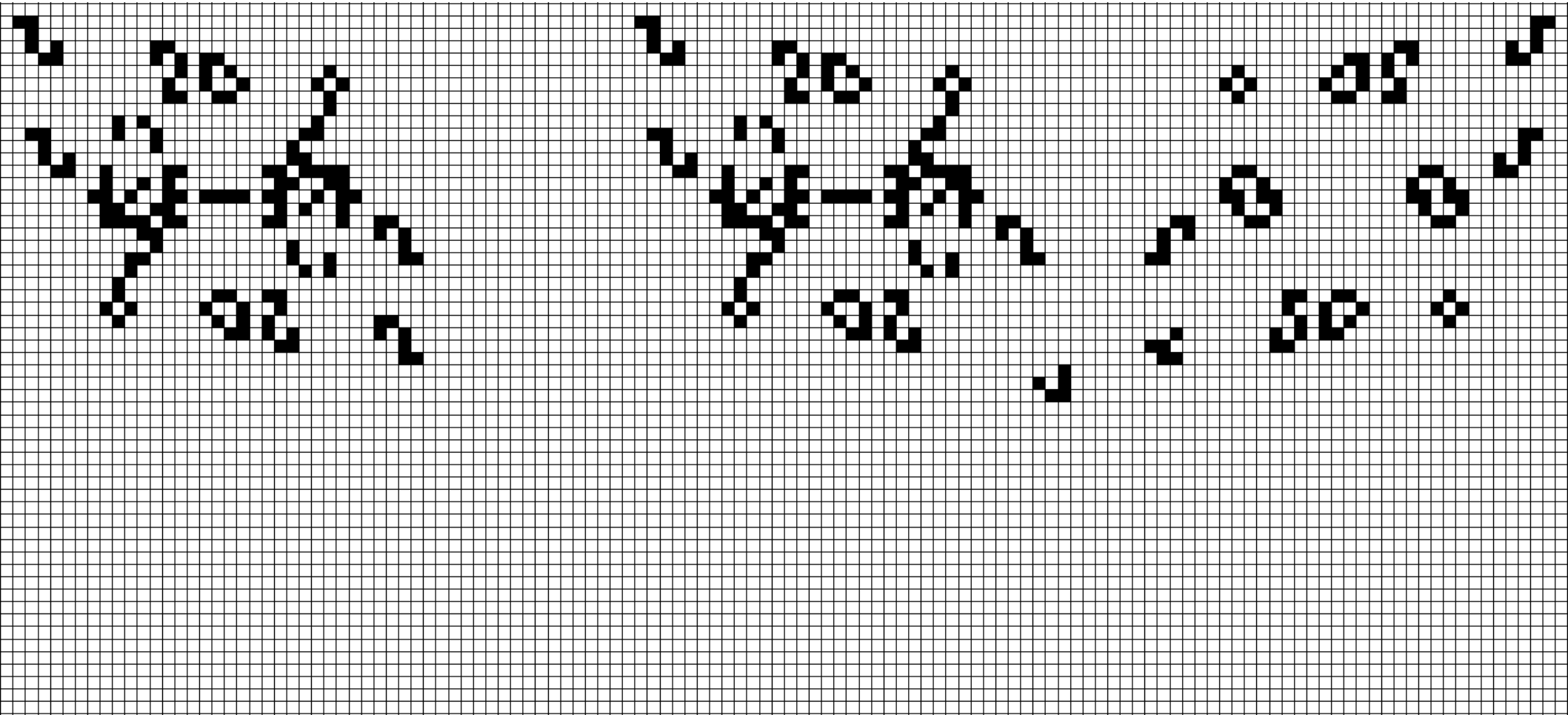
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

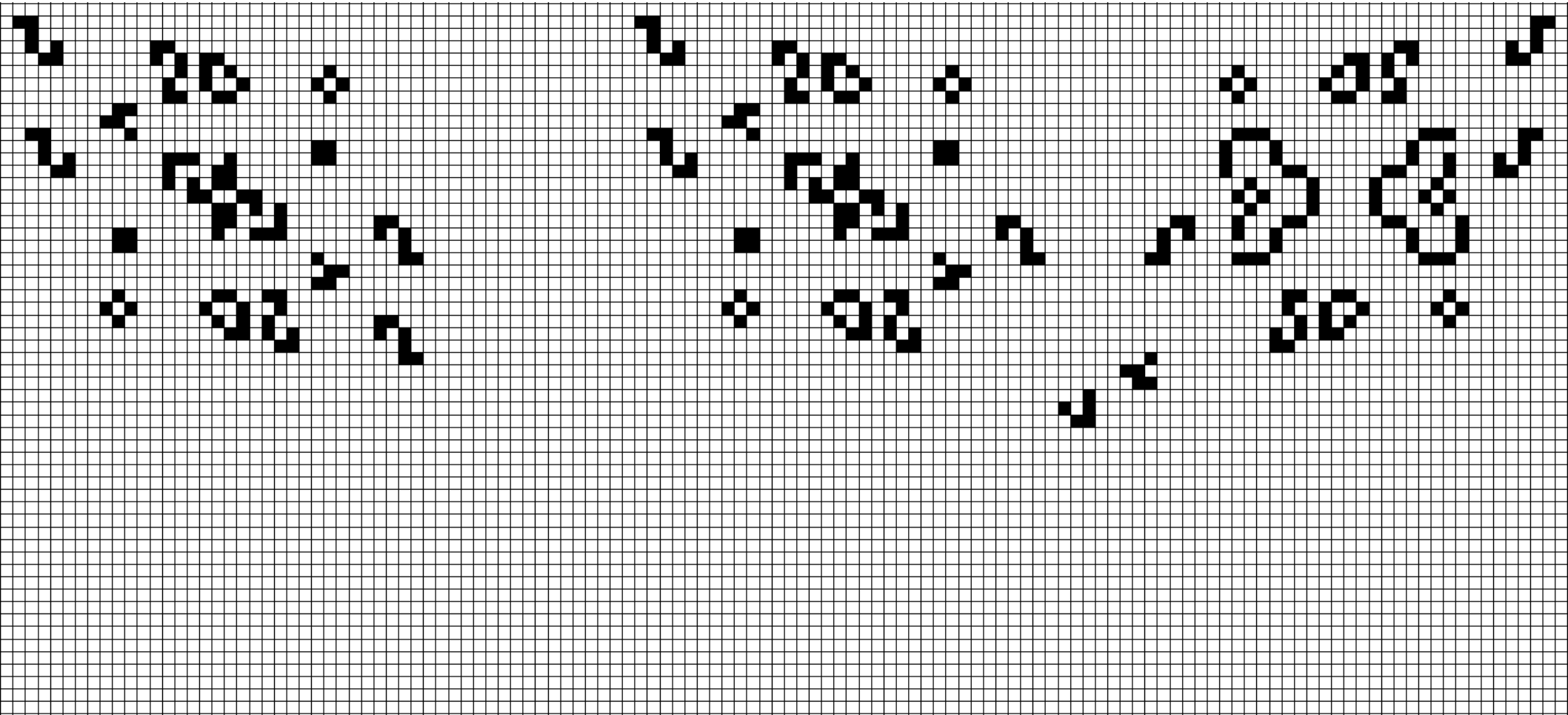
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

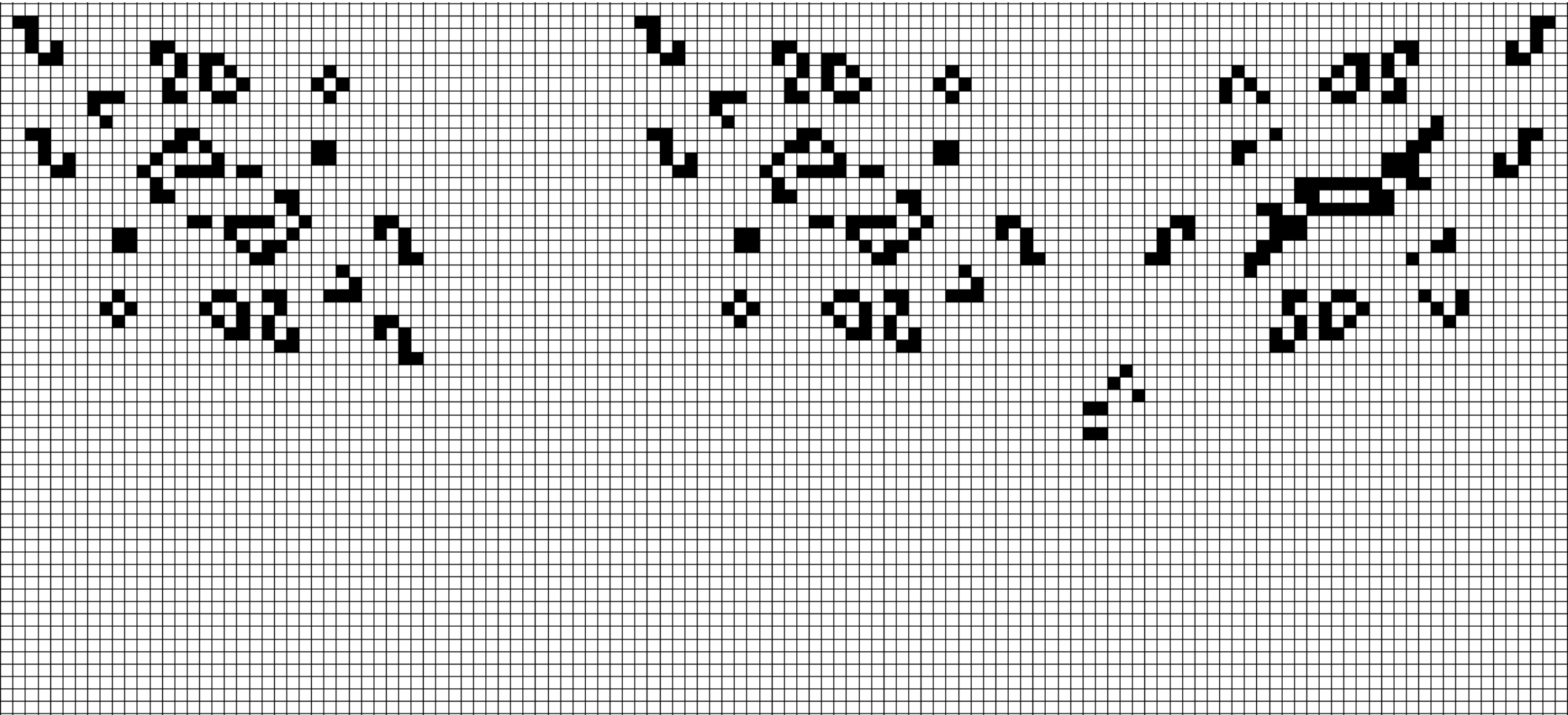
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

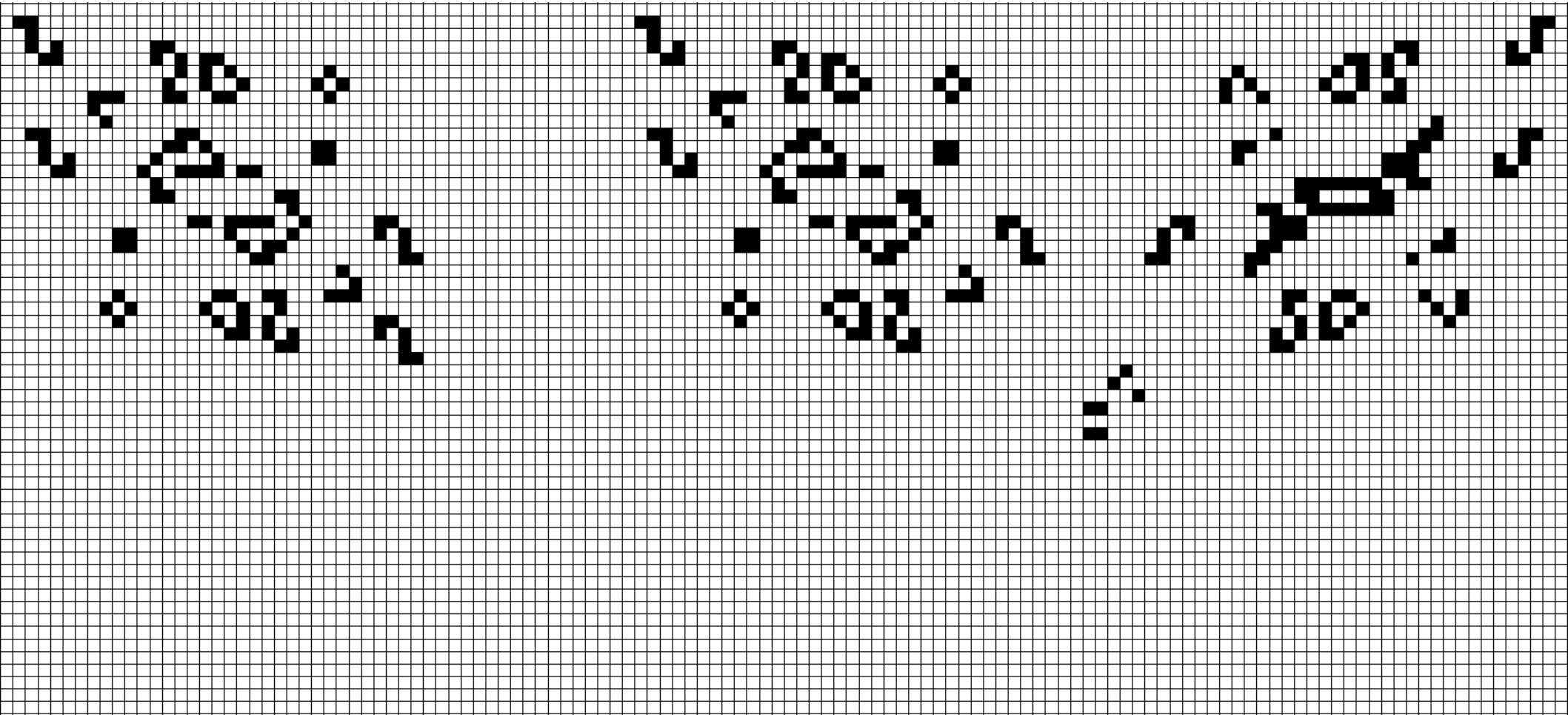
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

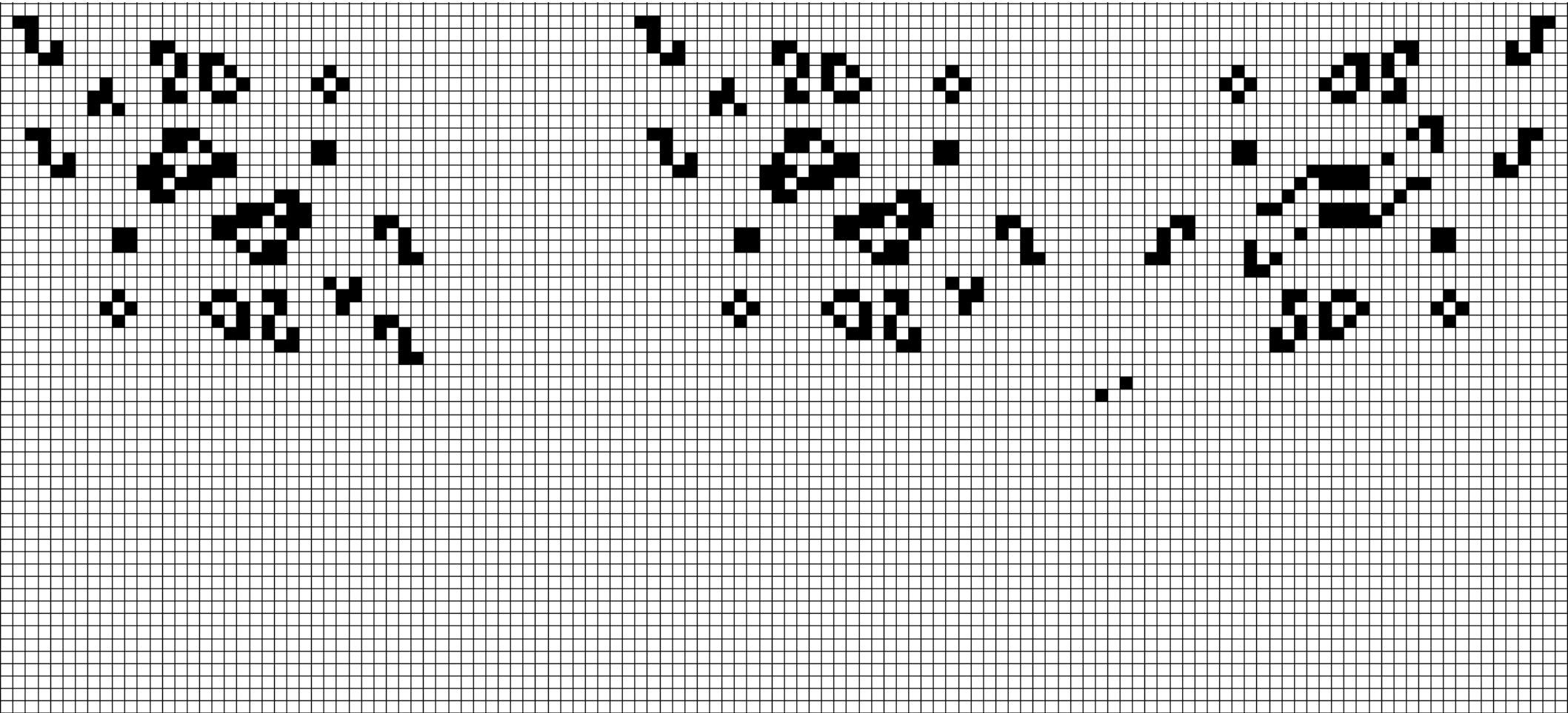
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

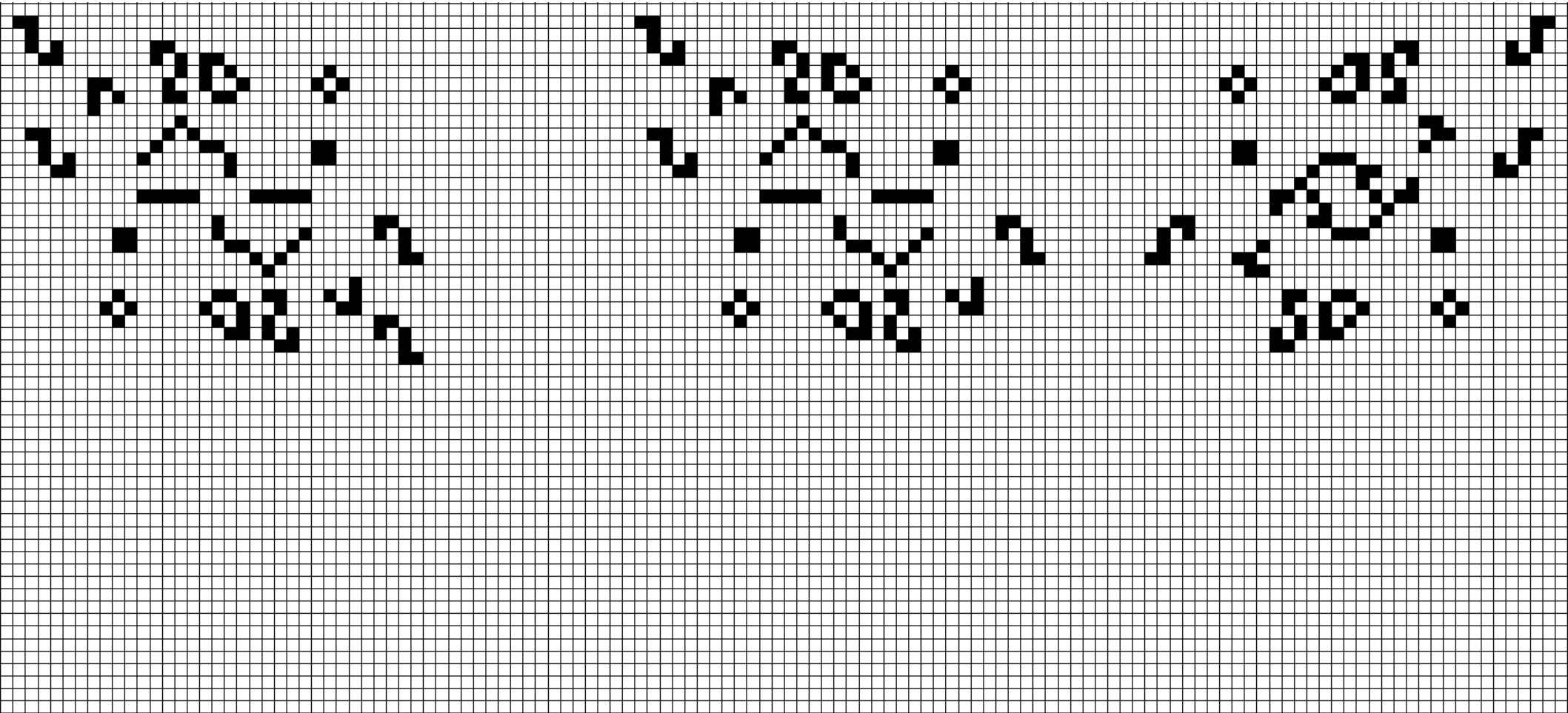
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

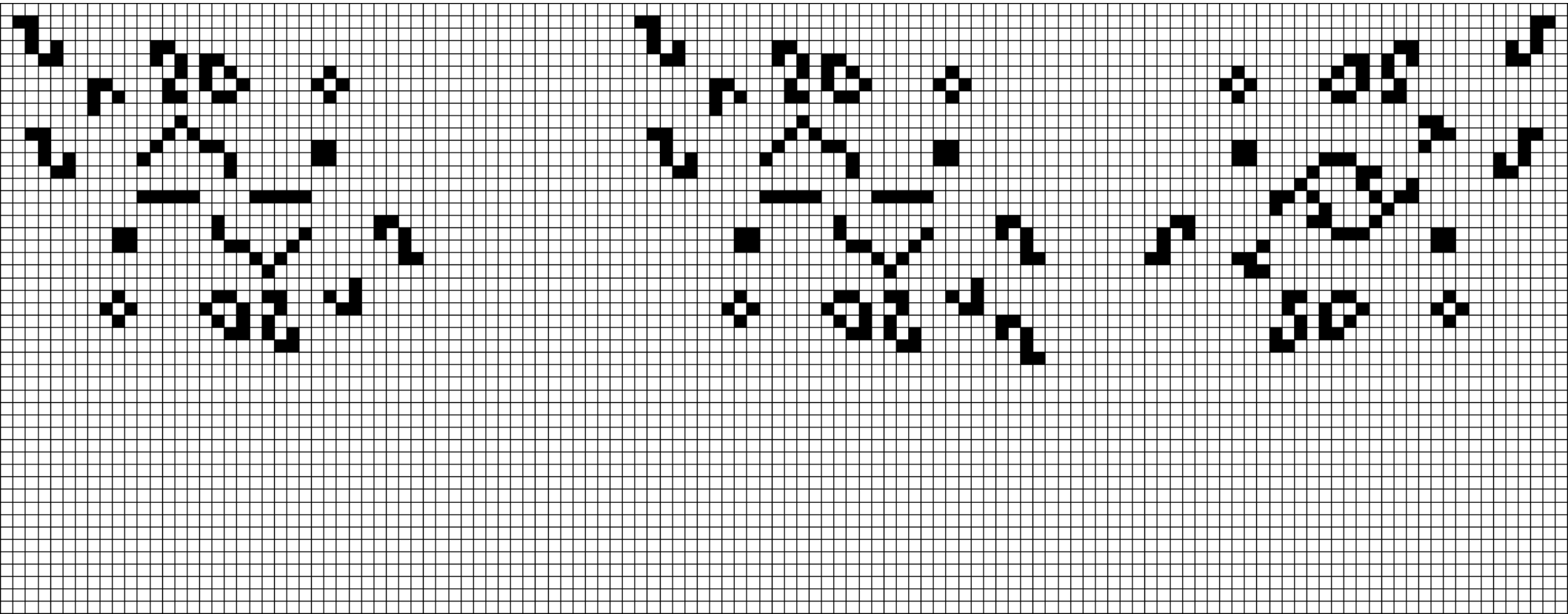
2. Portes logiques



AND : Entrée 0 1

II. Outils nécessaires et composition du filtre

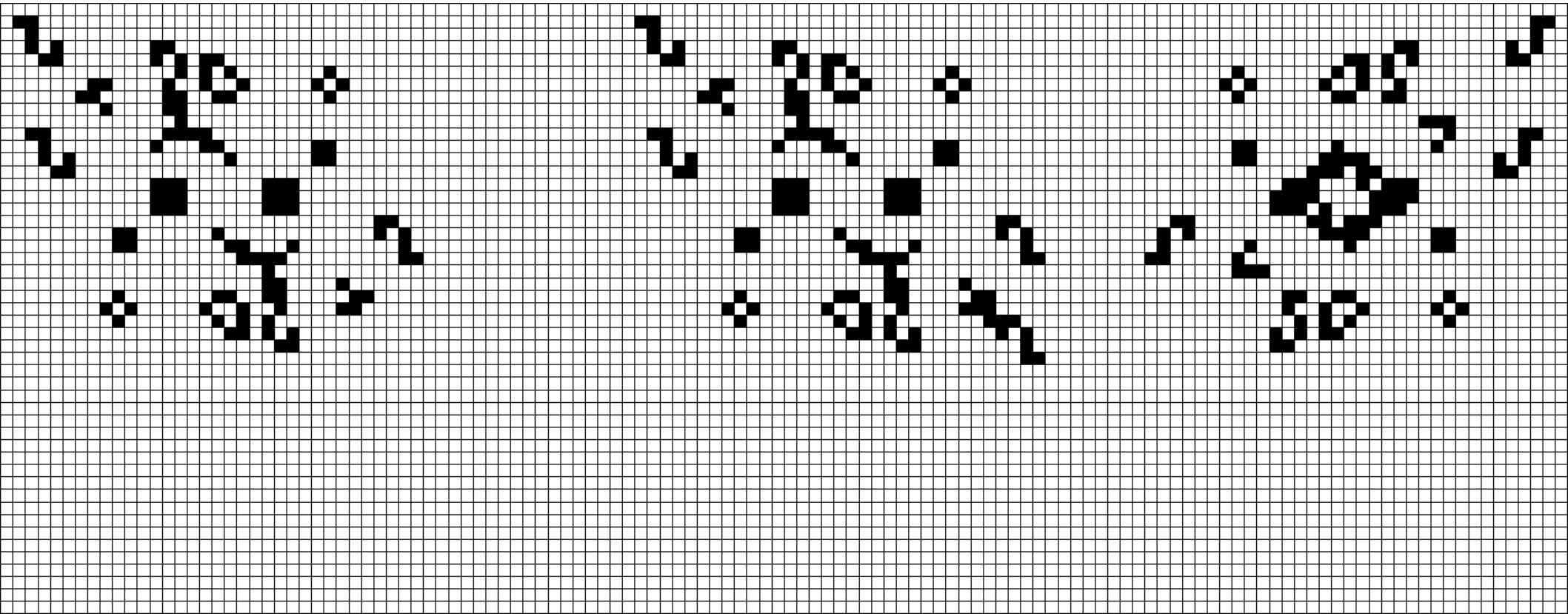
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

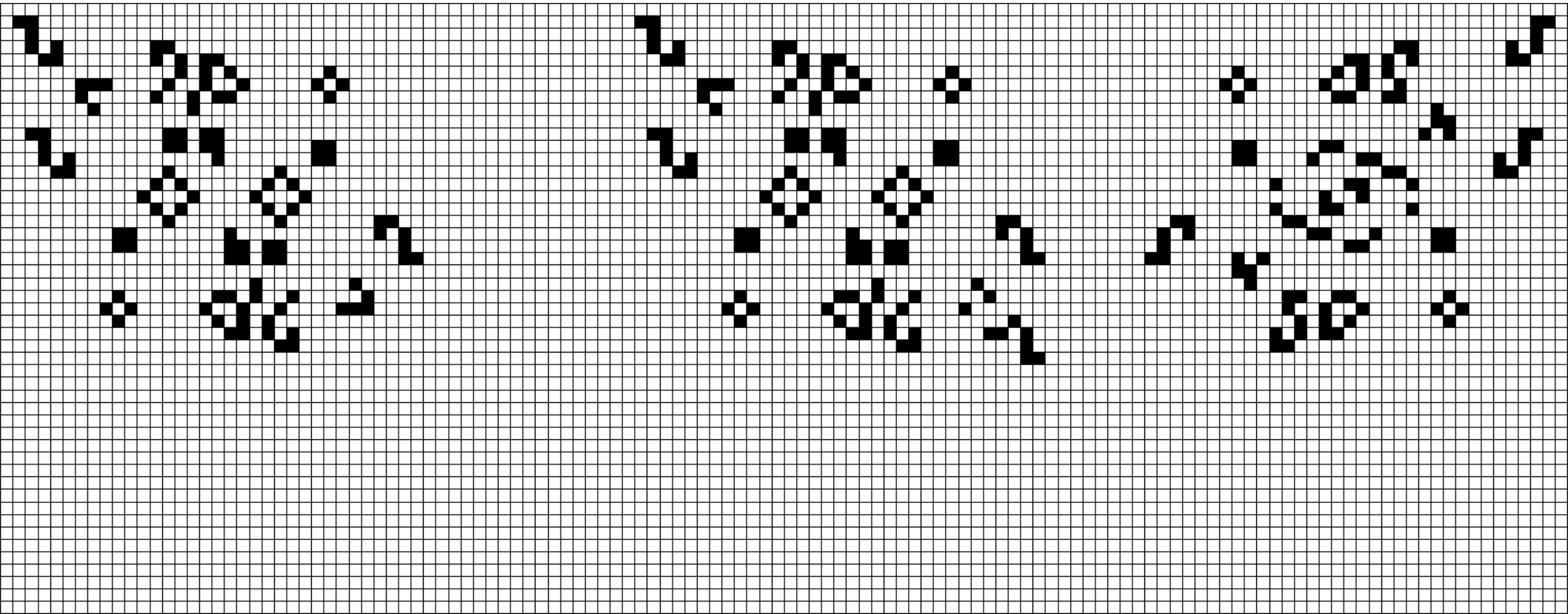
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

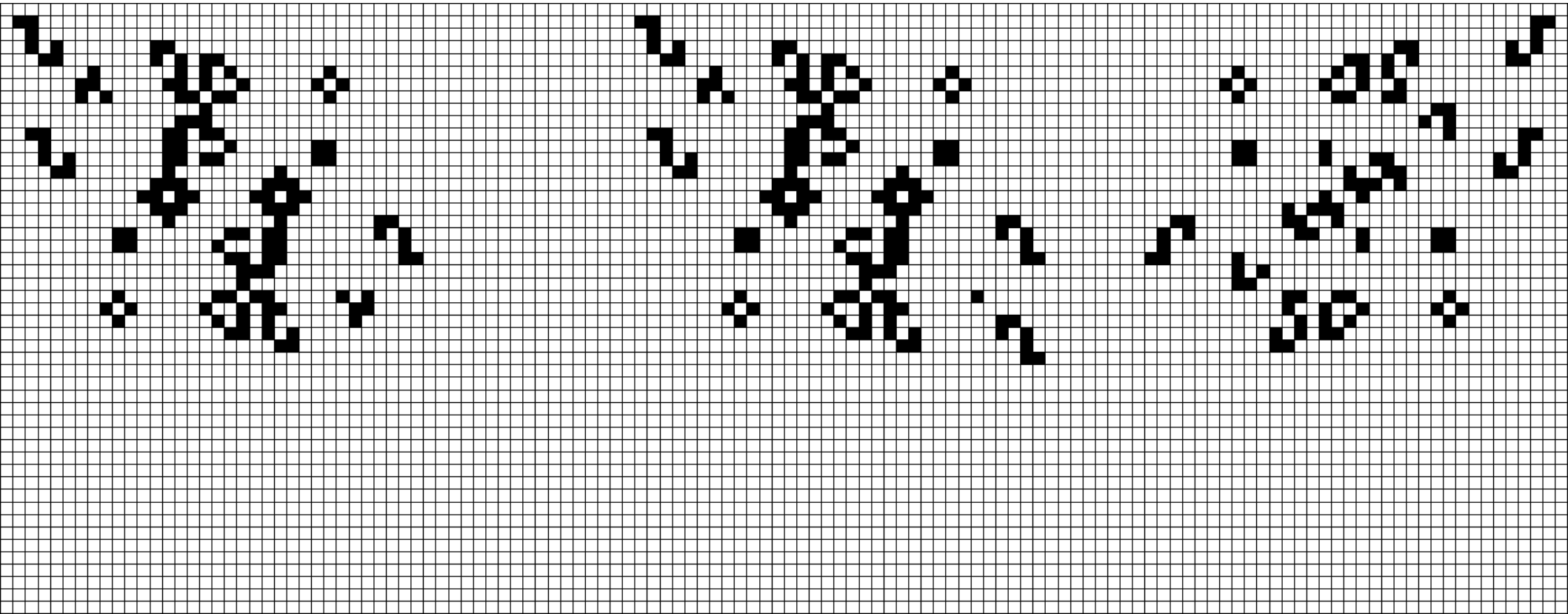
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

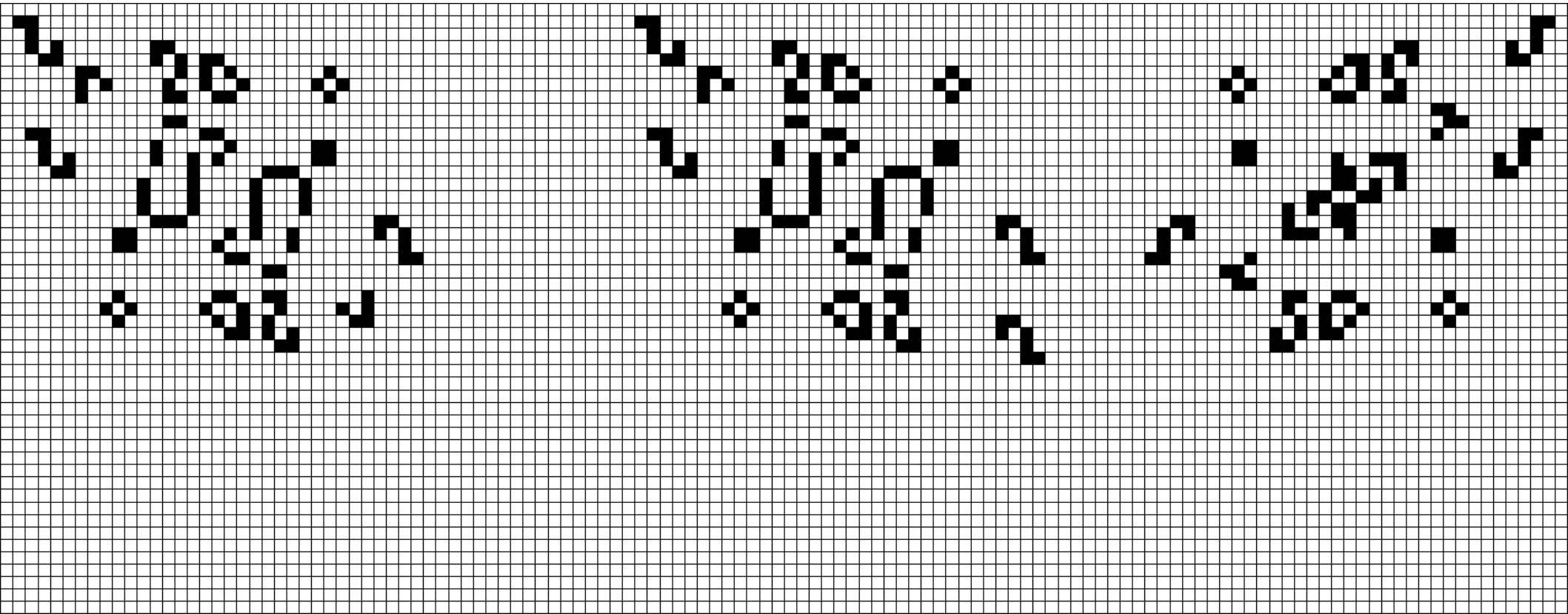
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

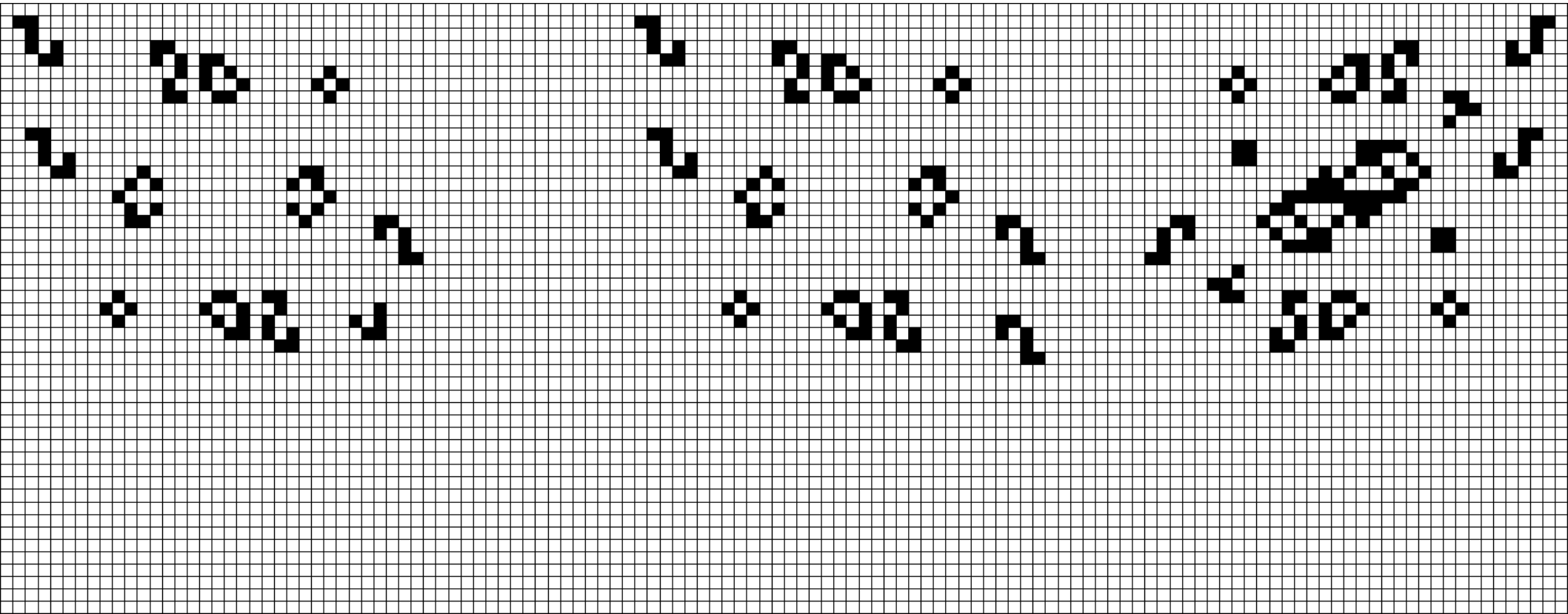
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

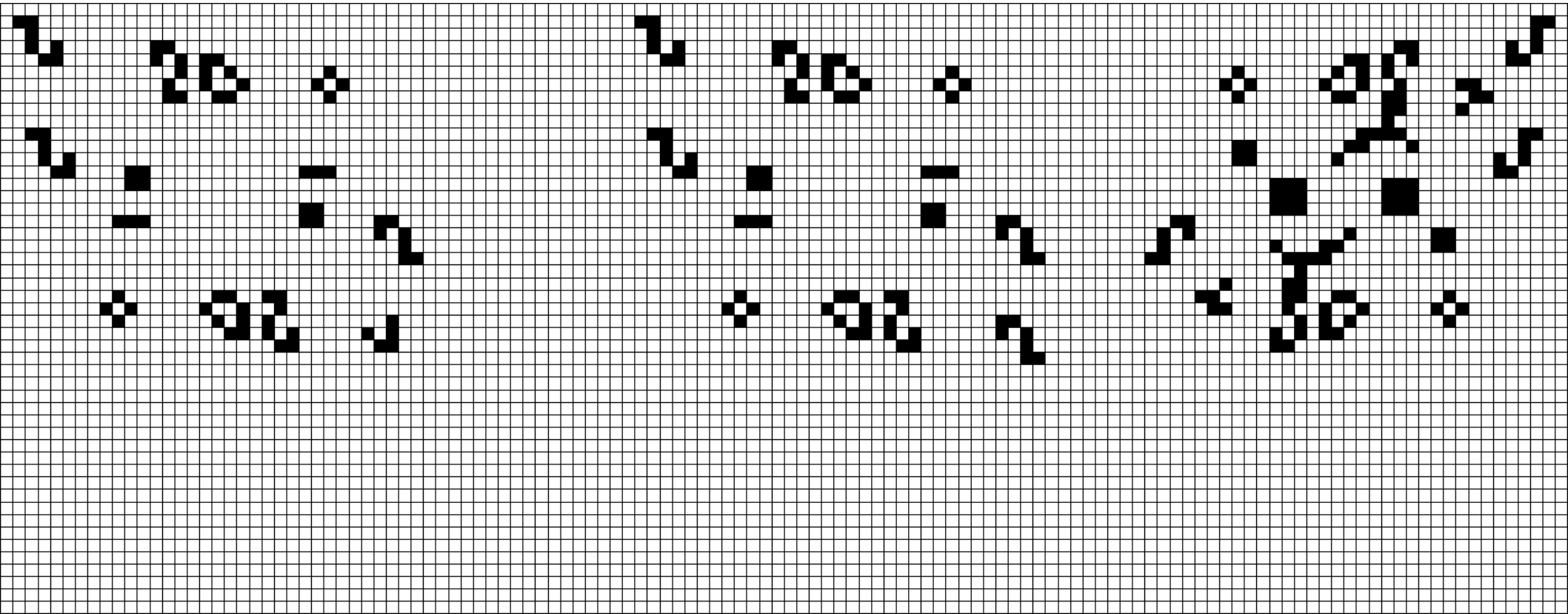
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

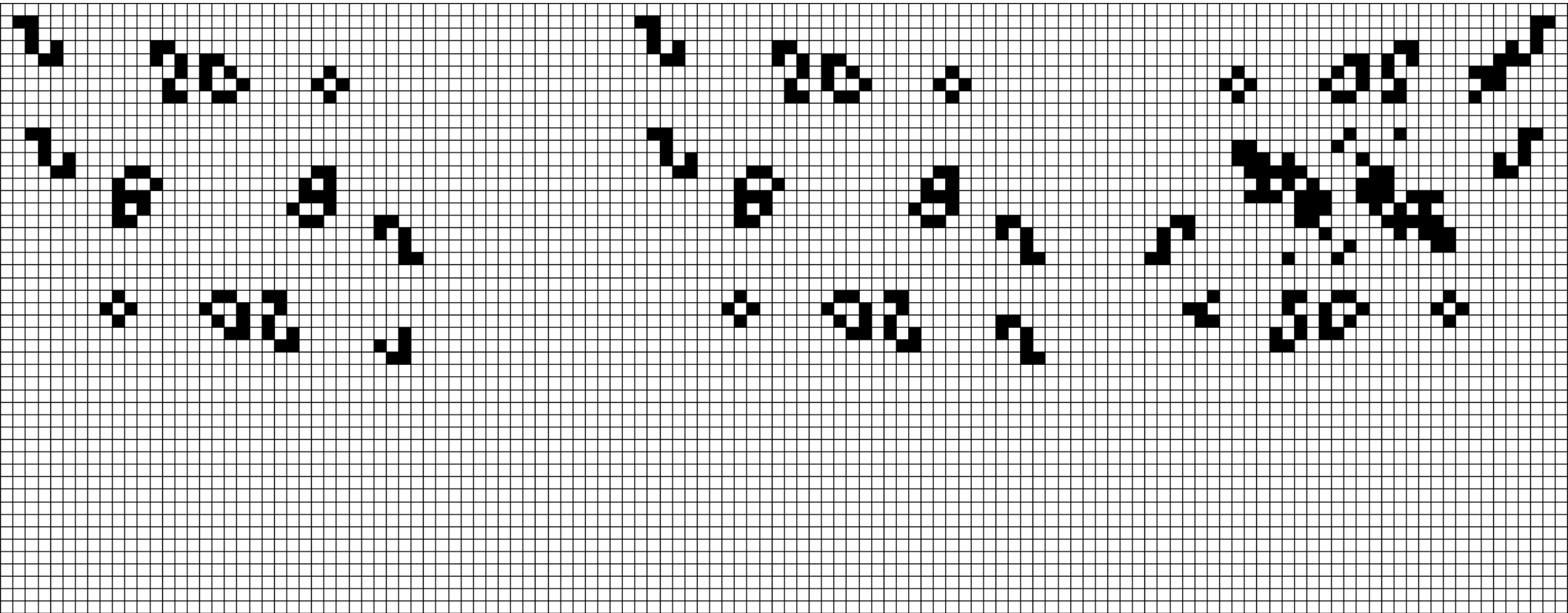
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

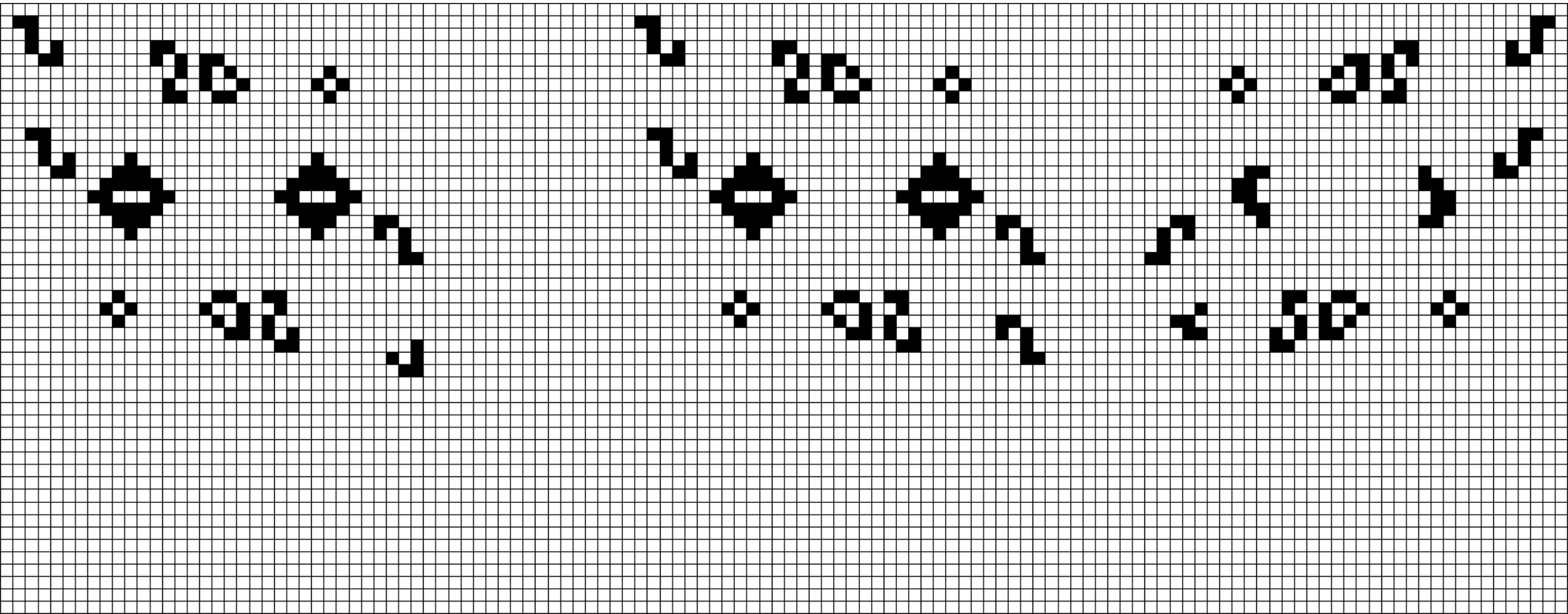
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

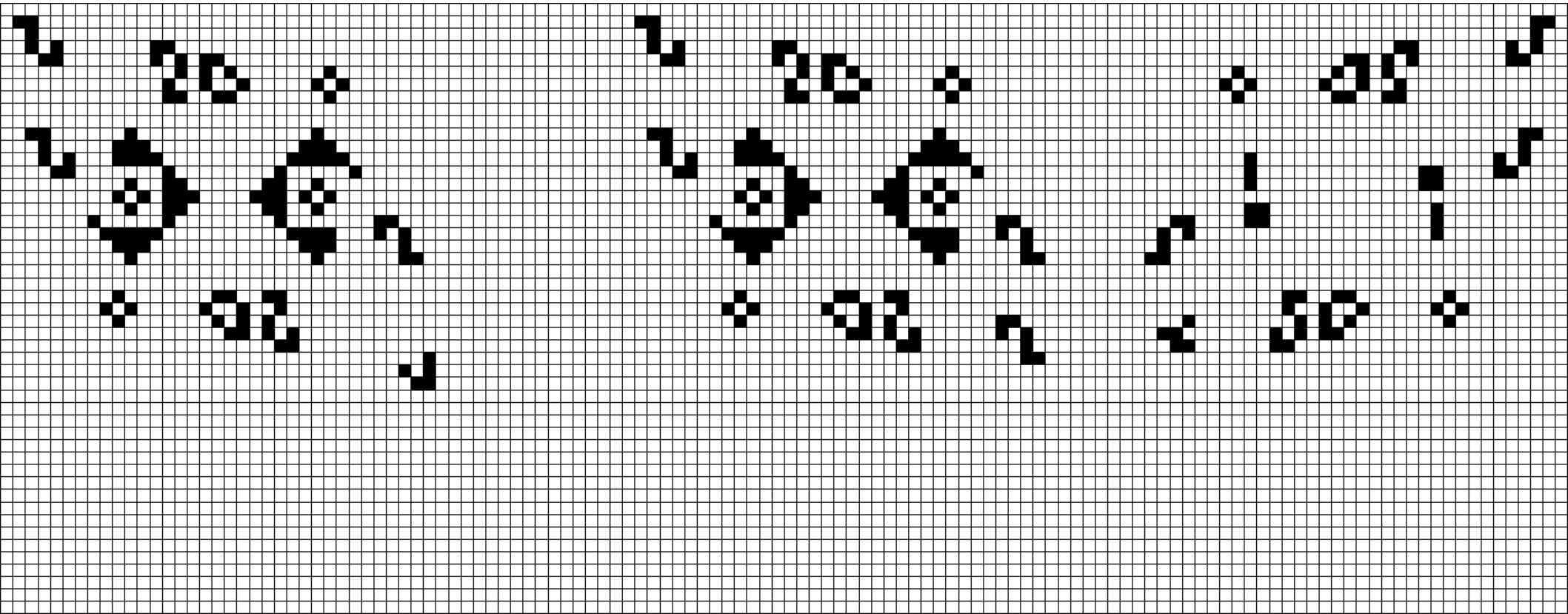
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

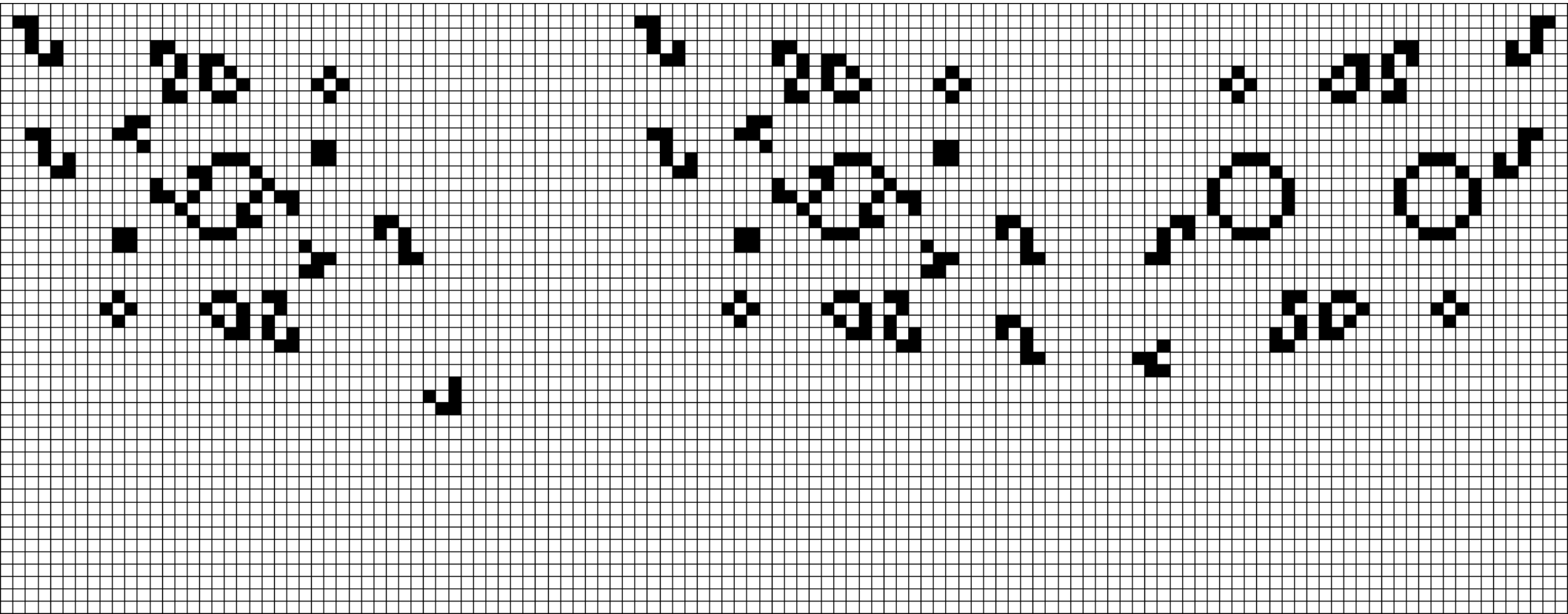
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

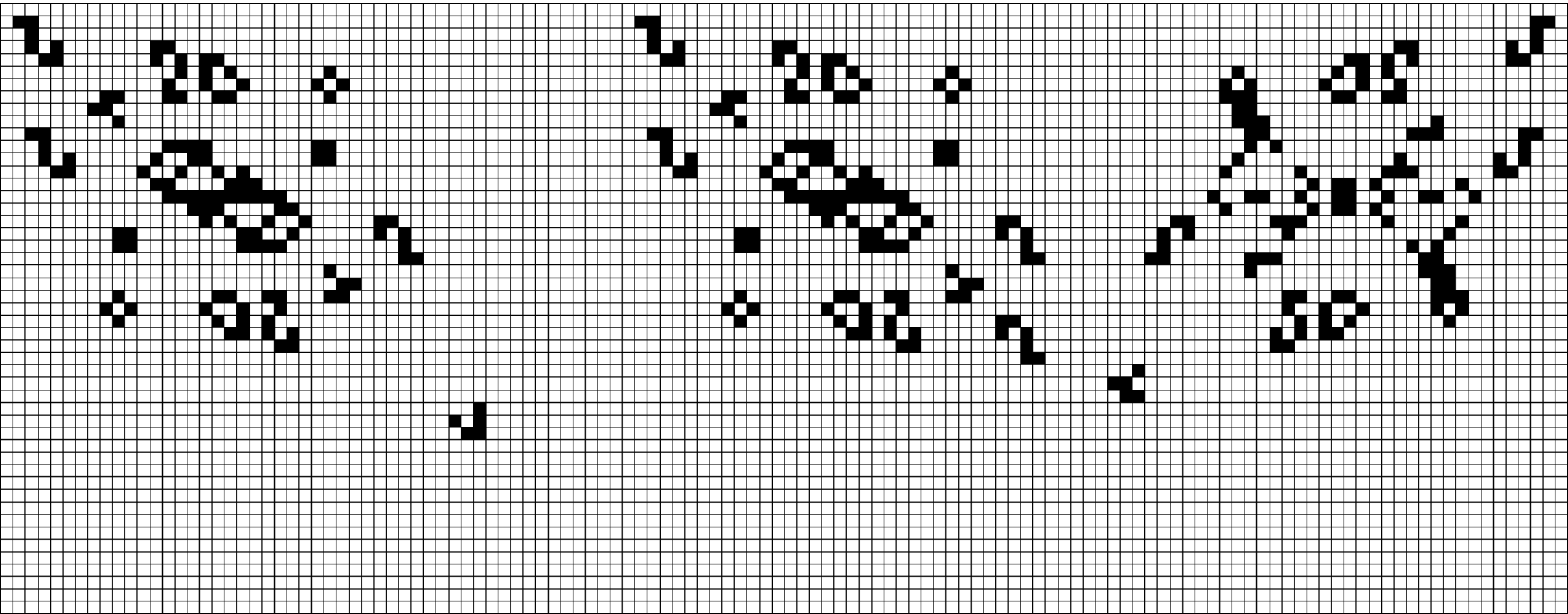
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

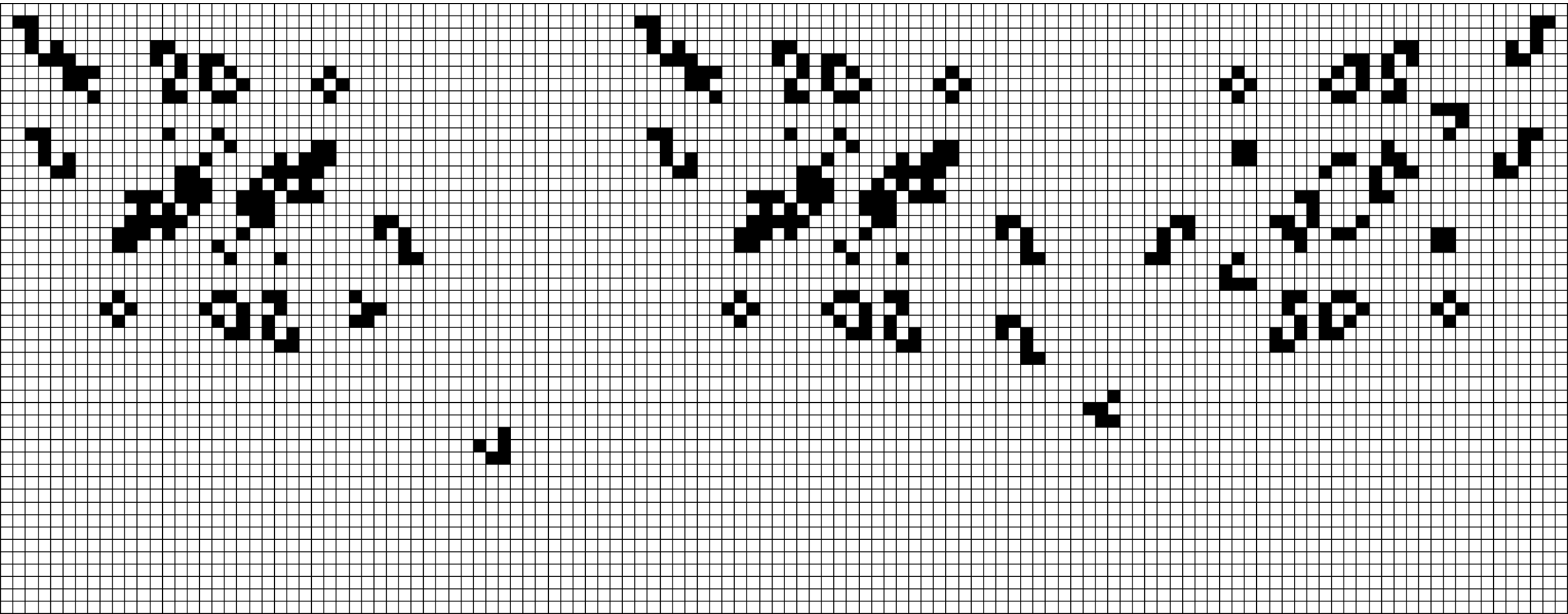
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

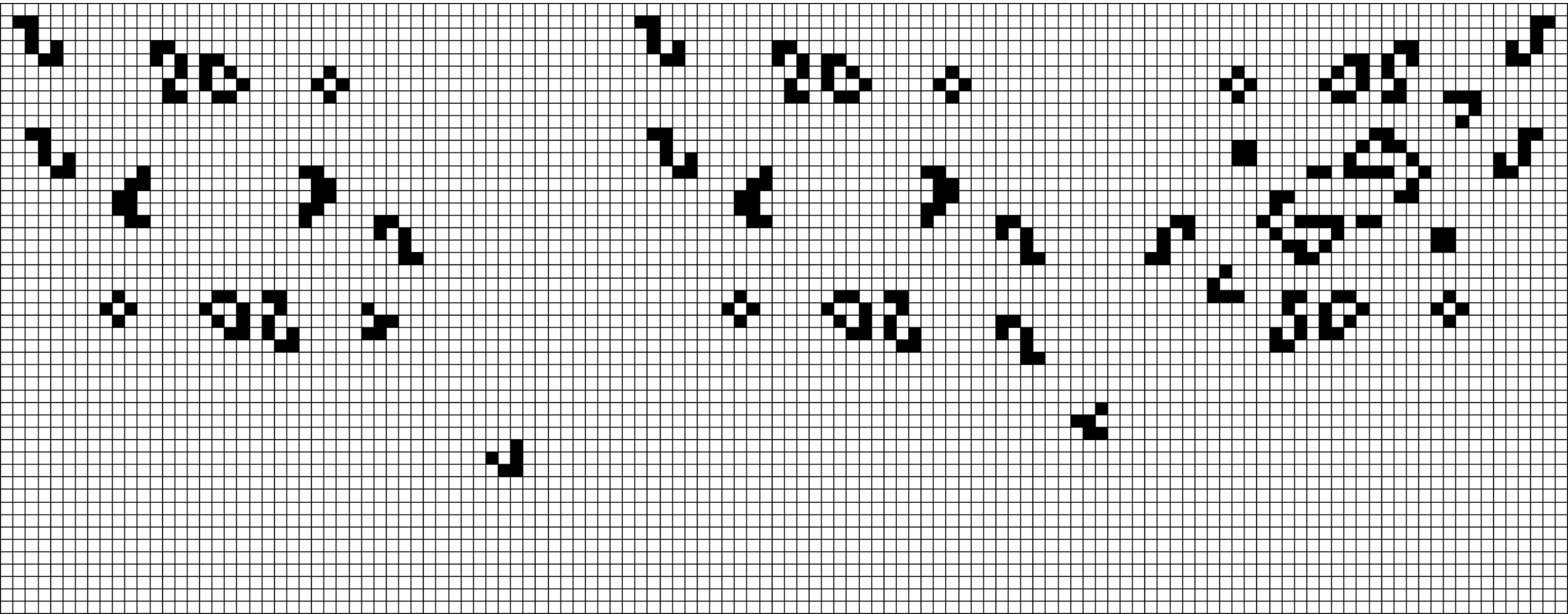
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

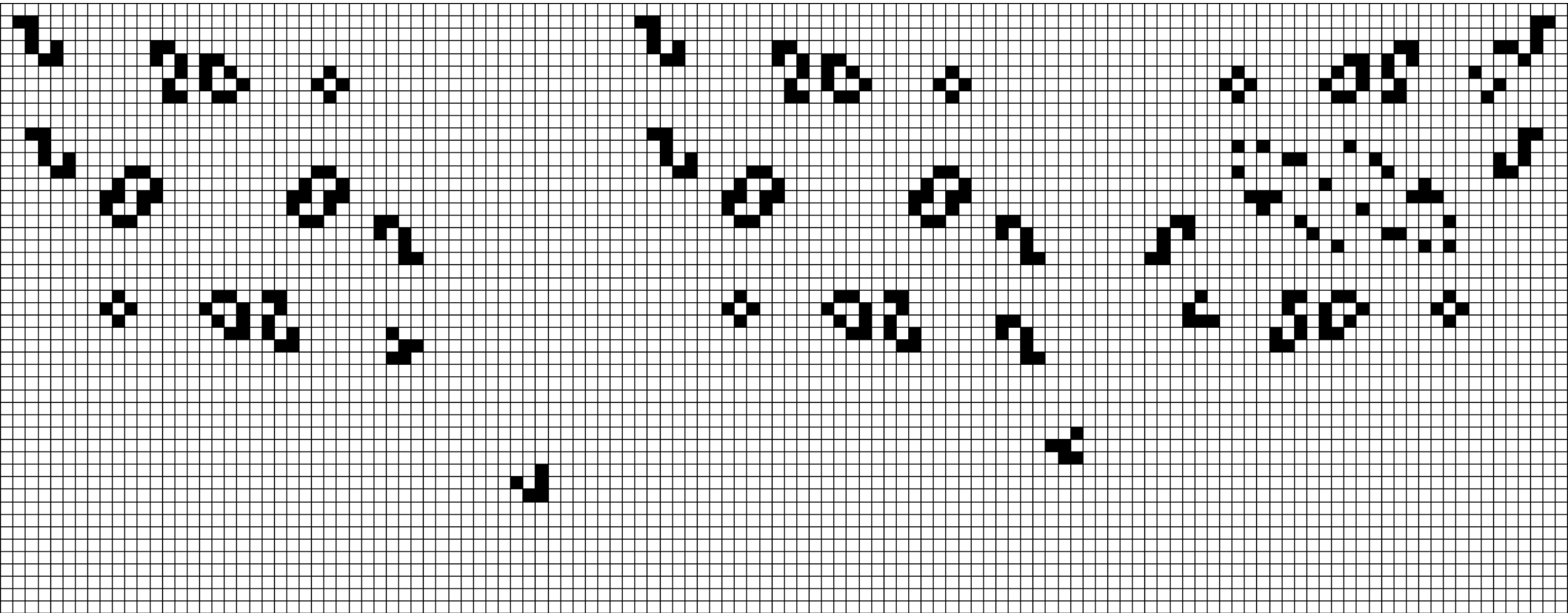
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

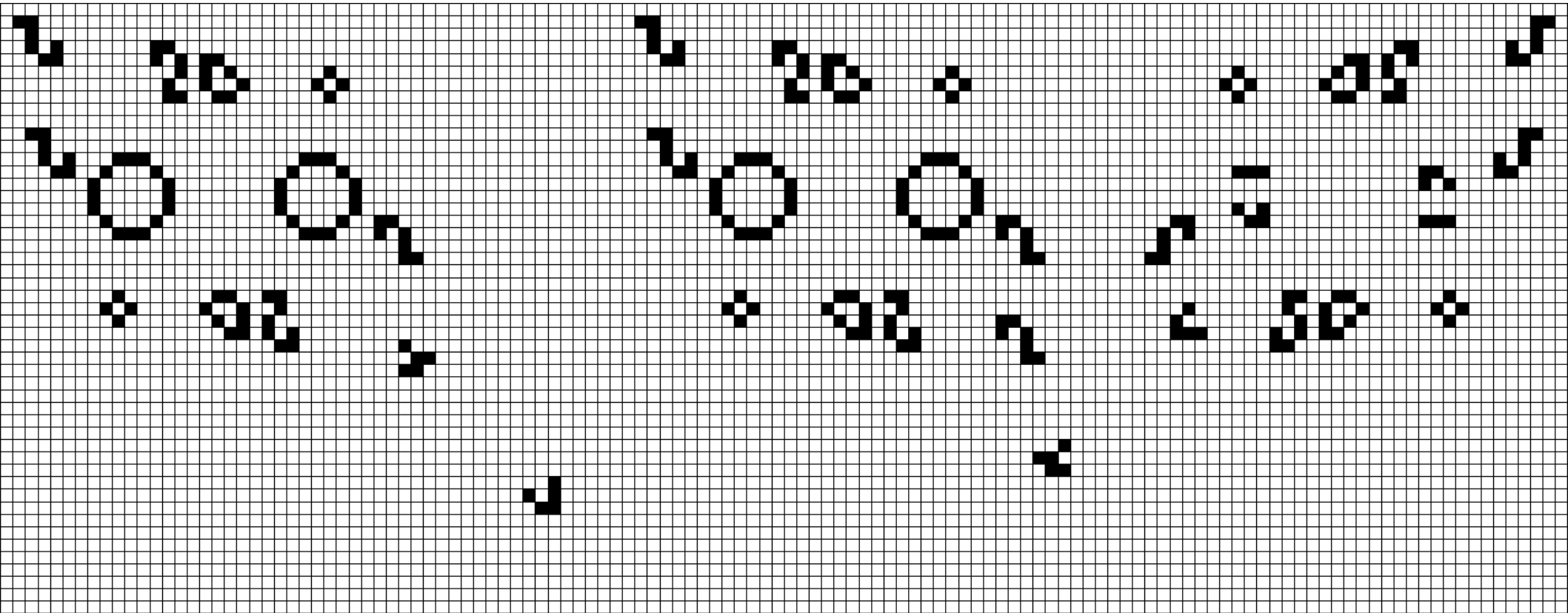
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

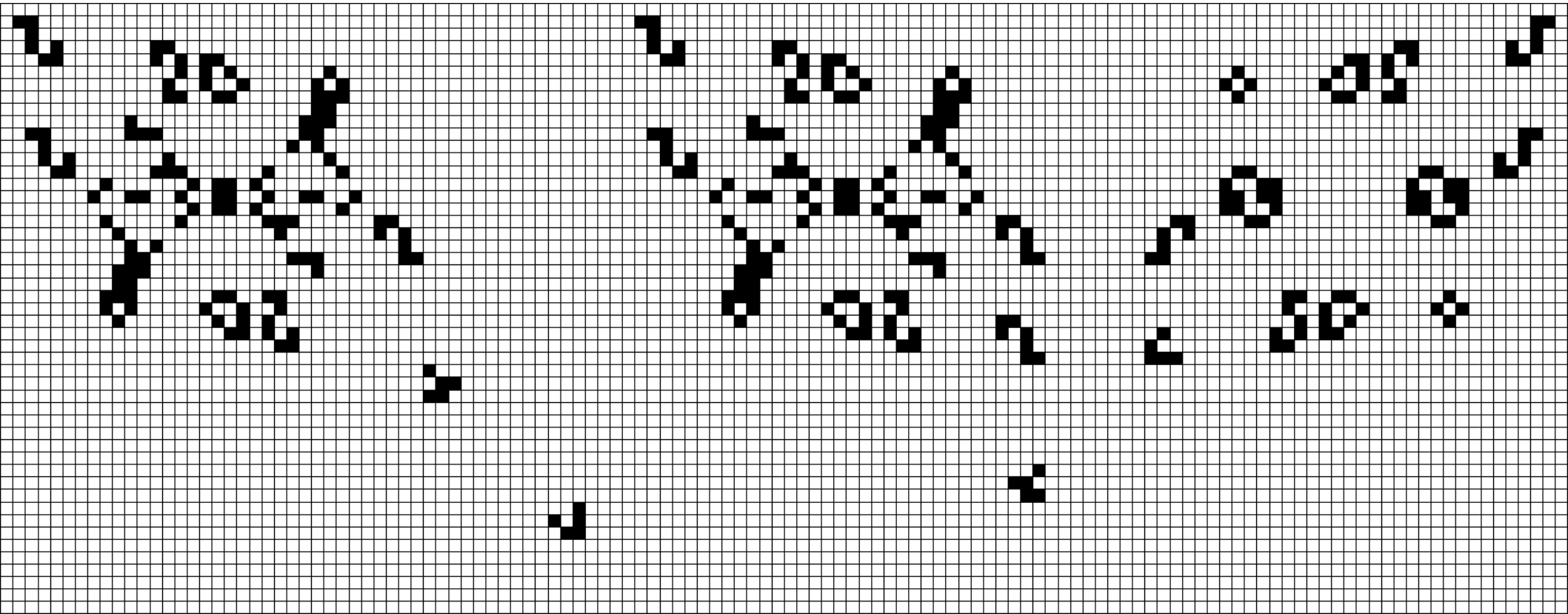
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

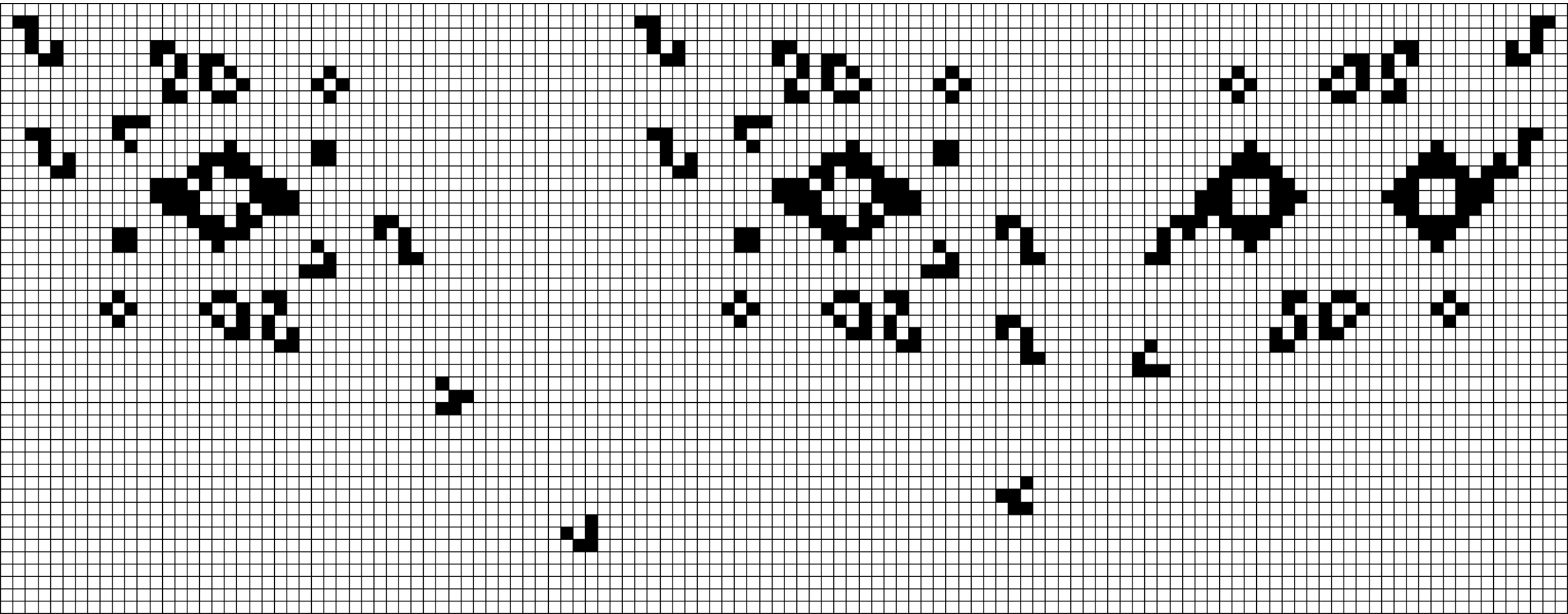
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

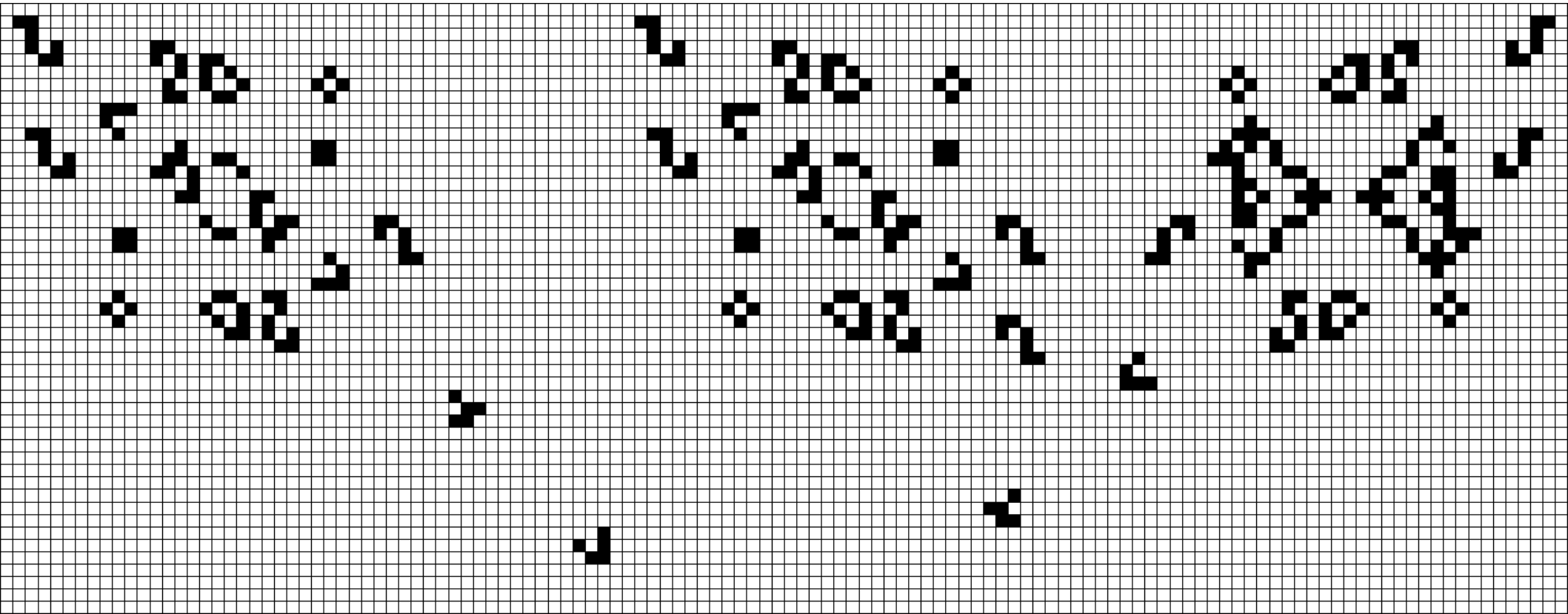
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

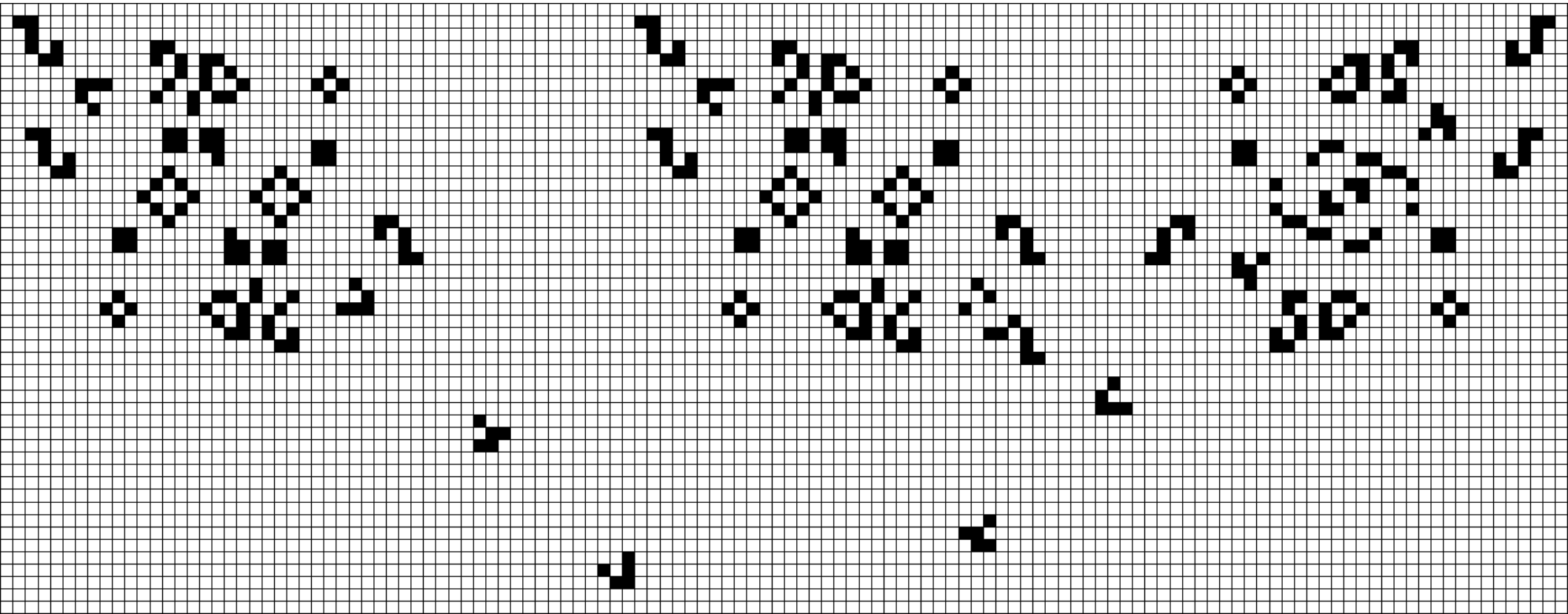
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

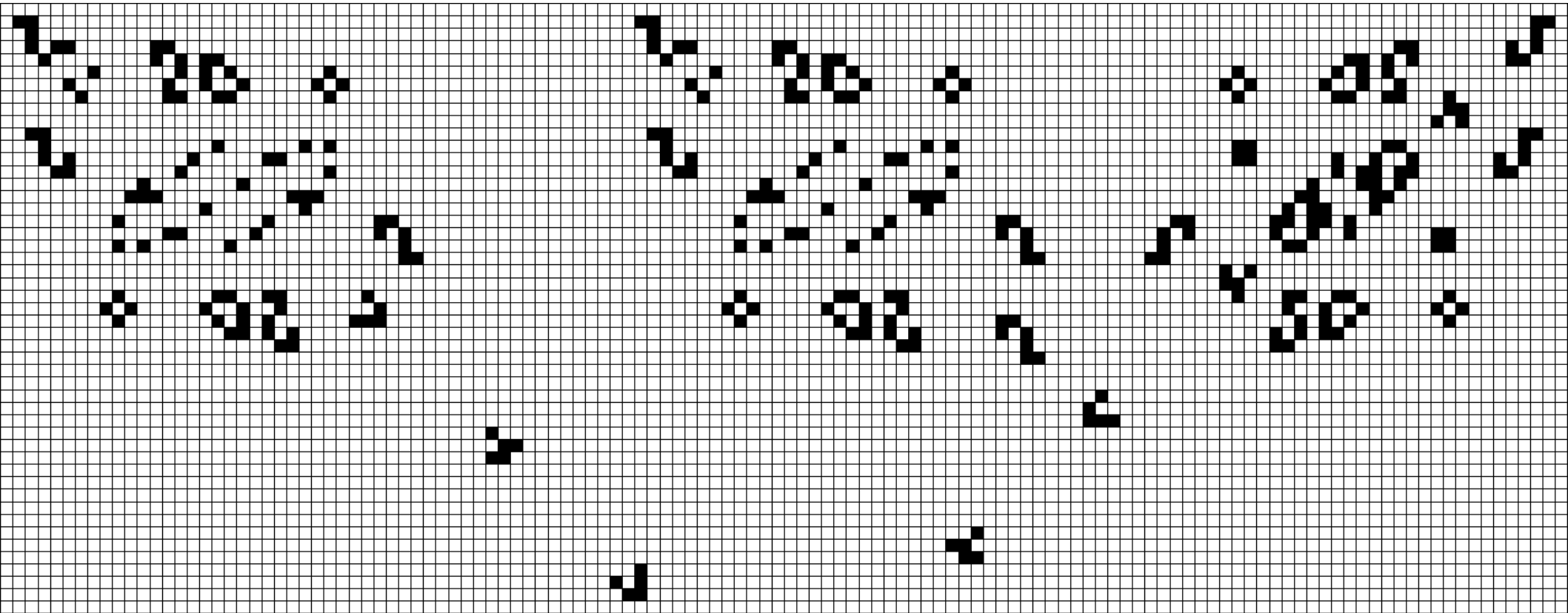
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

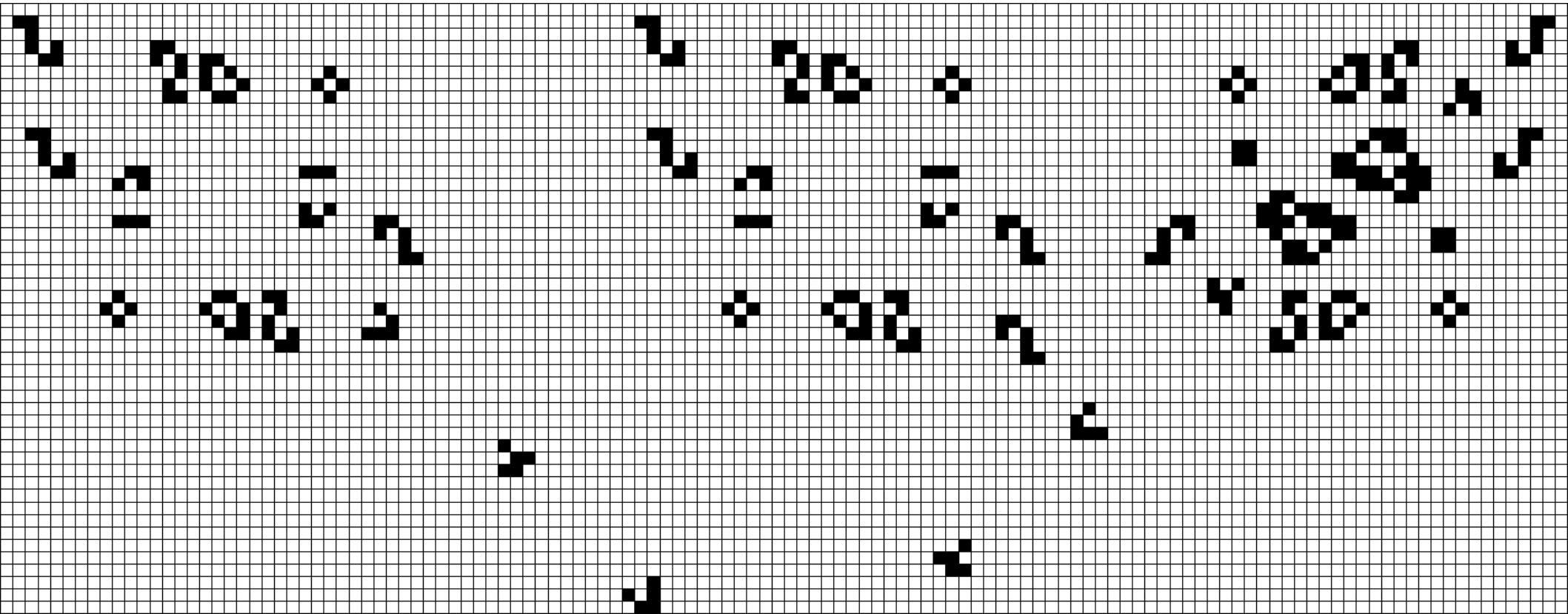
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

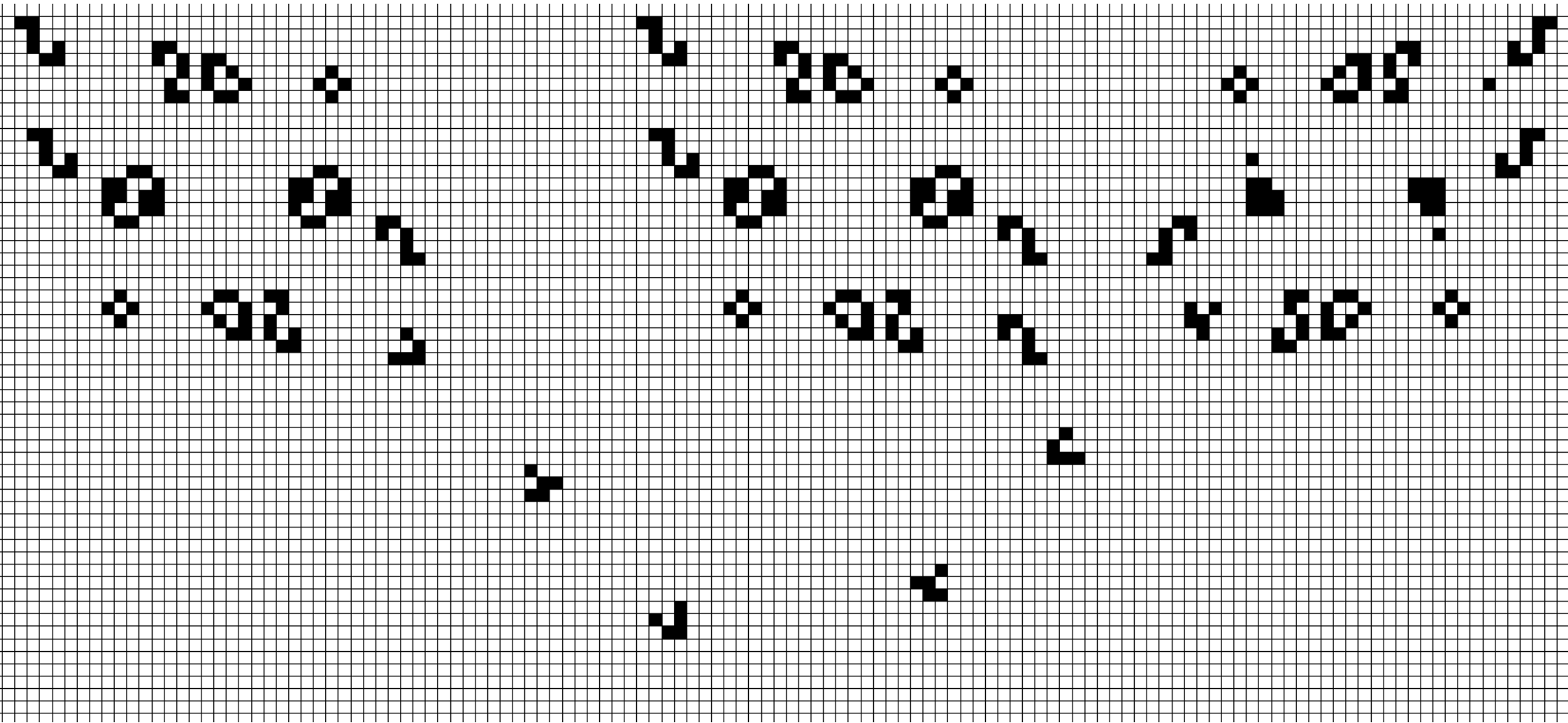
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

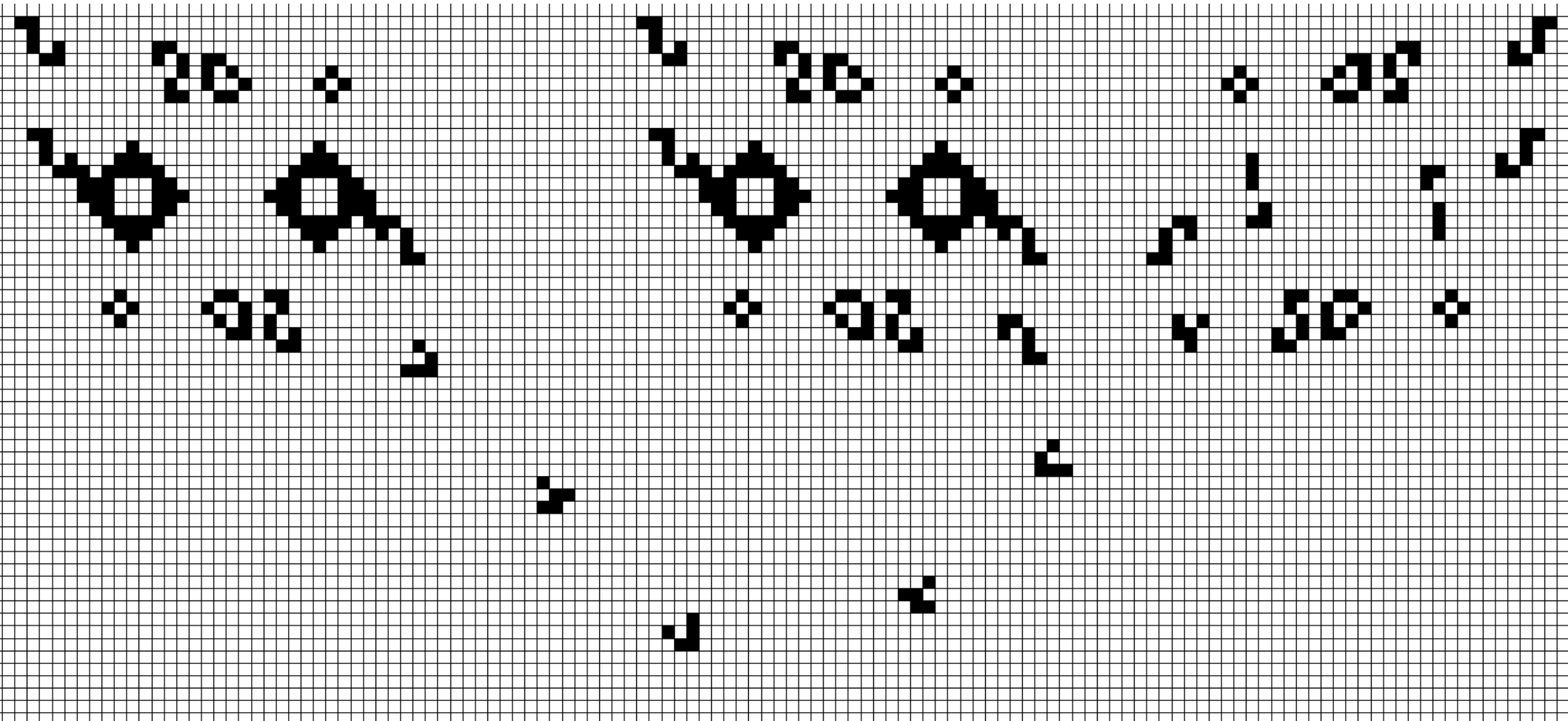
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

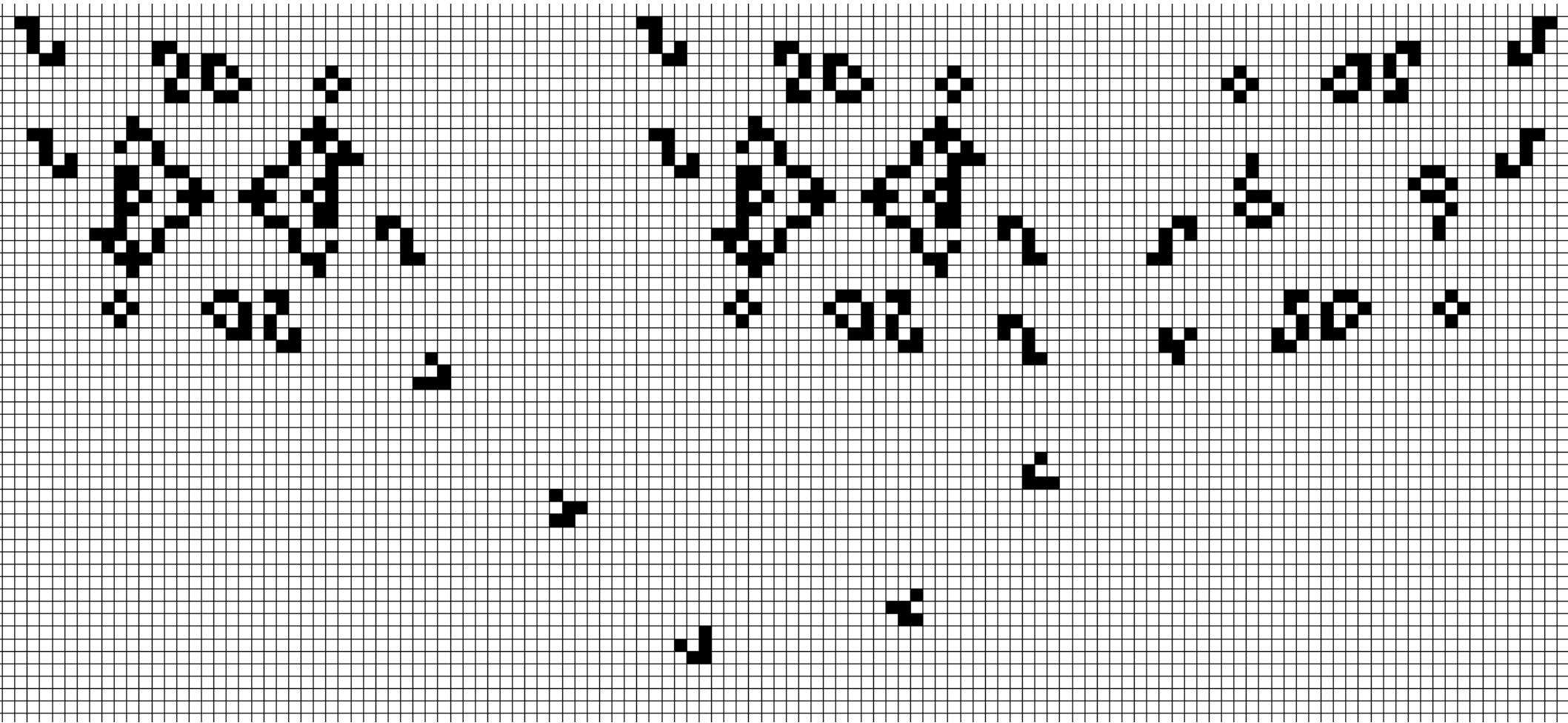
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

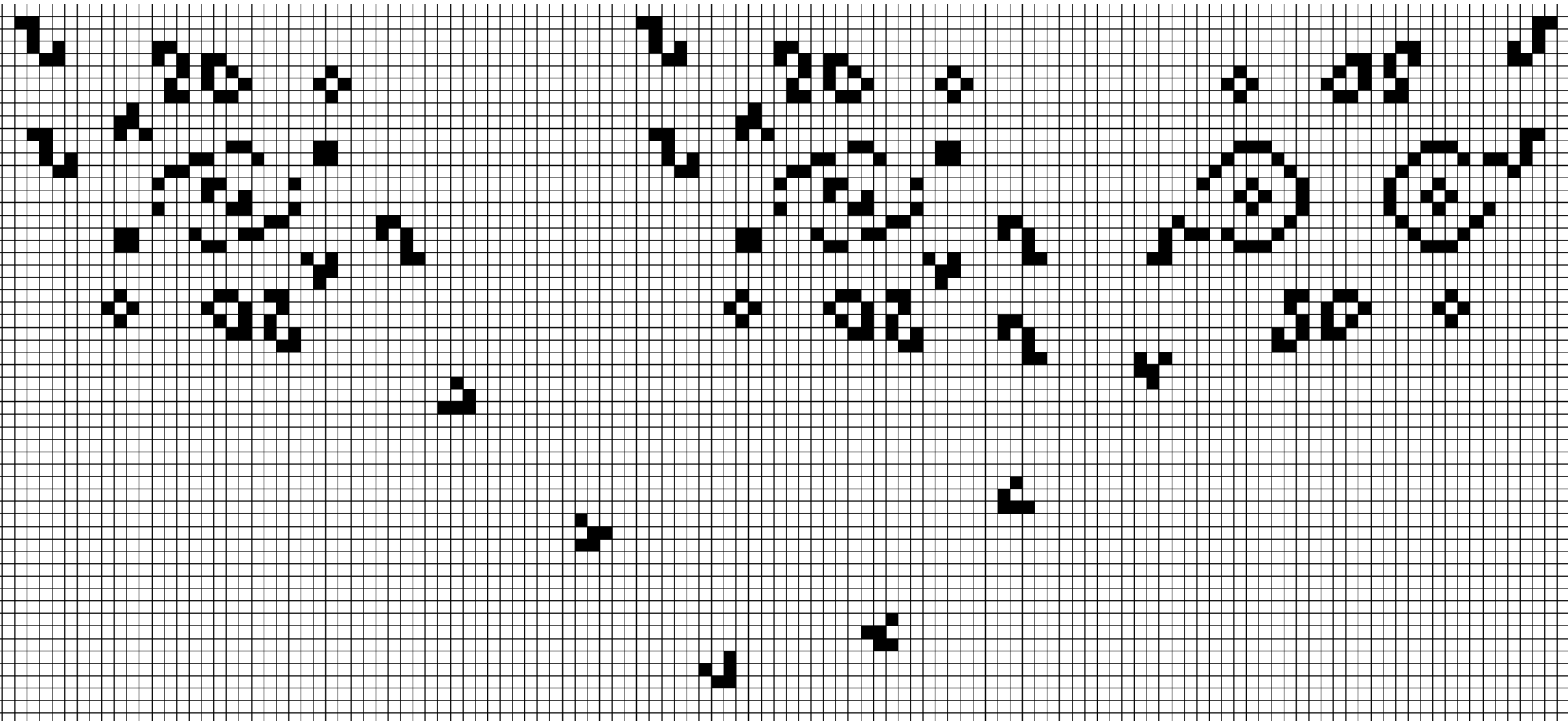
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

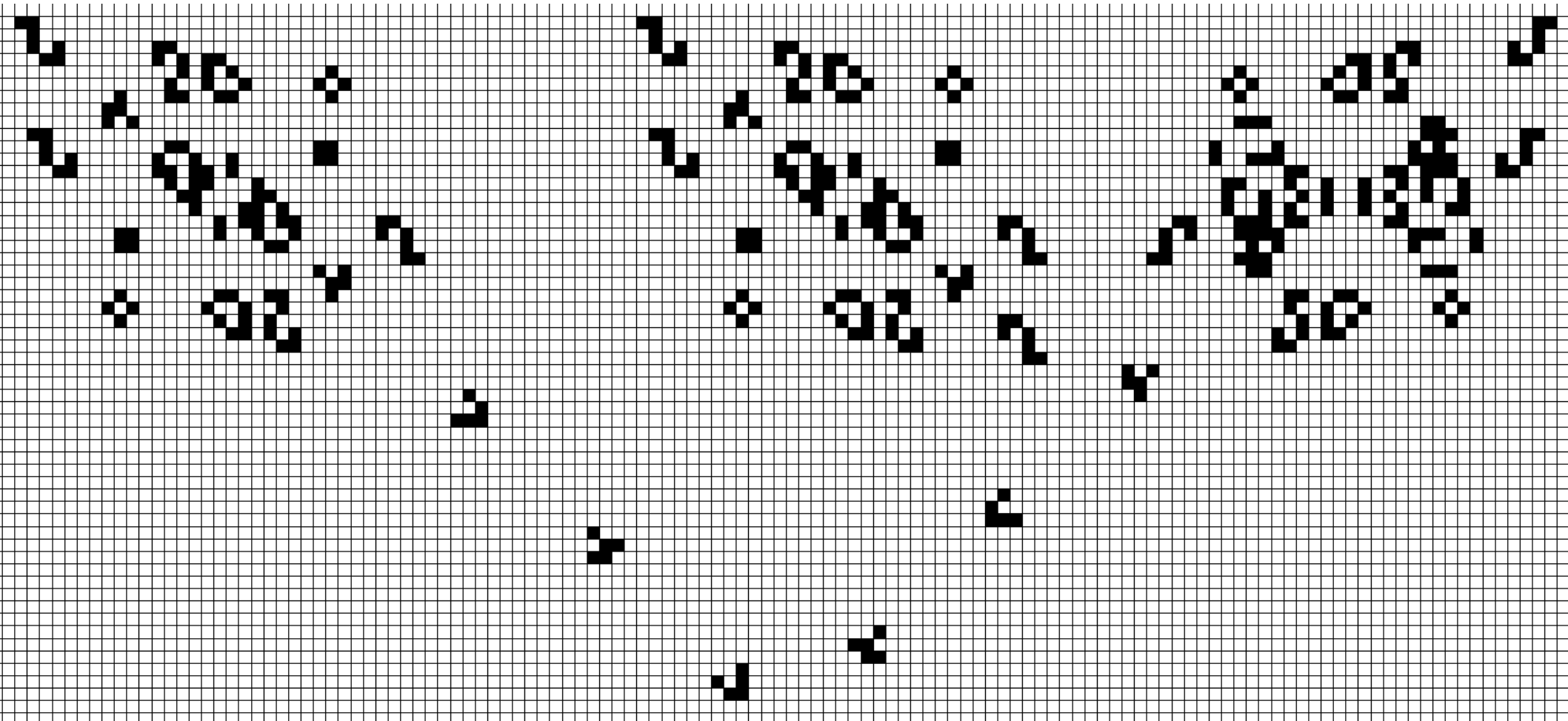
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

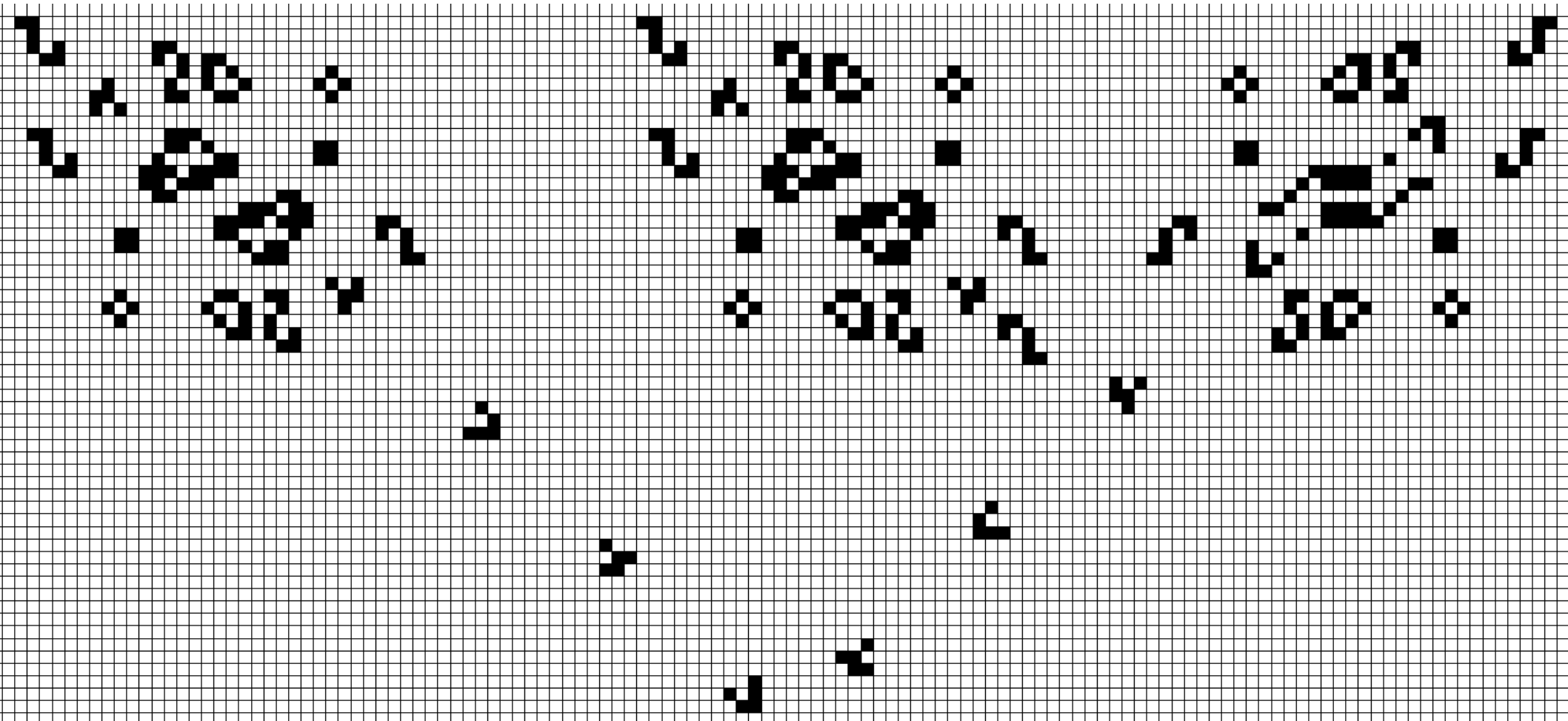
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

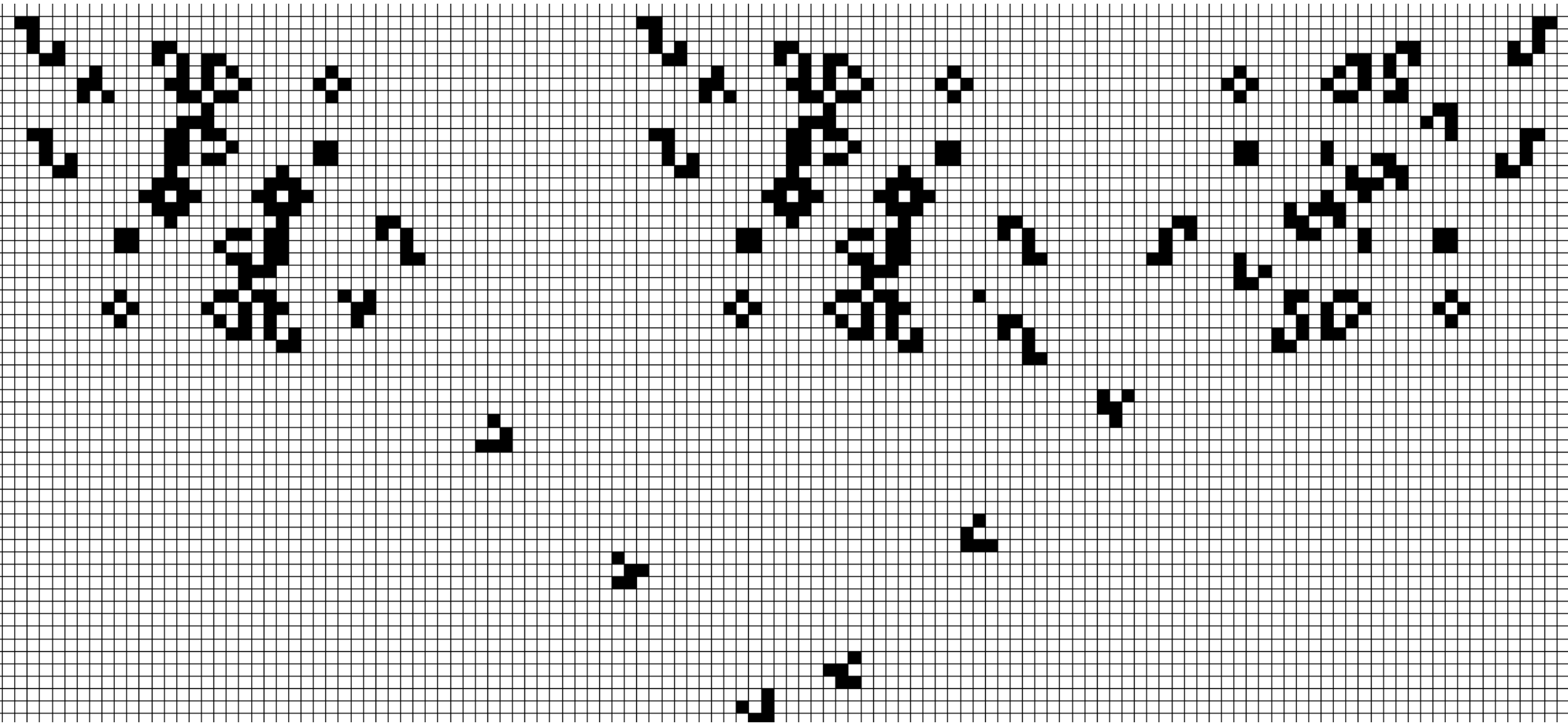
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

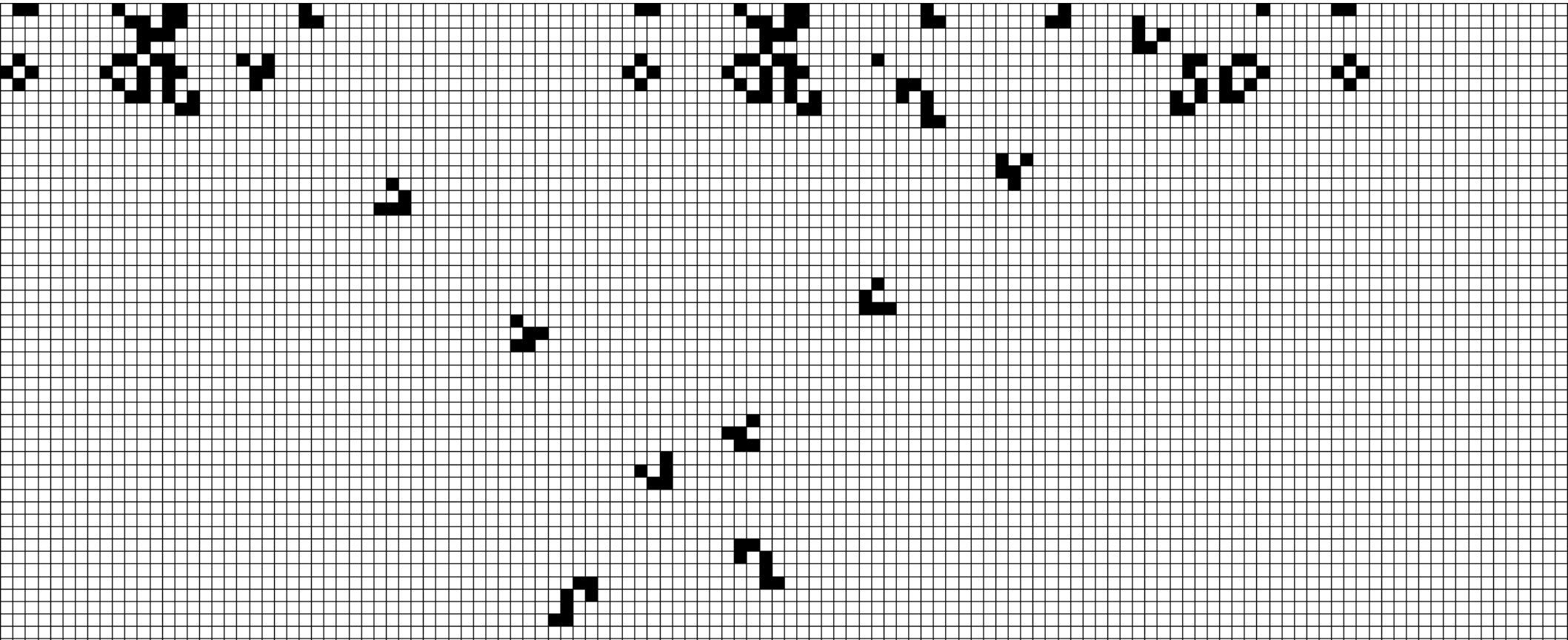
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

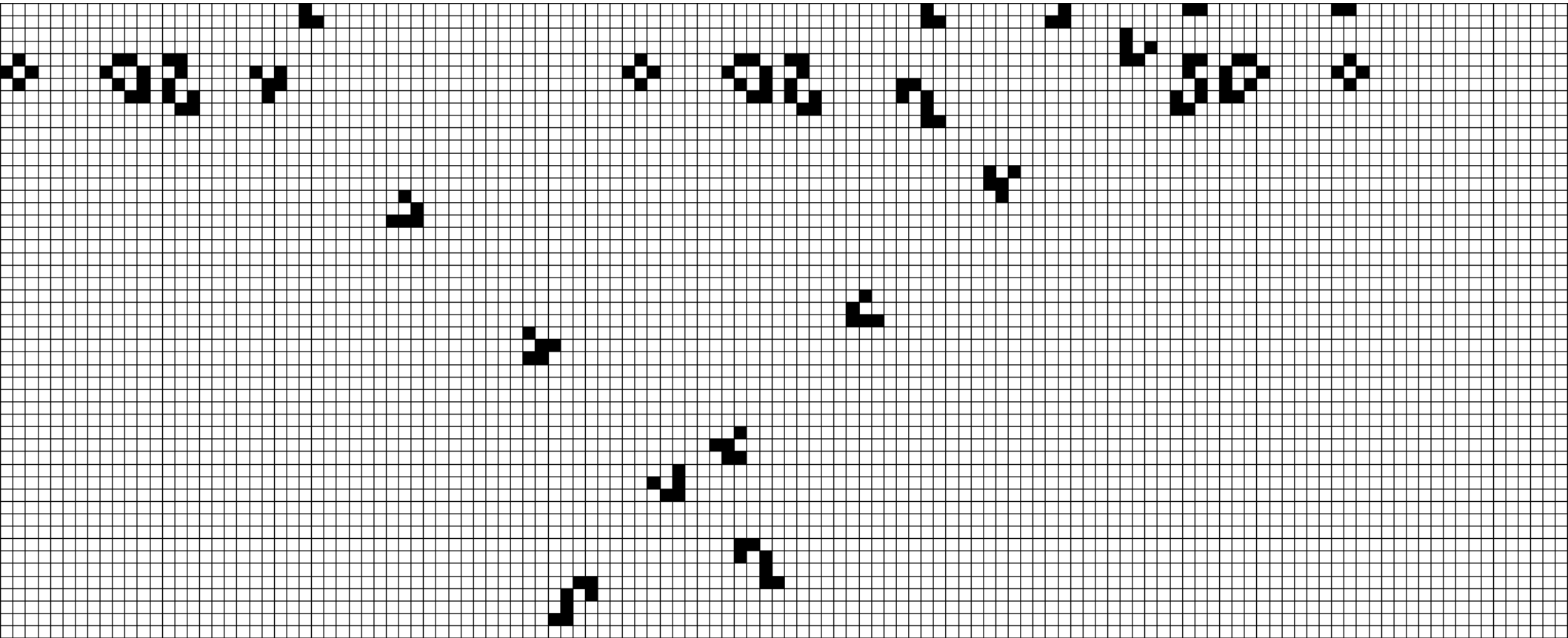
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

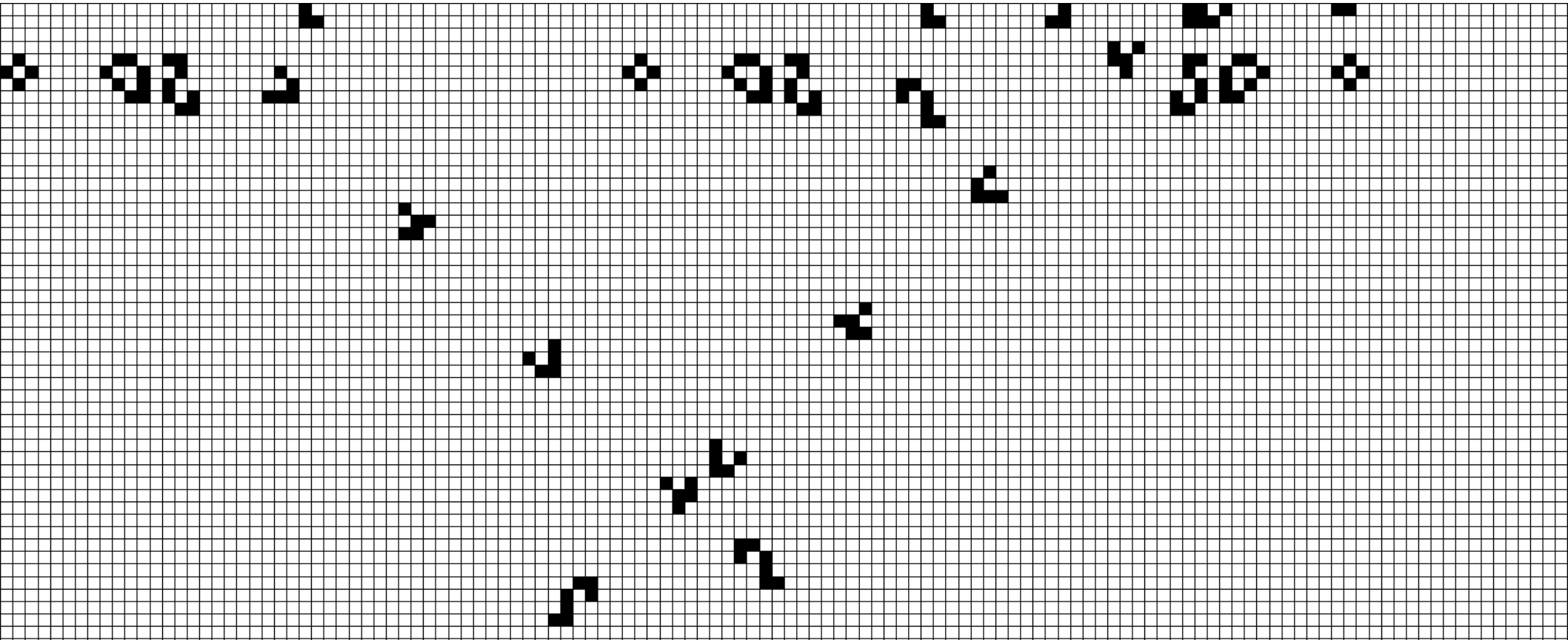
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

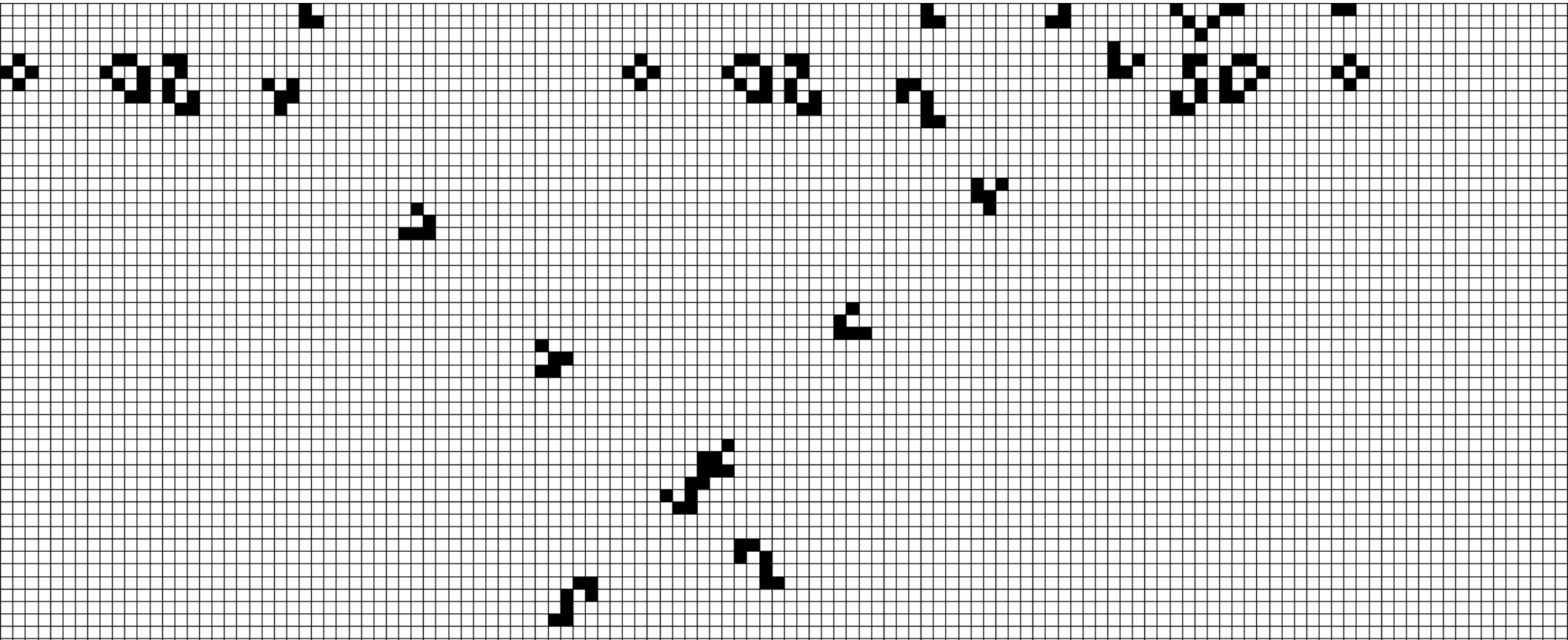
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

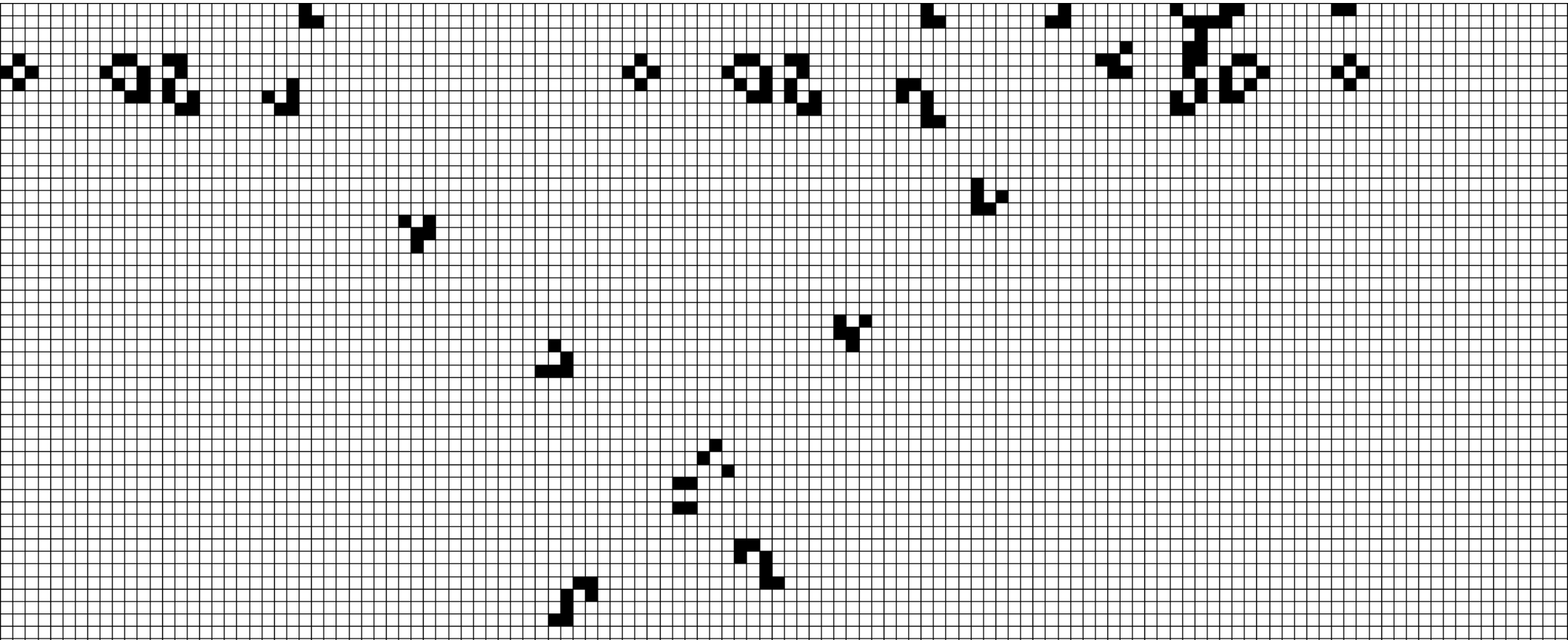
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

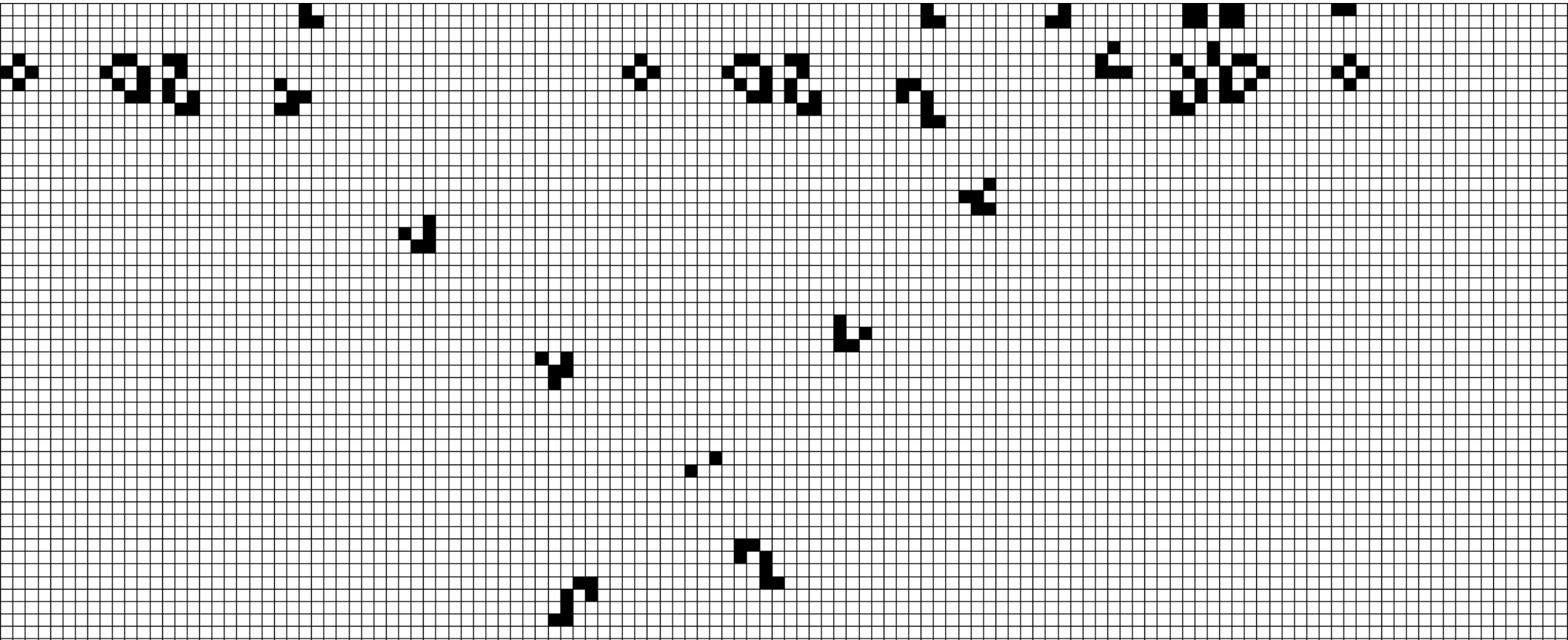
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

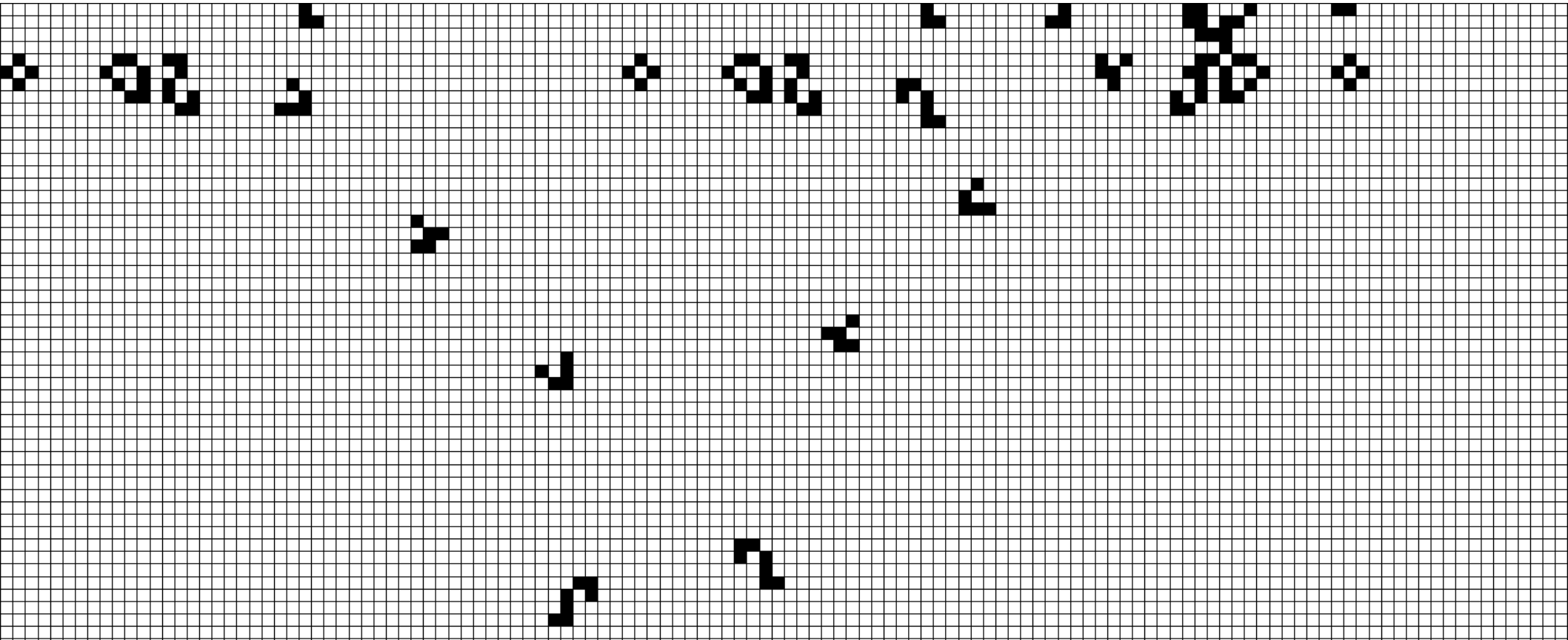
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

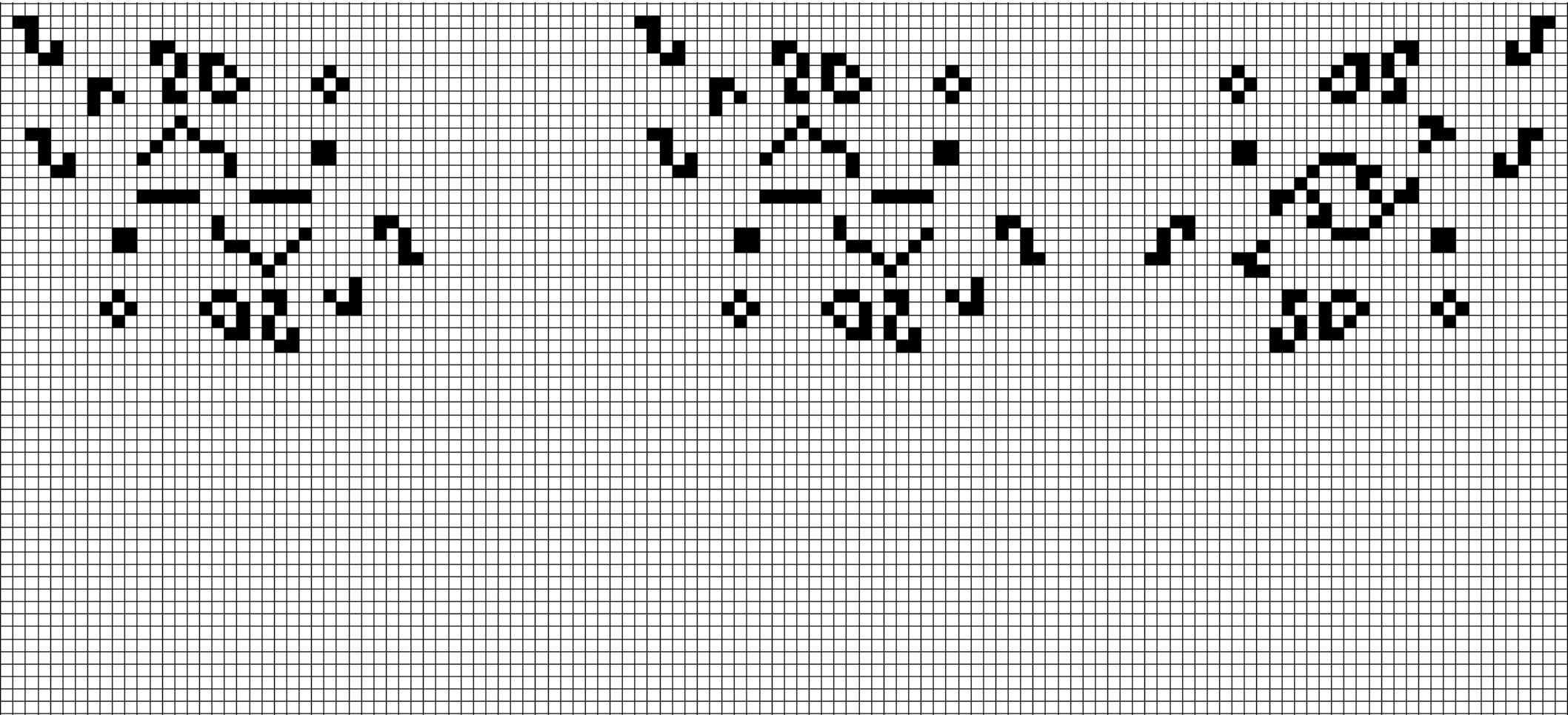
2. Portes logiques



AND : Entrée 1 0

II. Outils nécessaires et composition du filtre

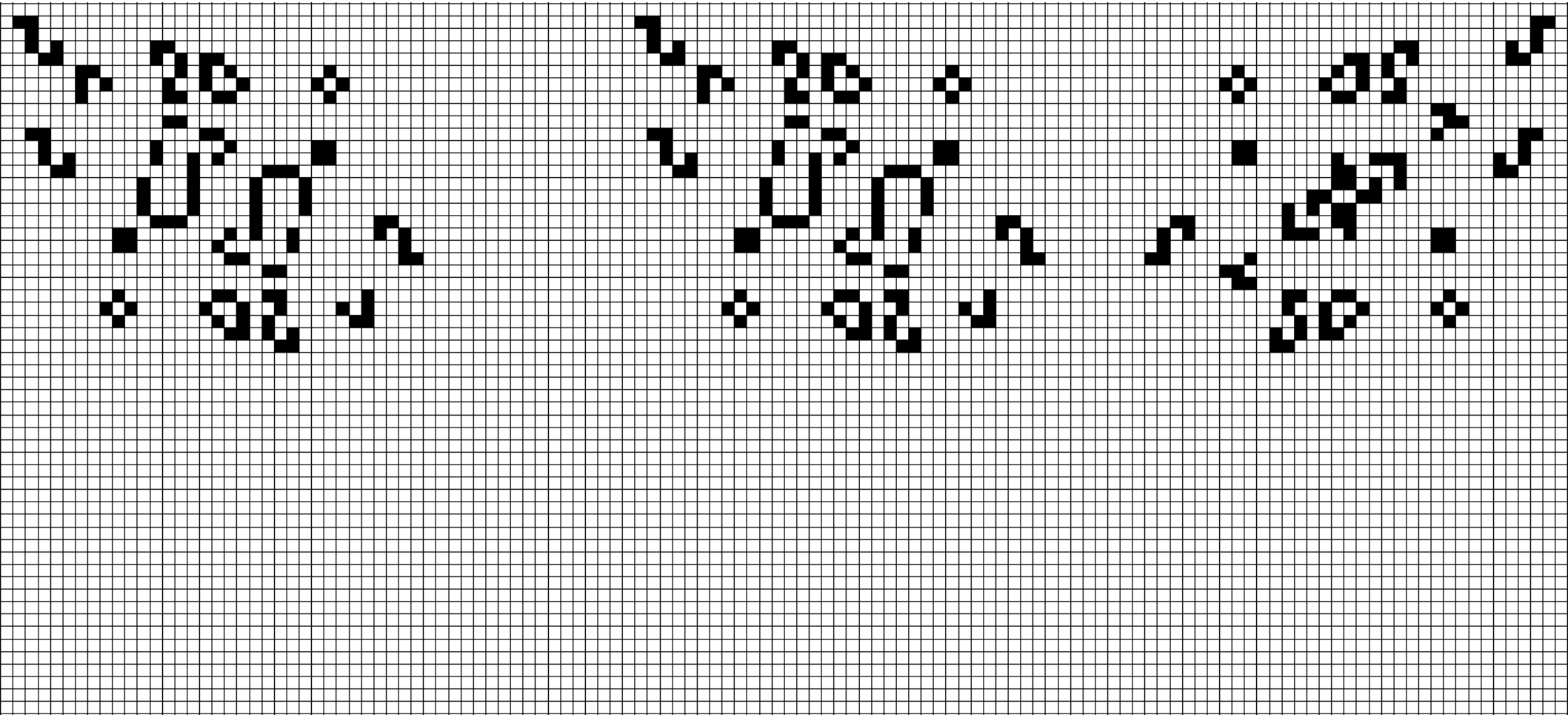
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

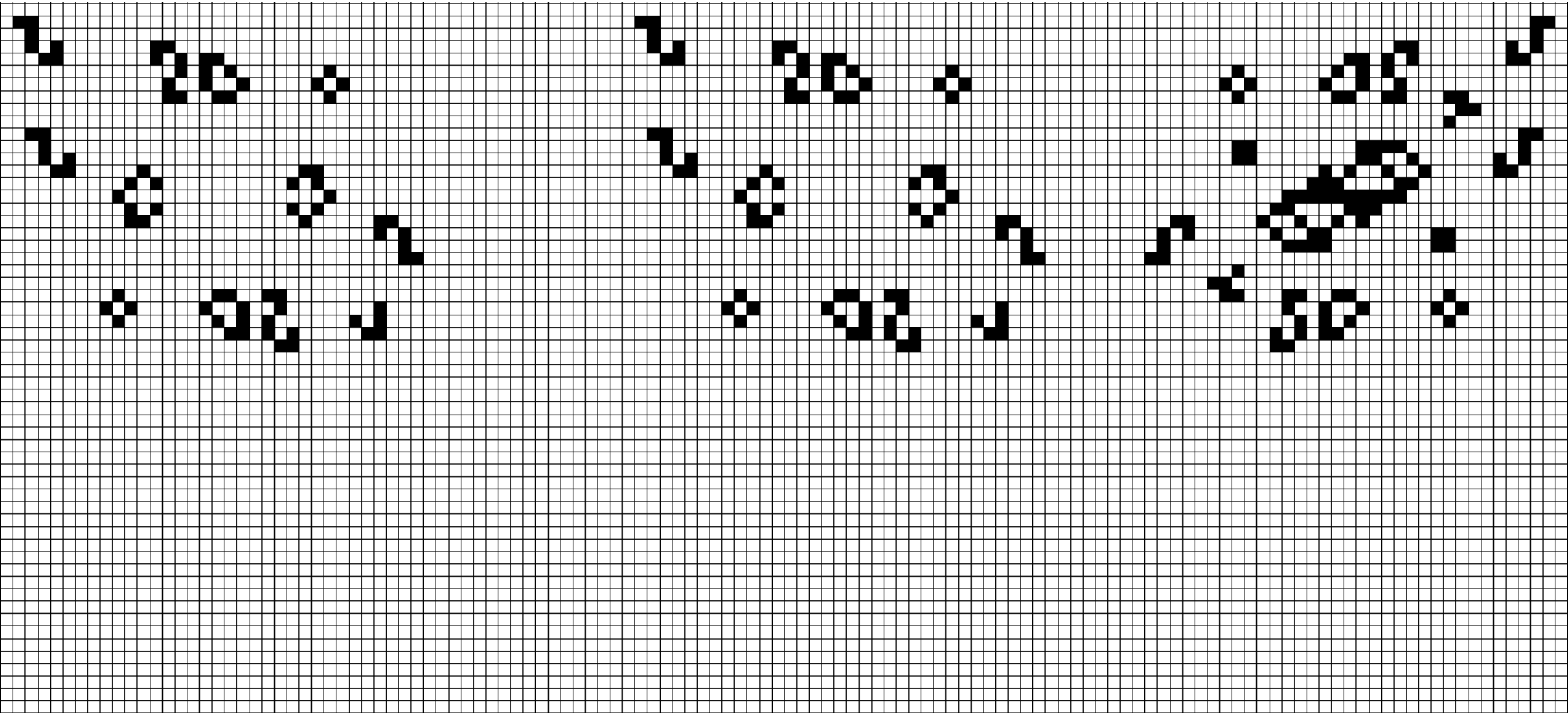
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

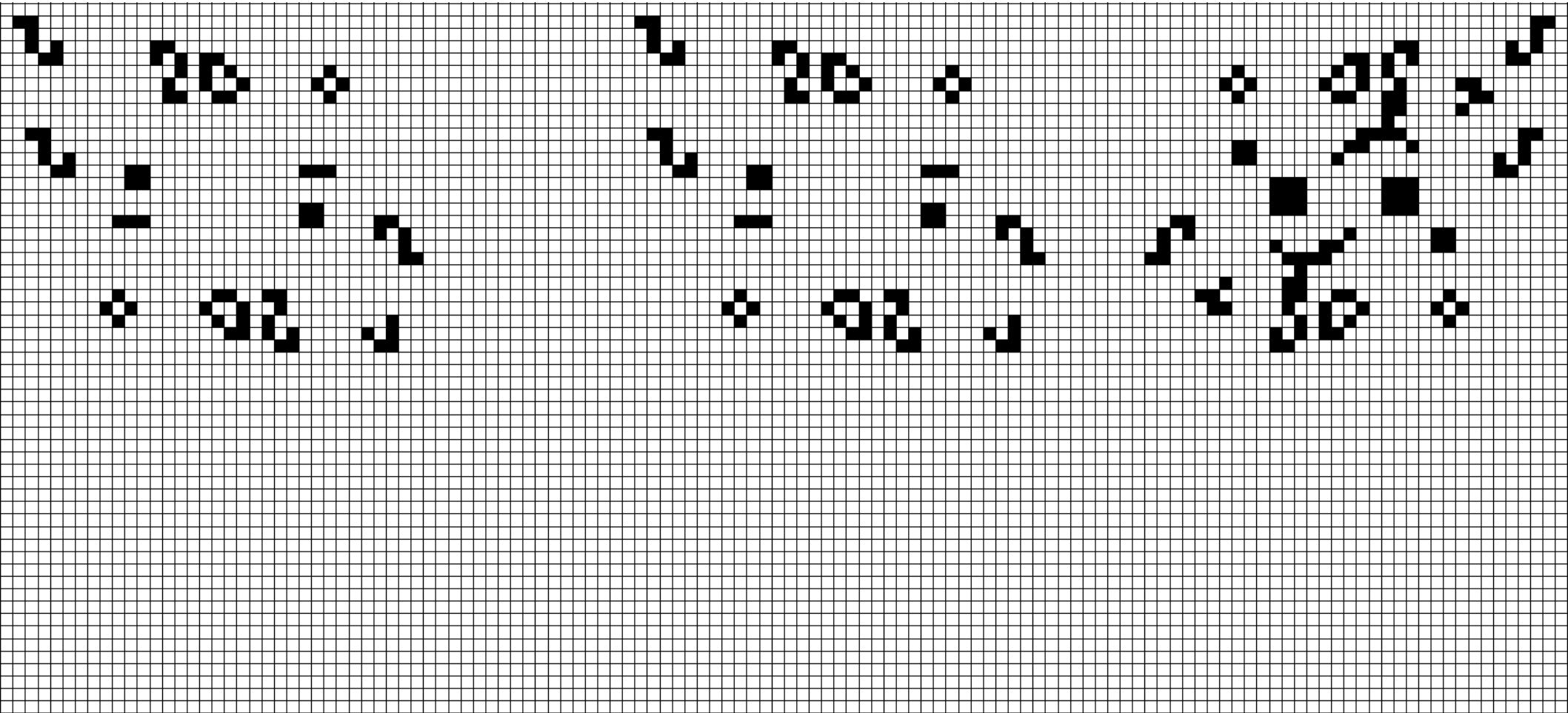
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

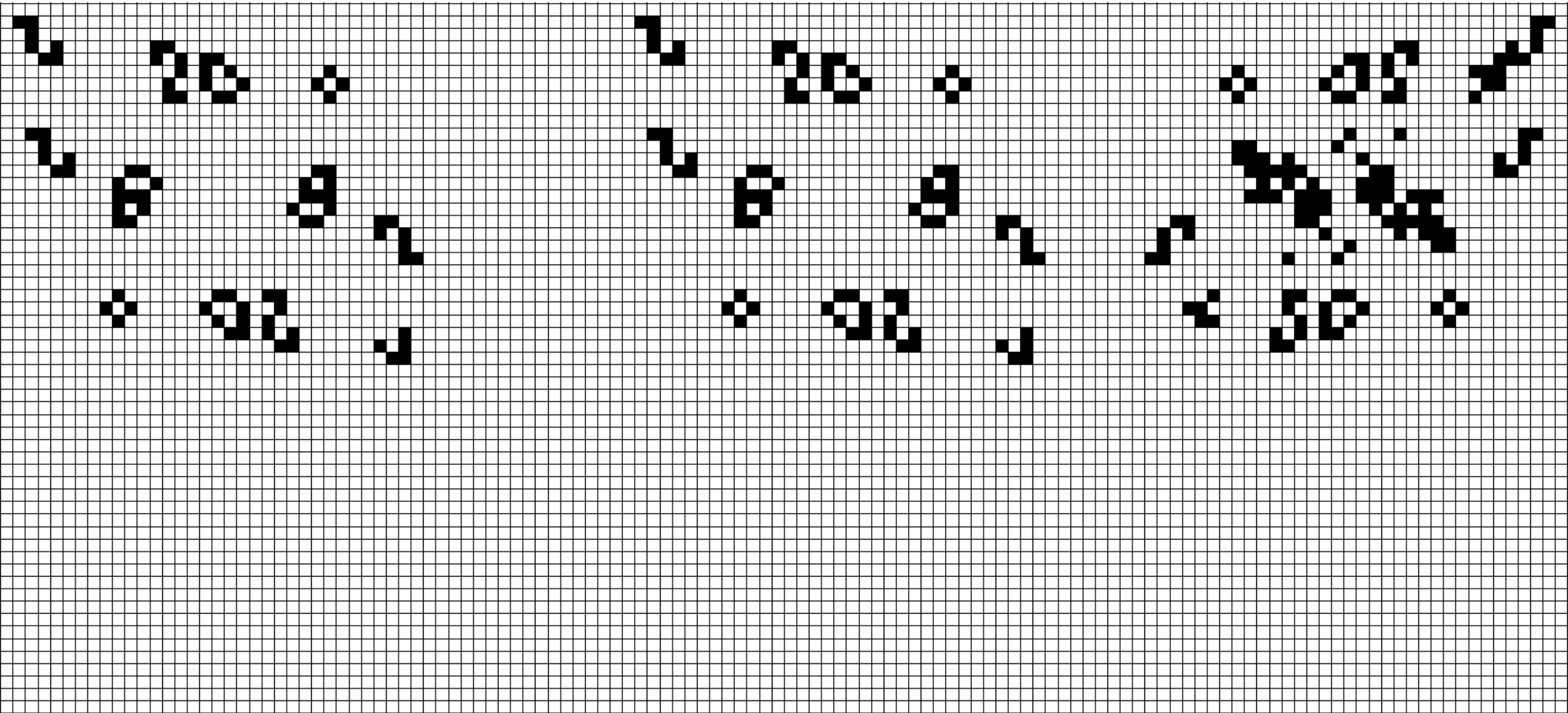
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

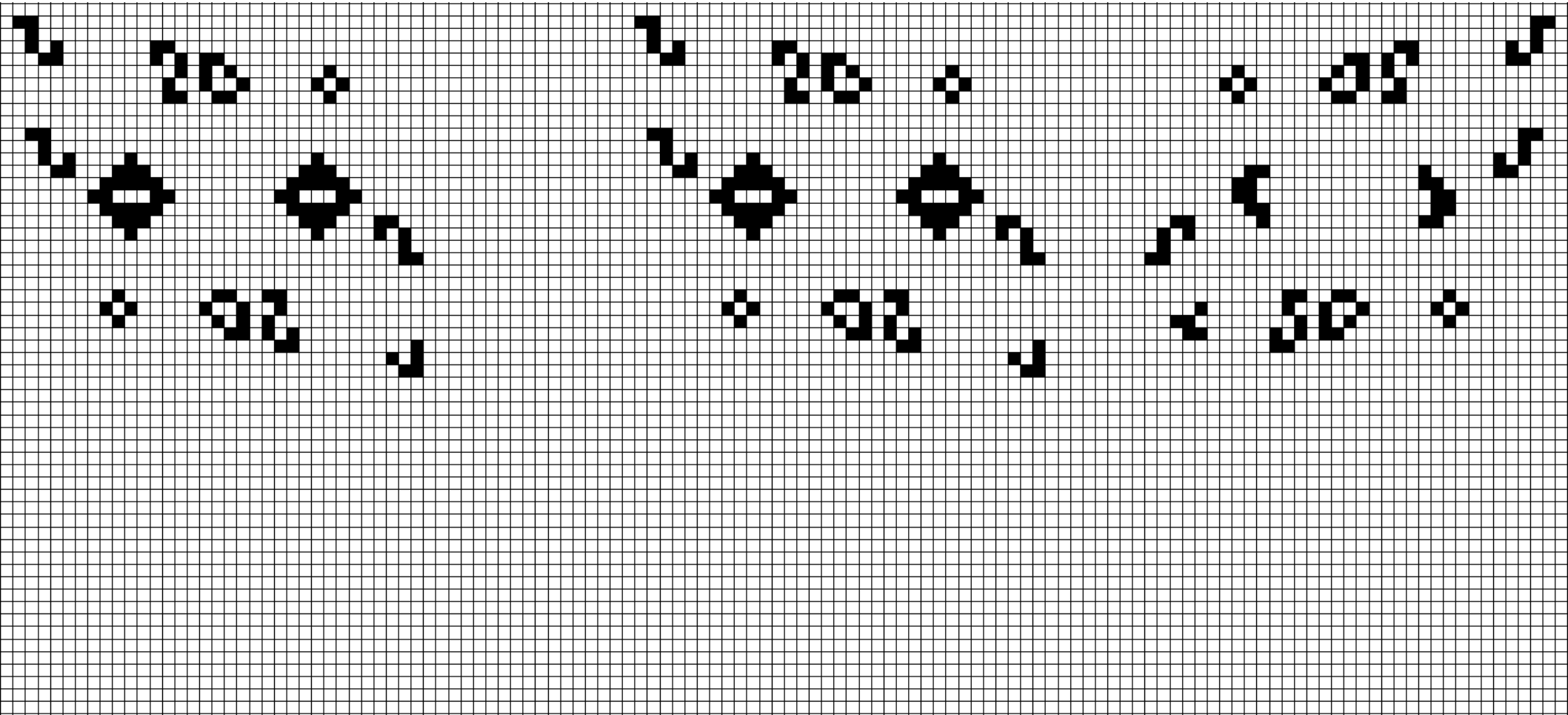
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

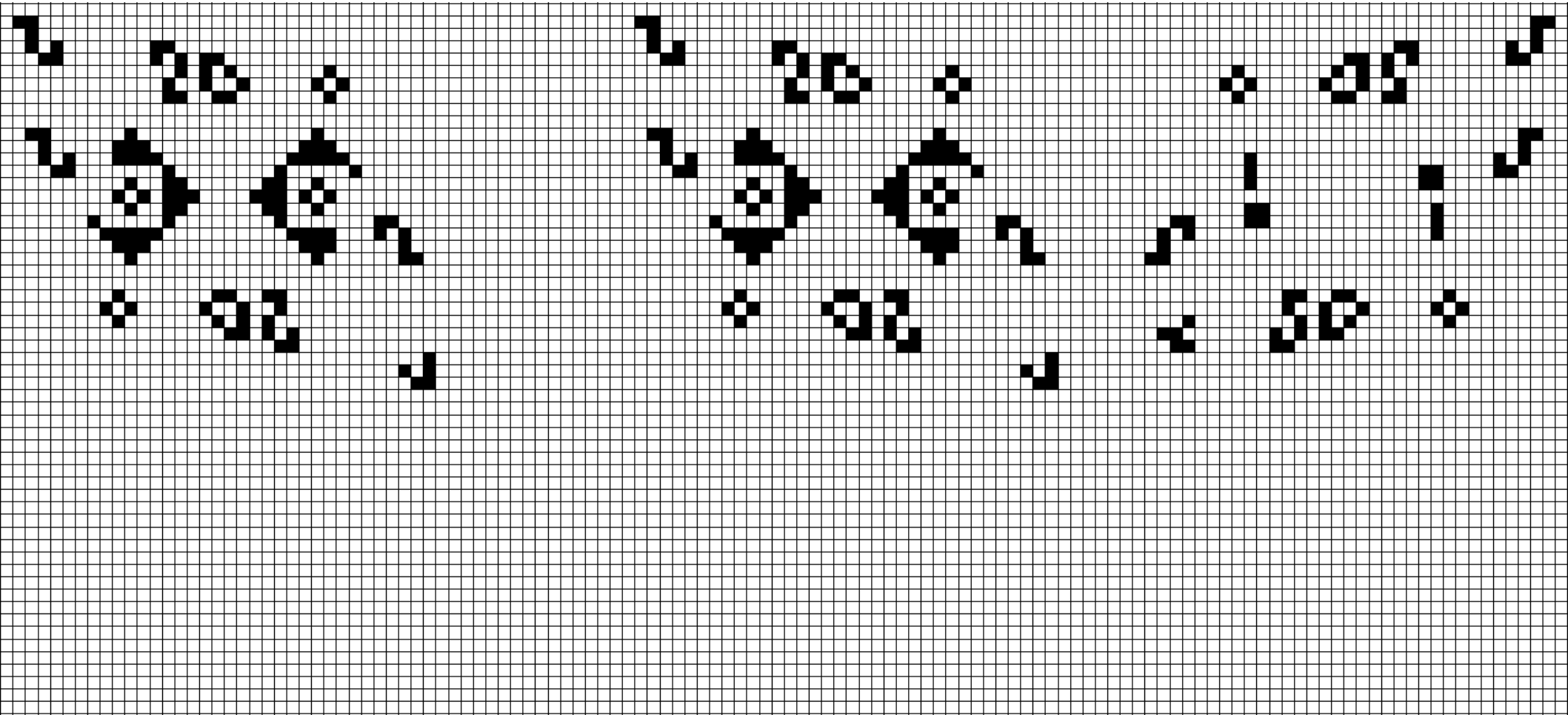
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

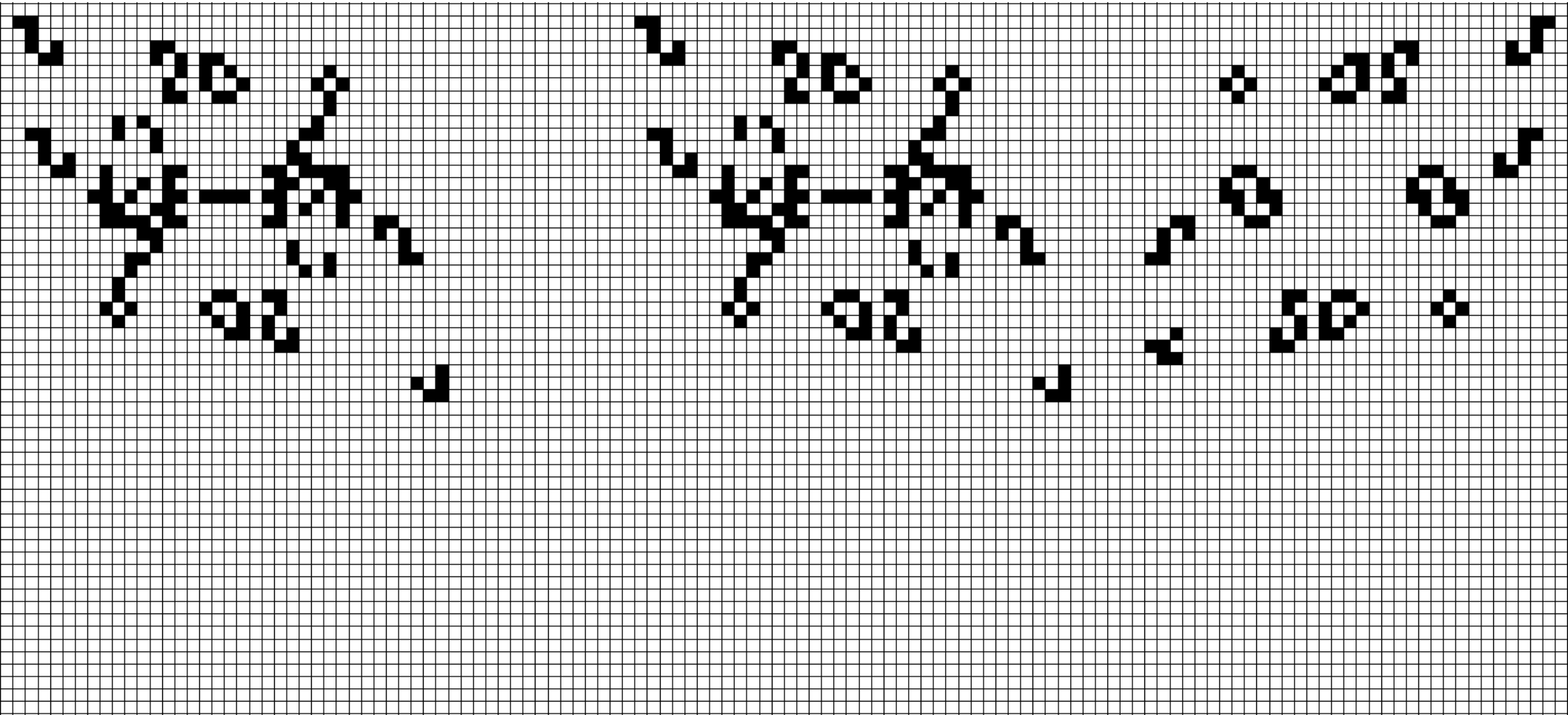
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

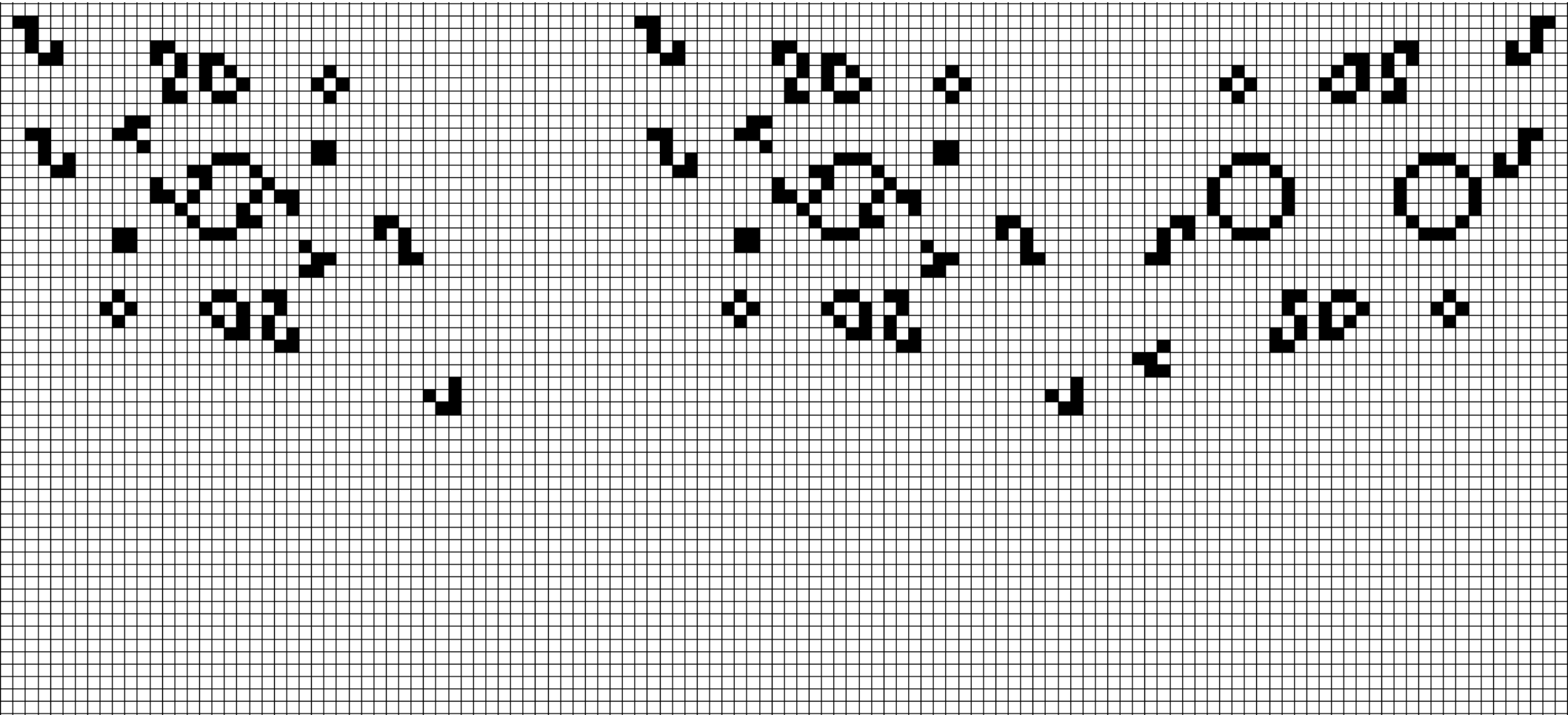
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

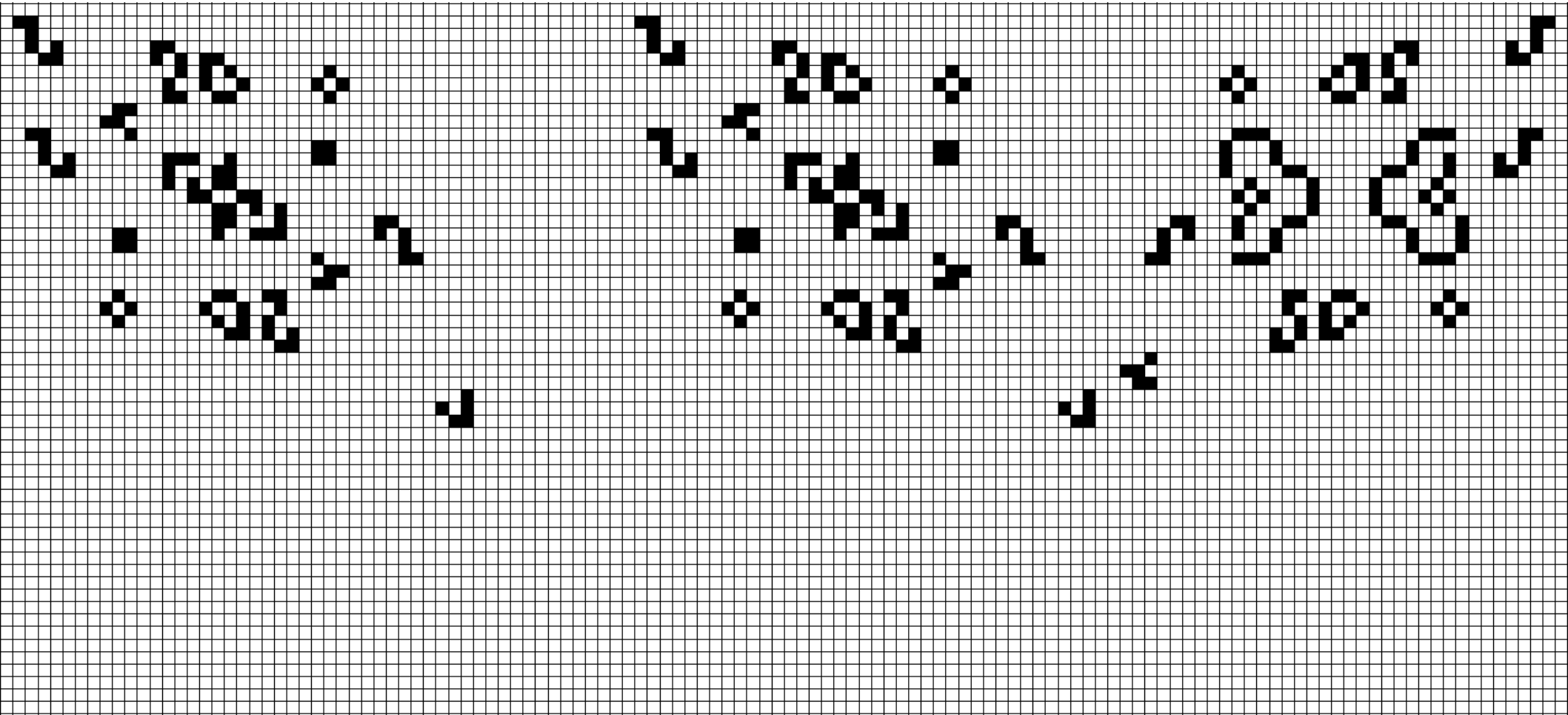
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

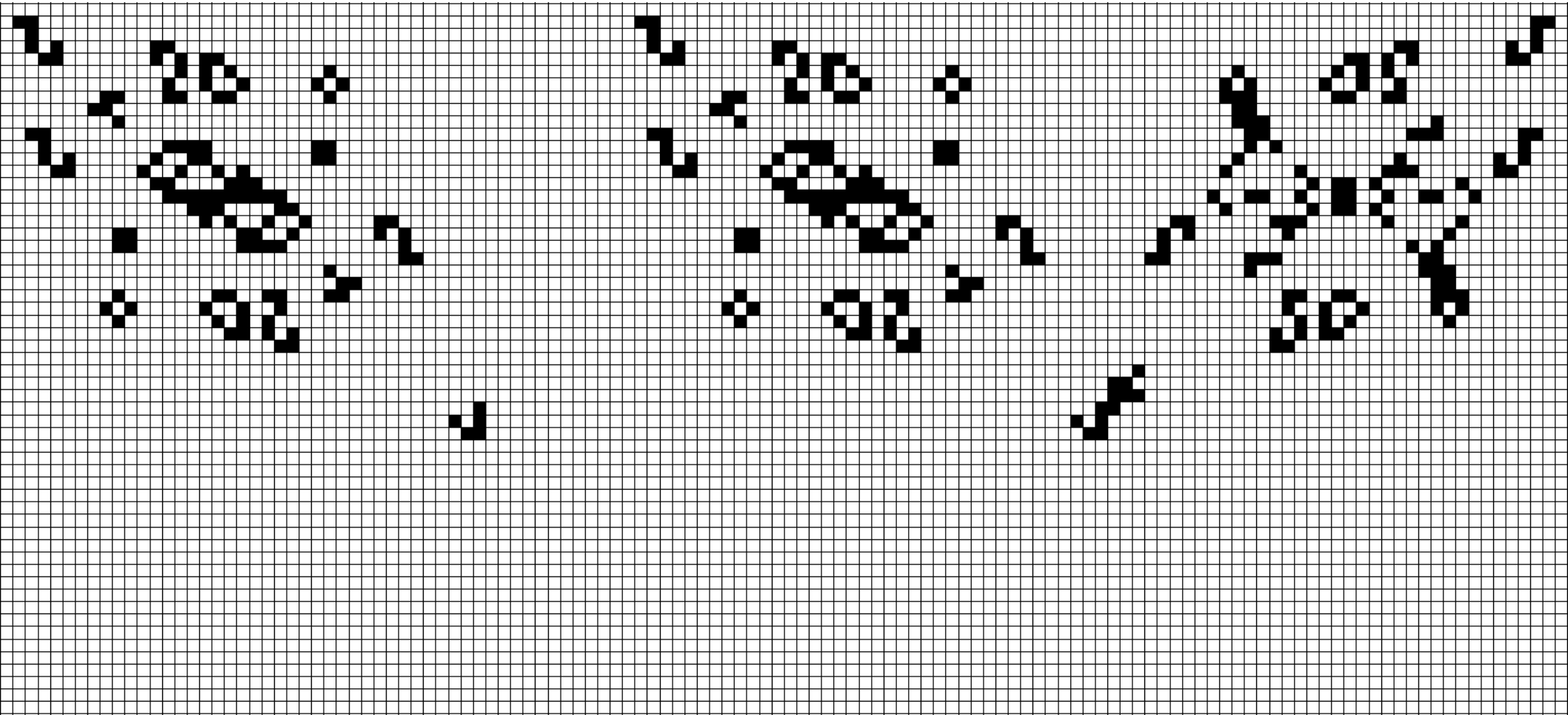
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

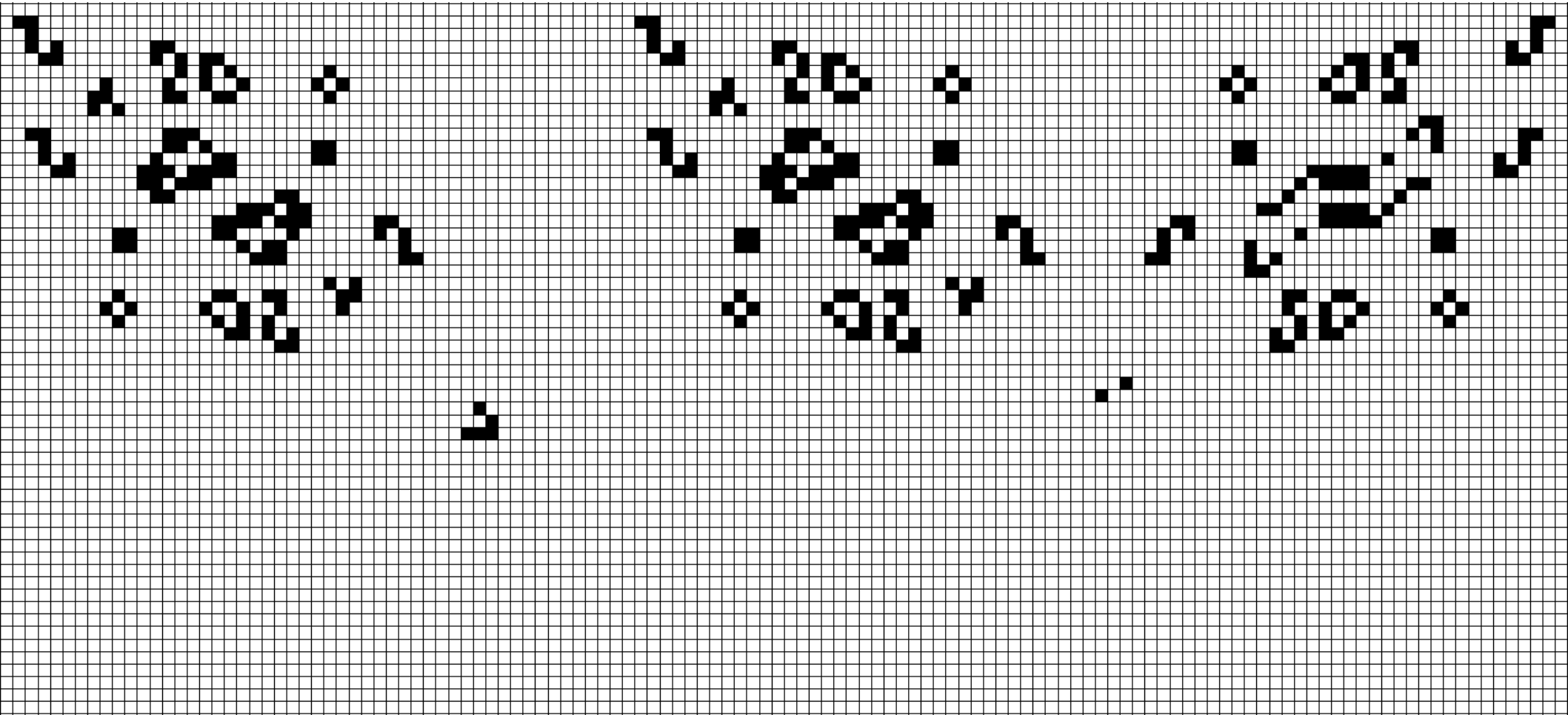
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

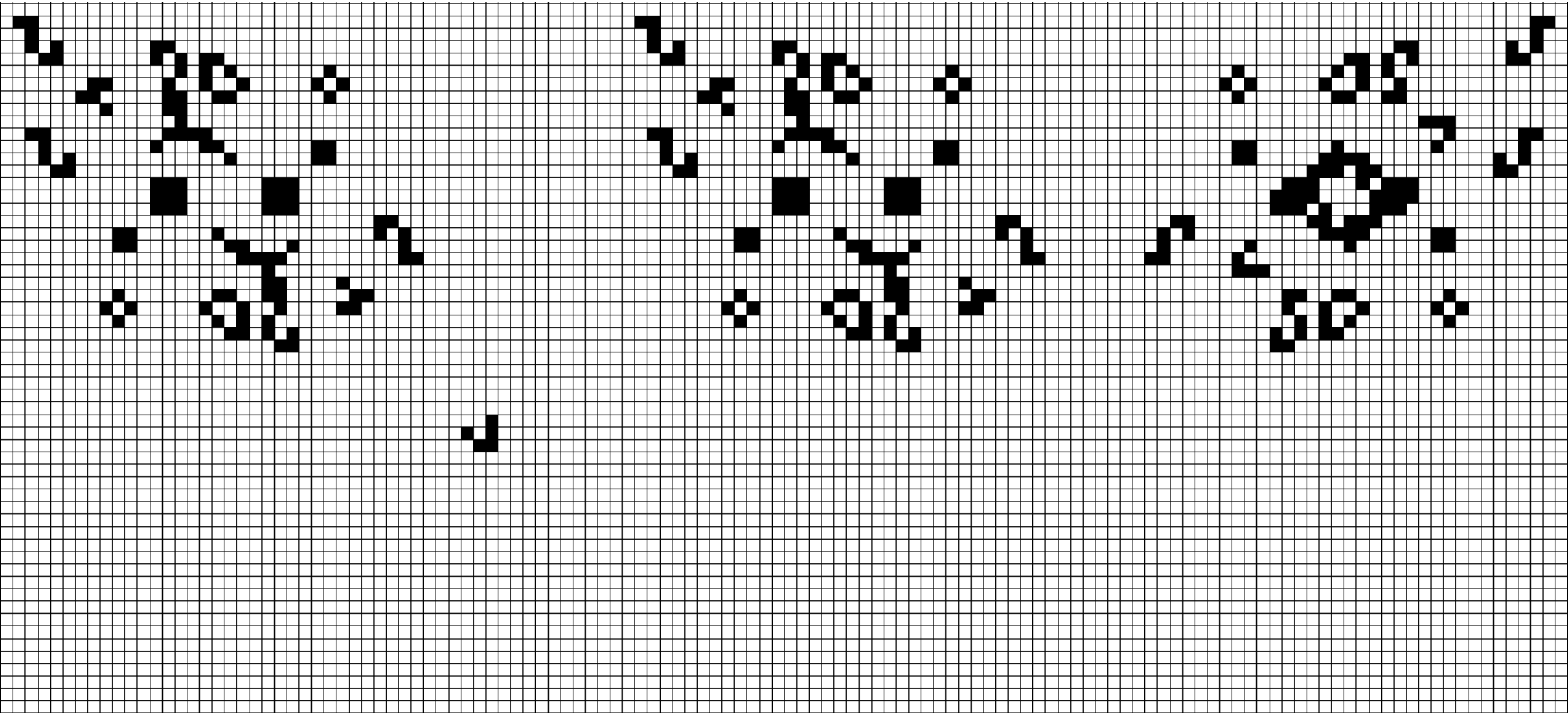
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

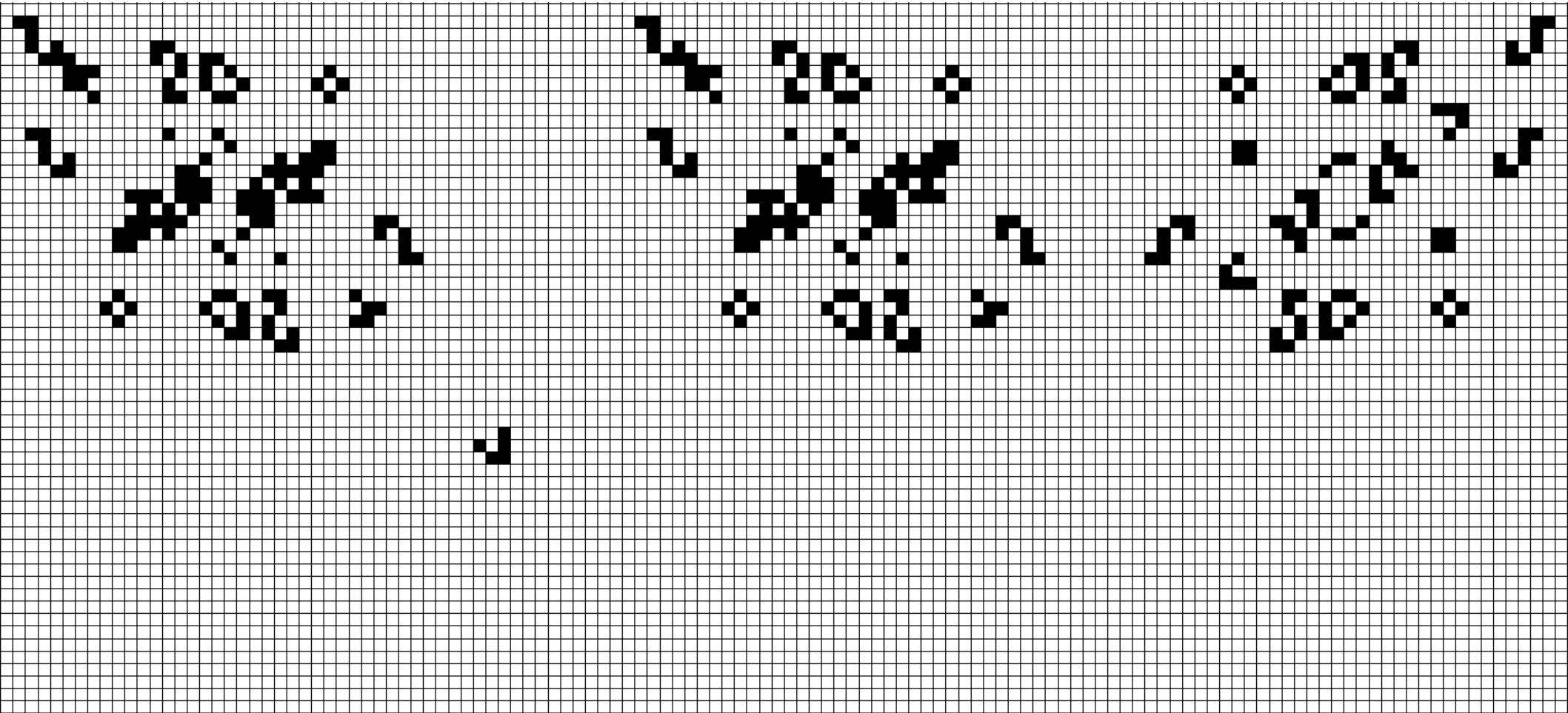
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

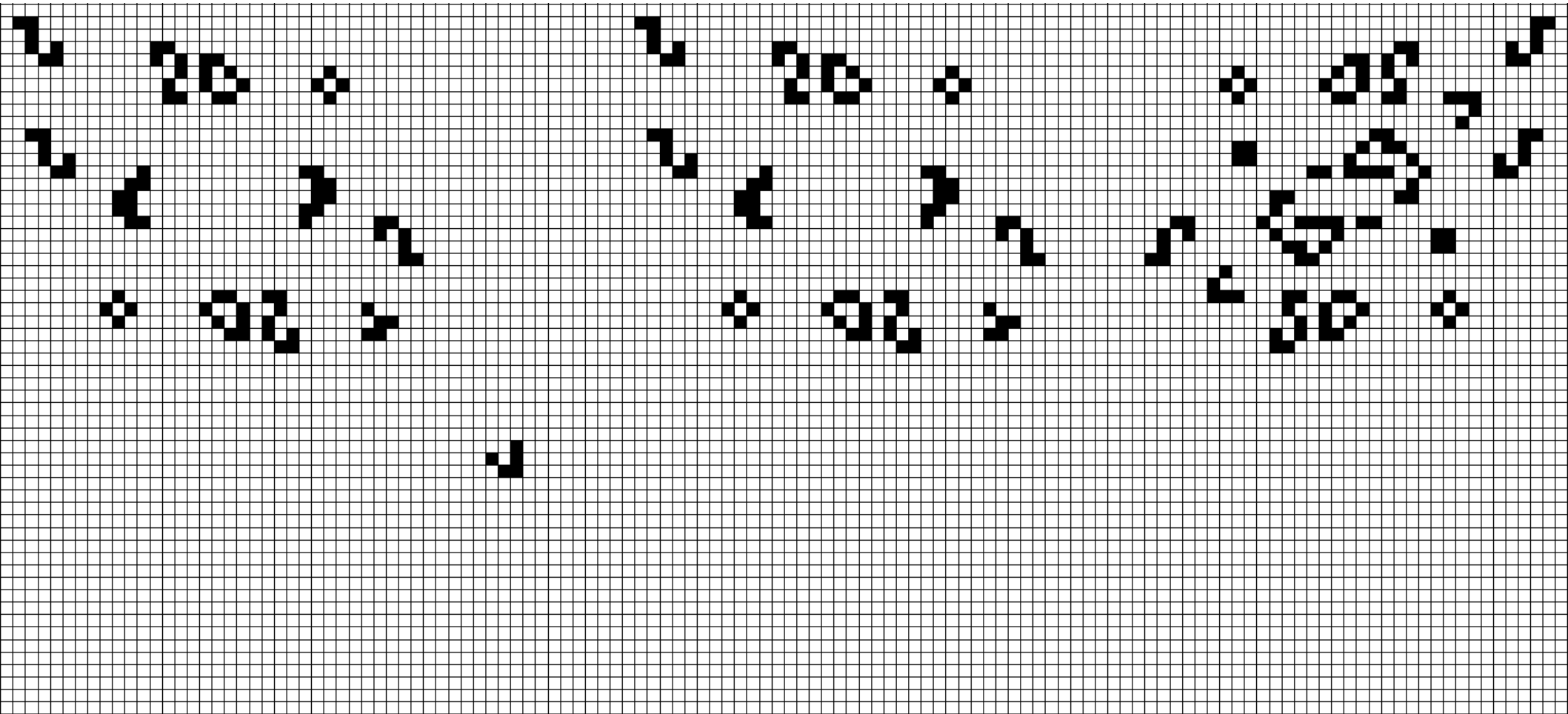
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

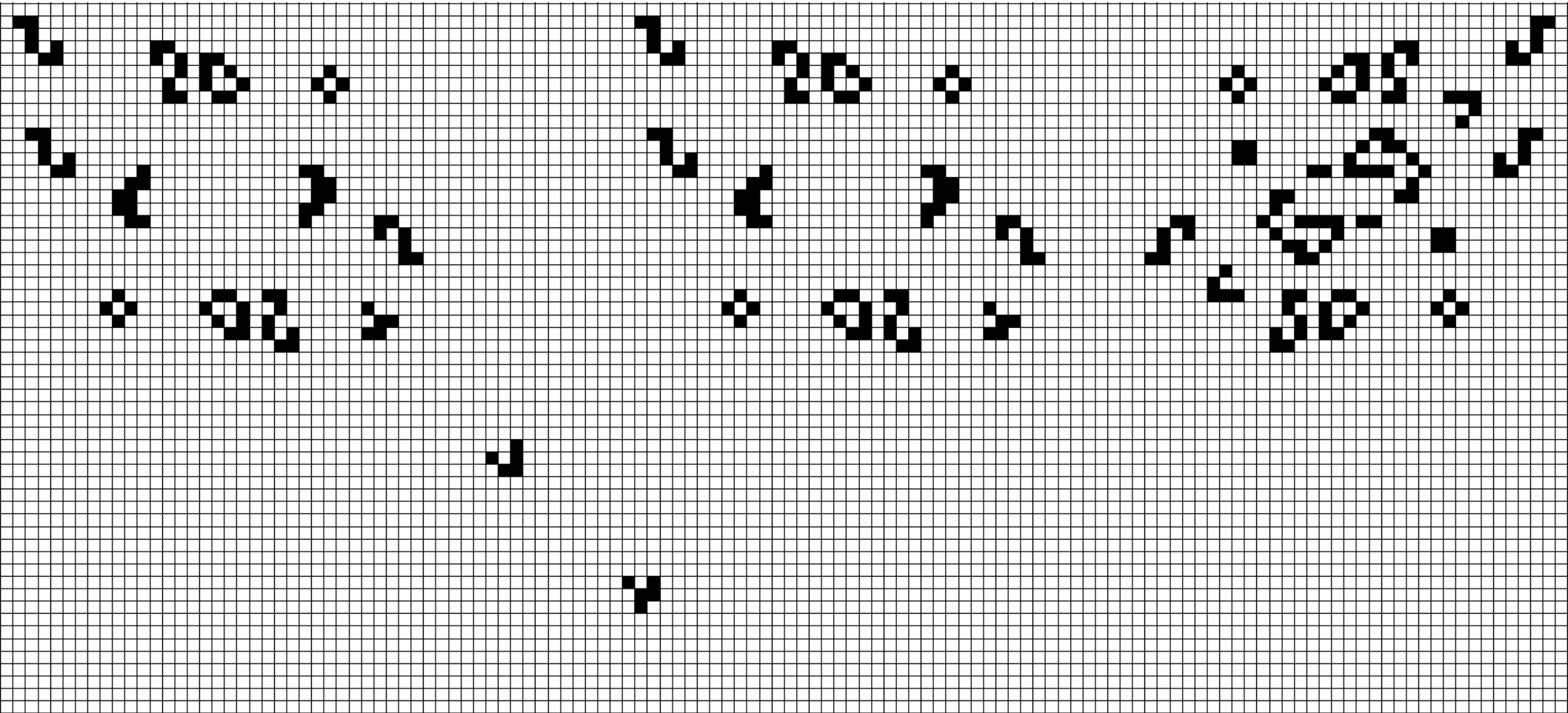
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

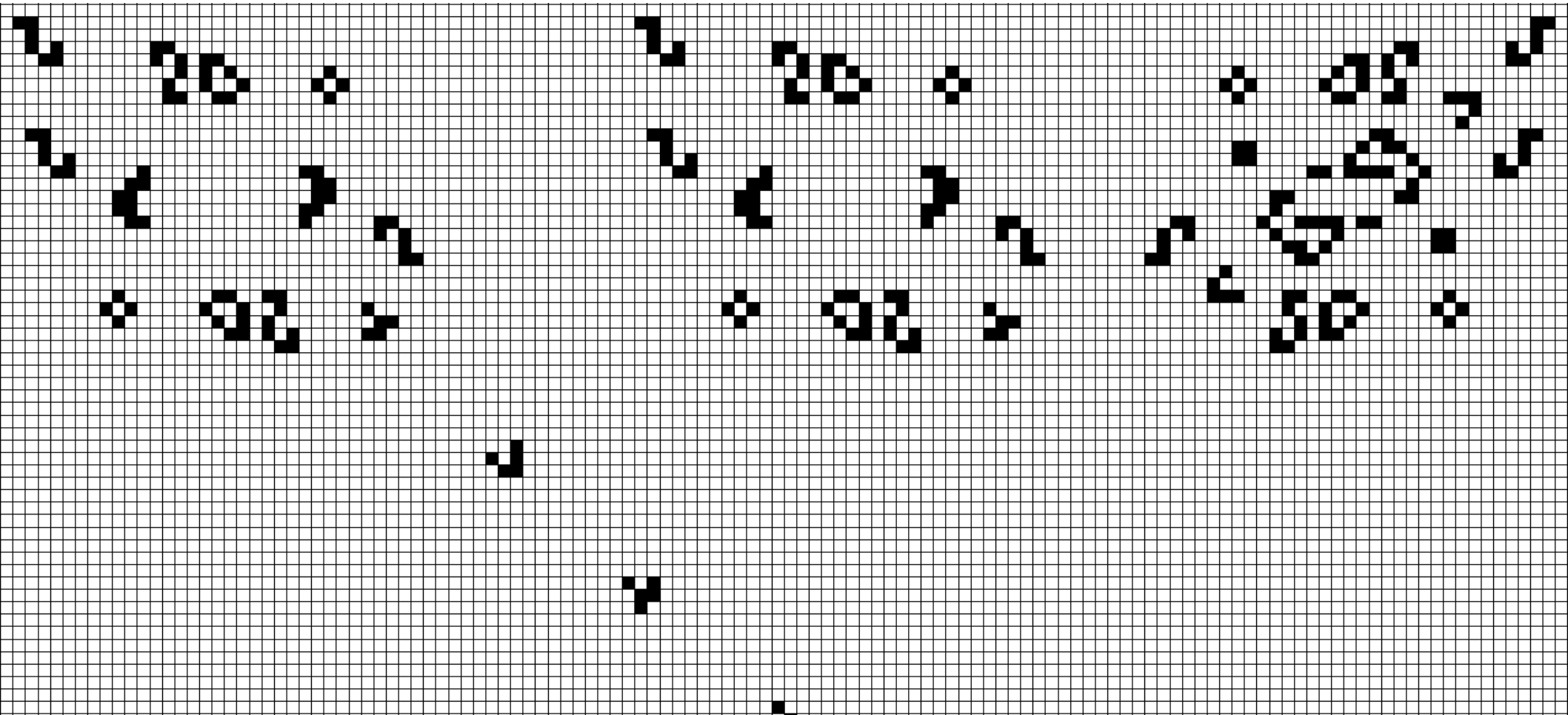
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

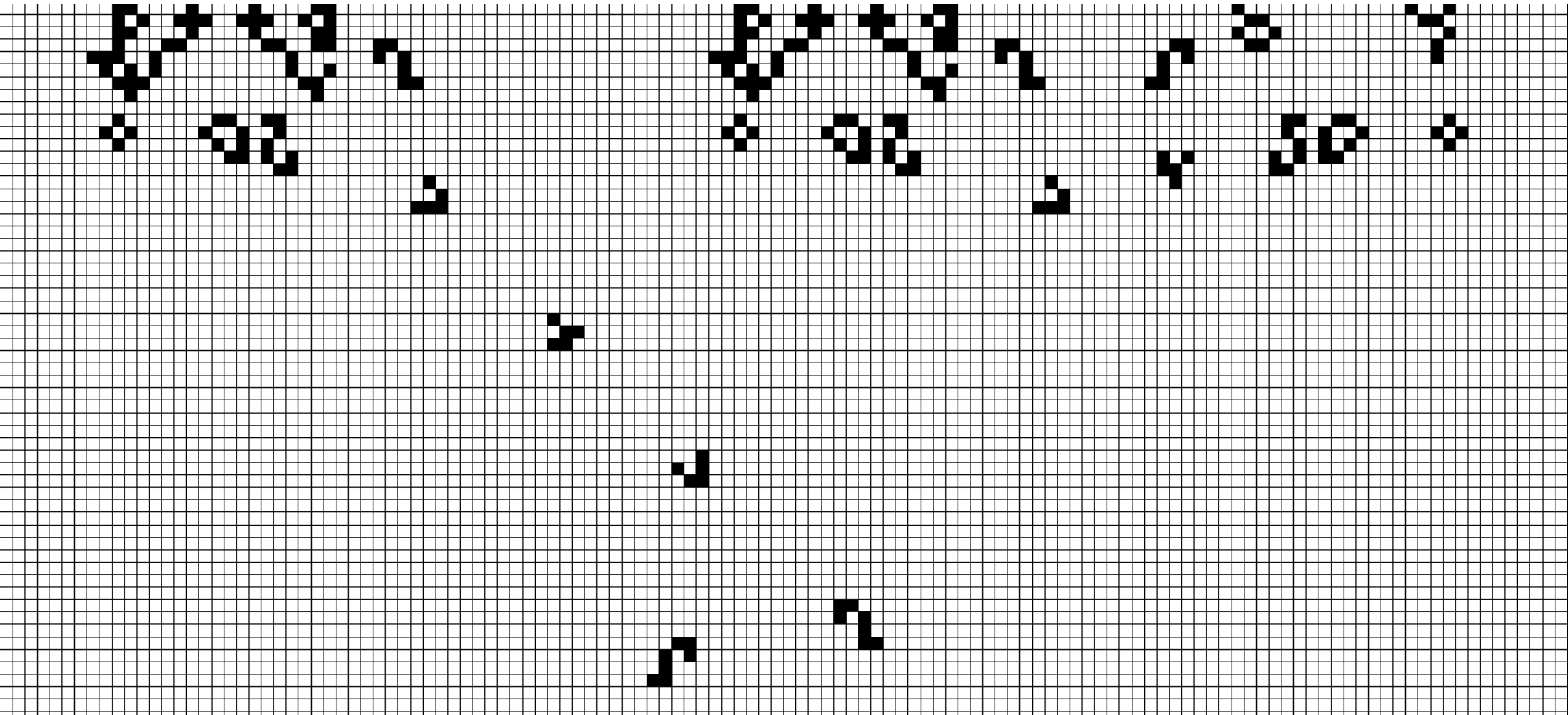
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

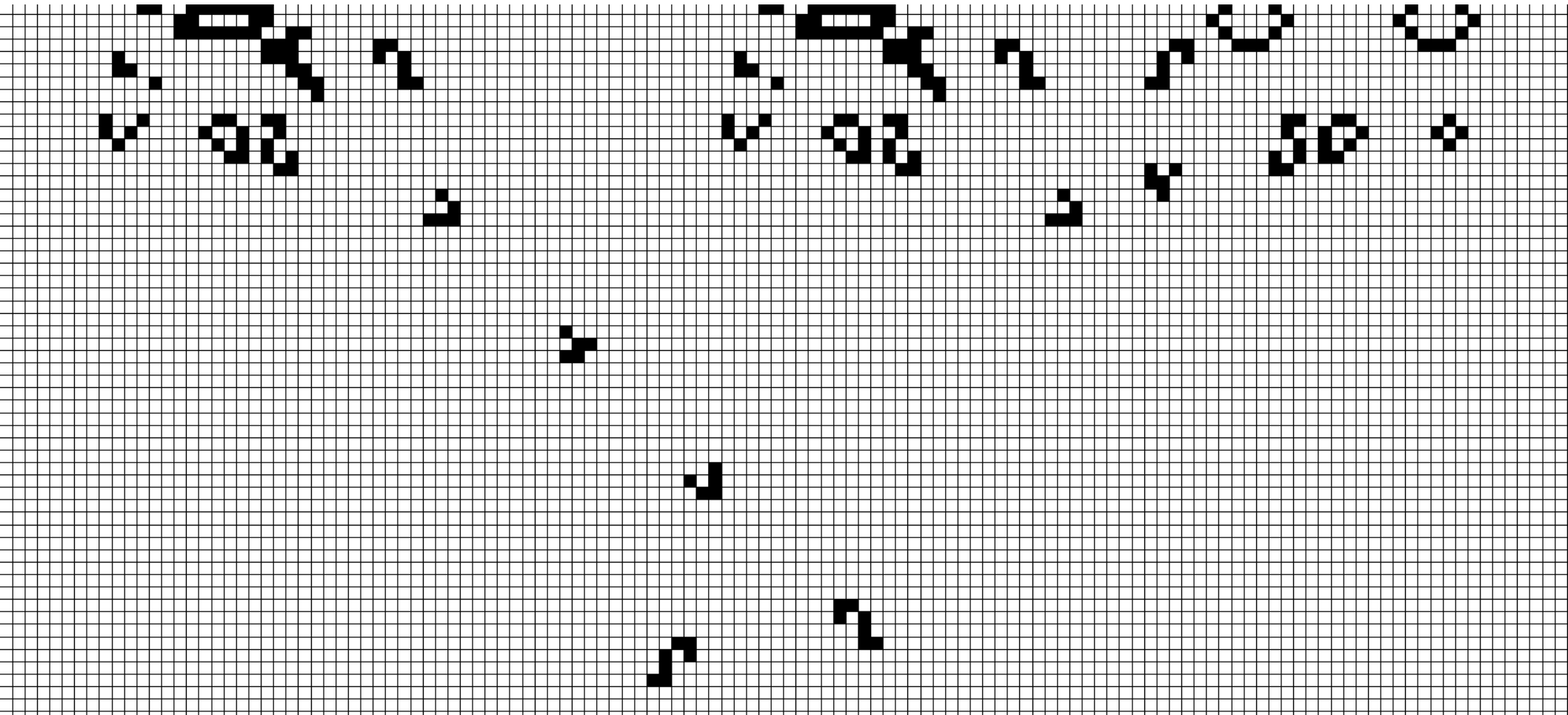
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

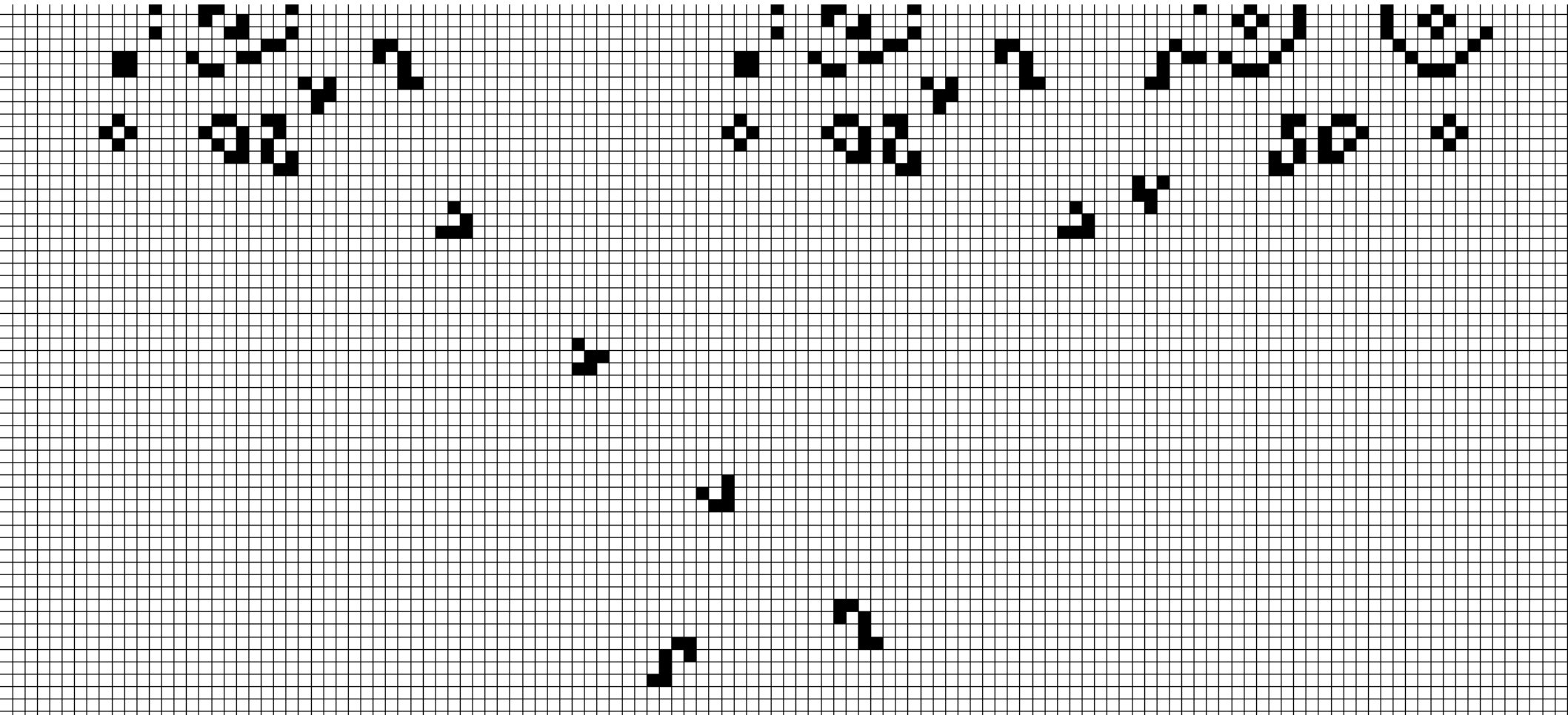
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

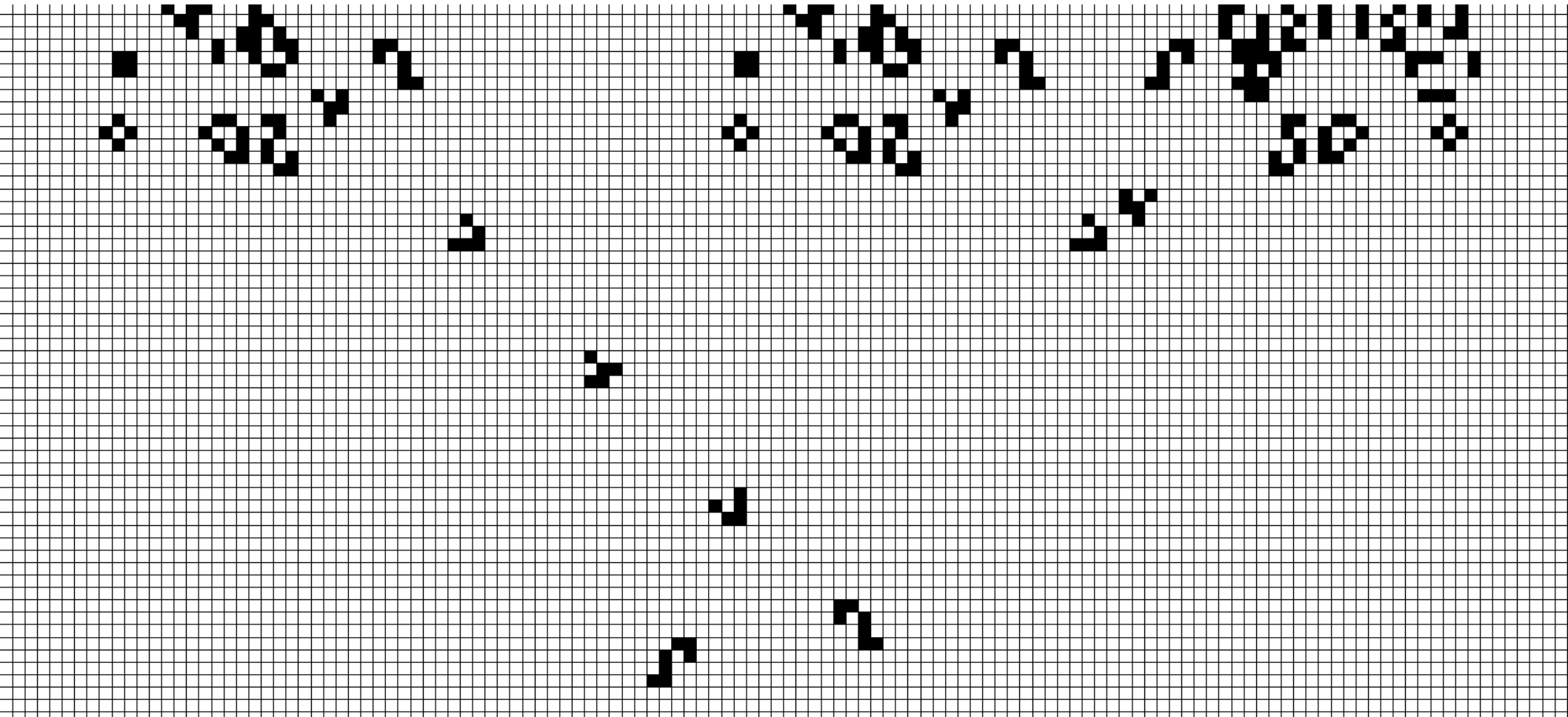
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

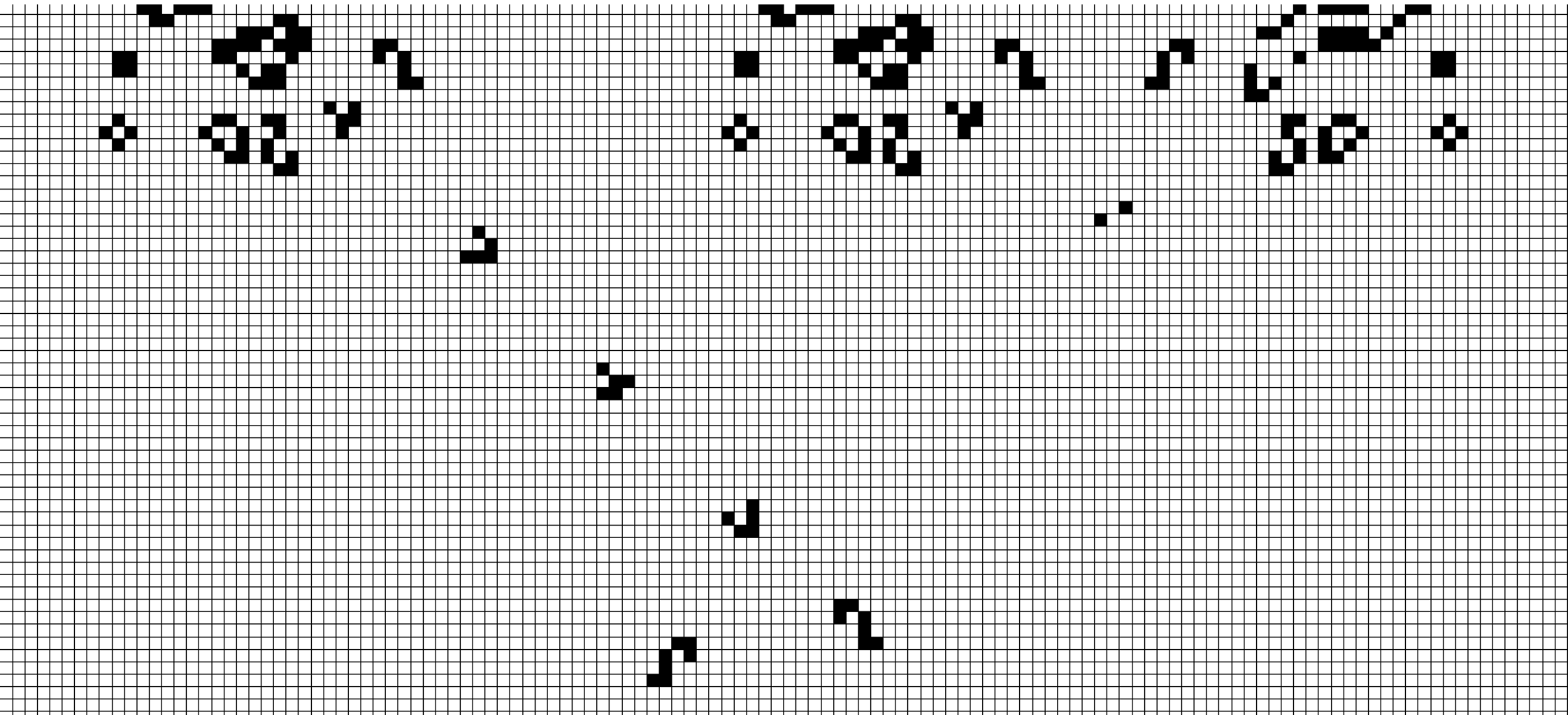
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

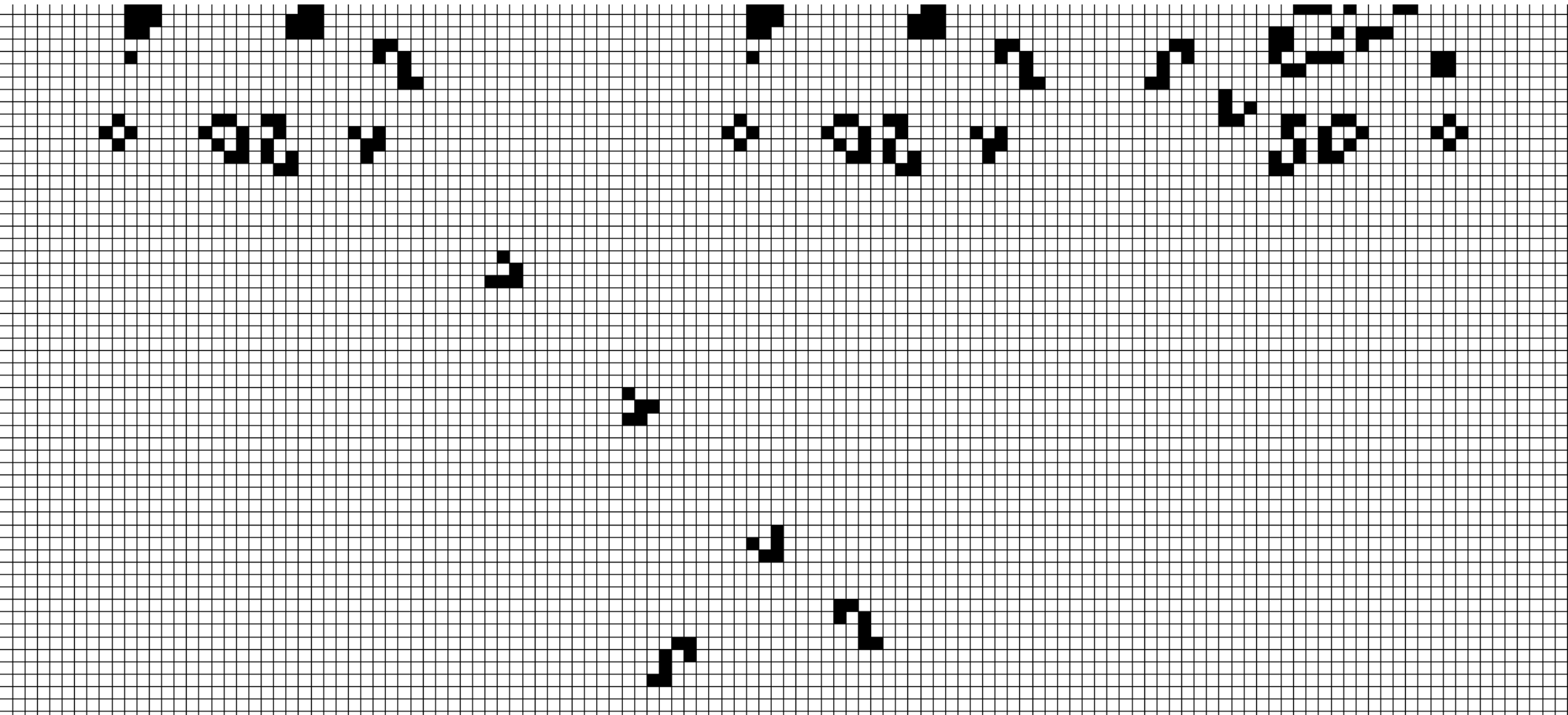
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

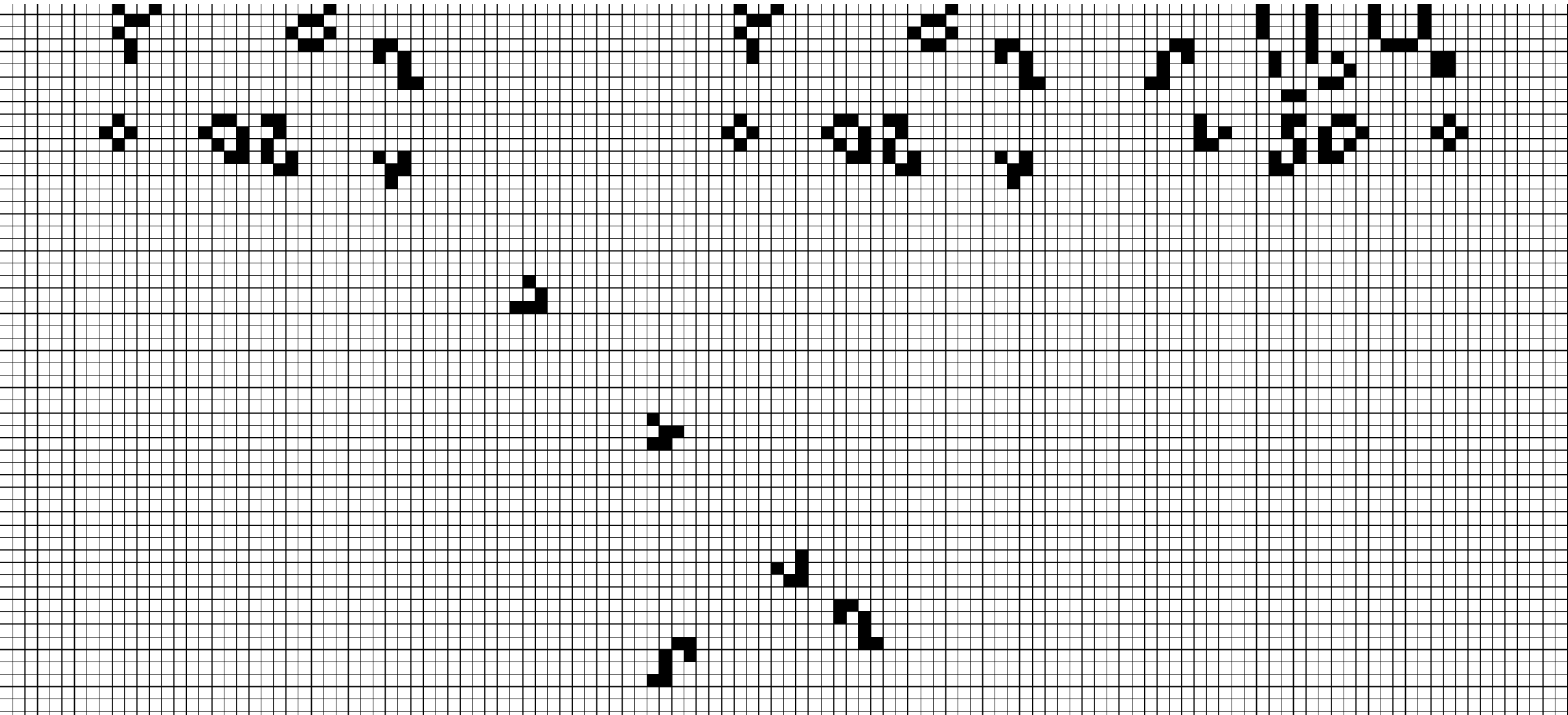
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

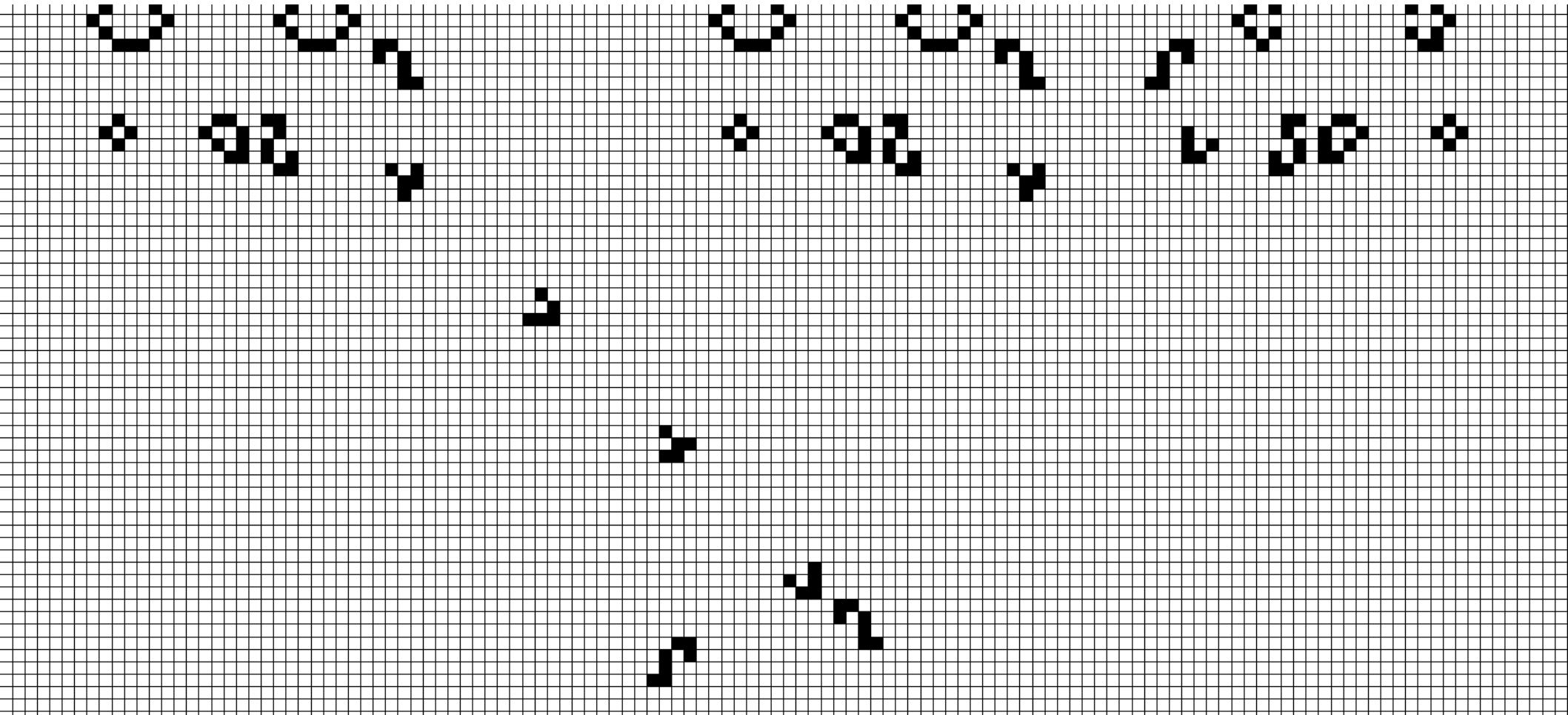
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

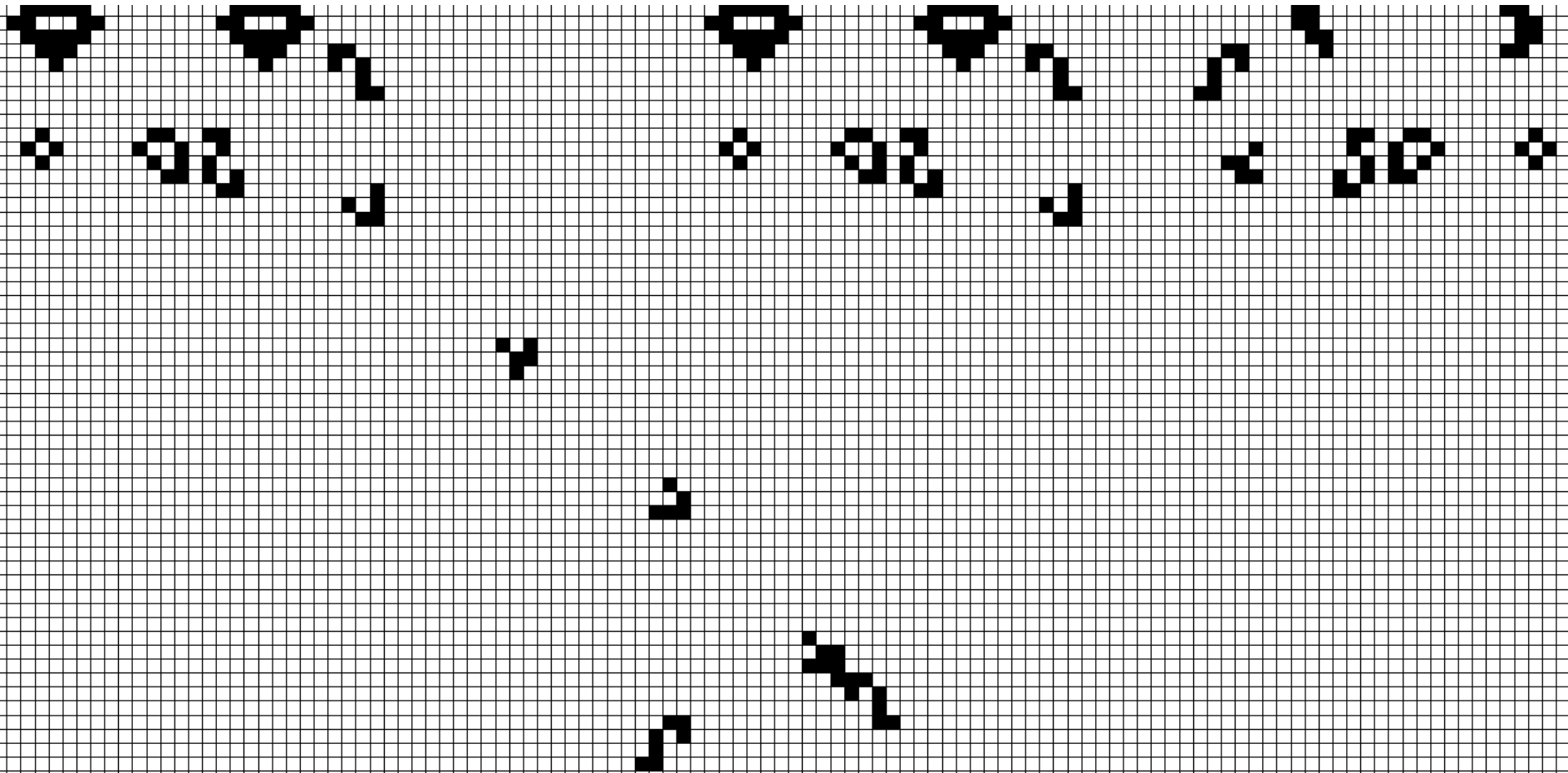
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

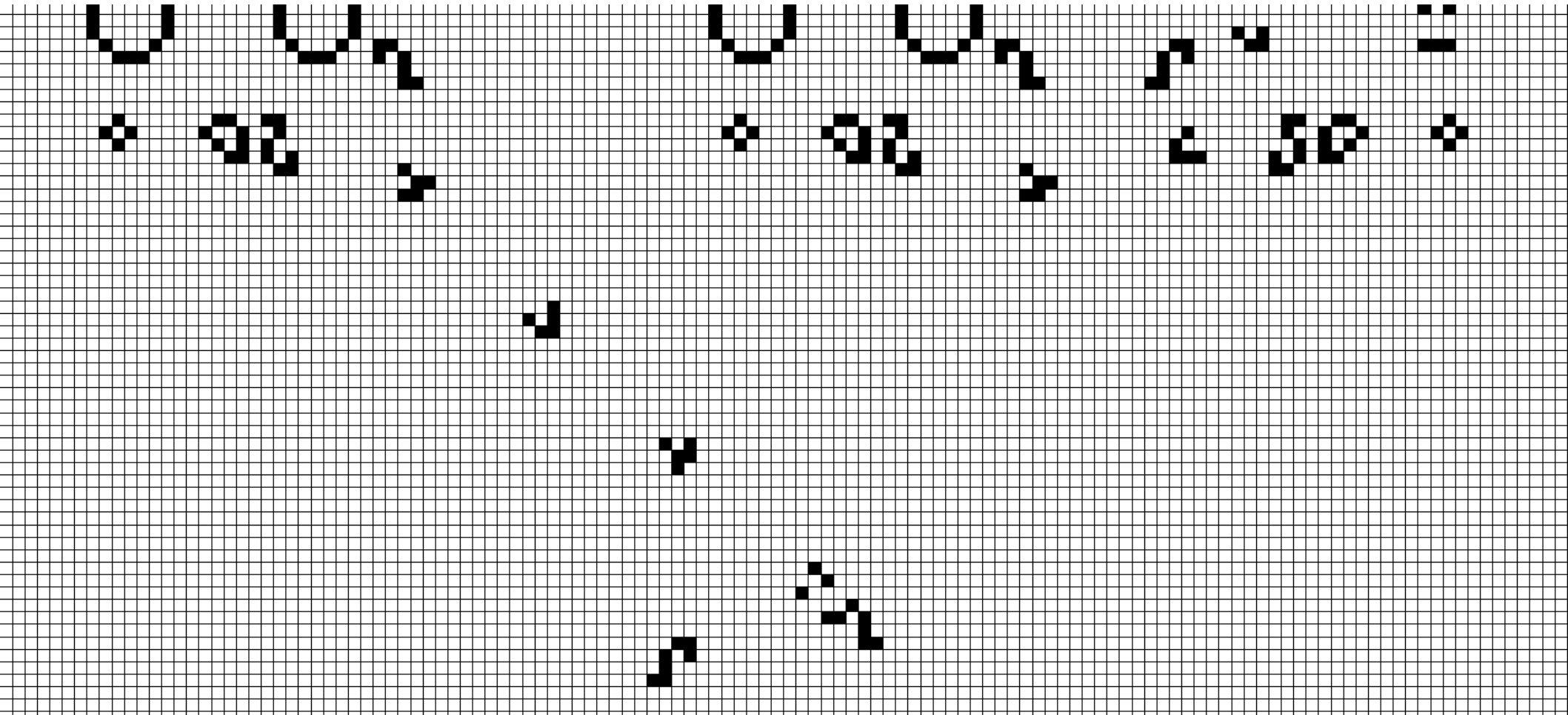
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

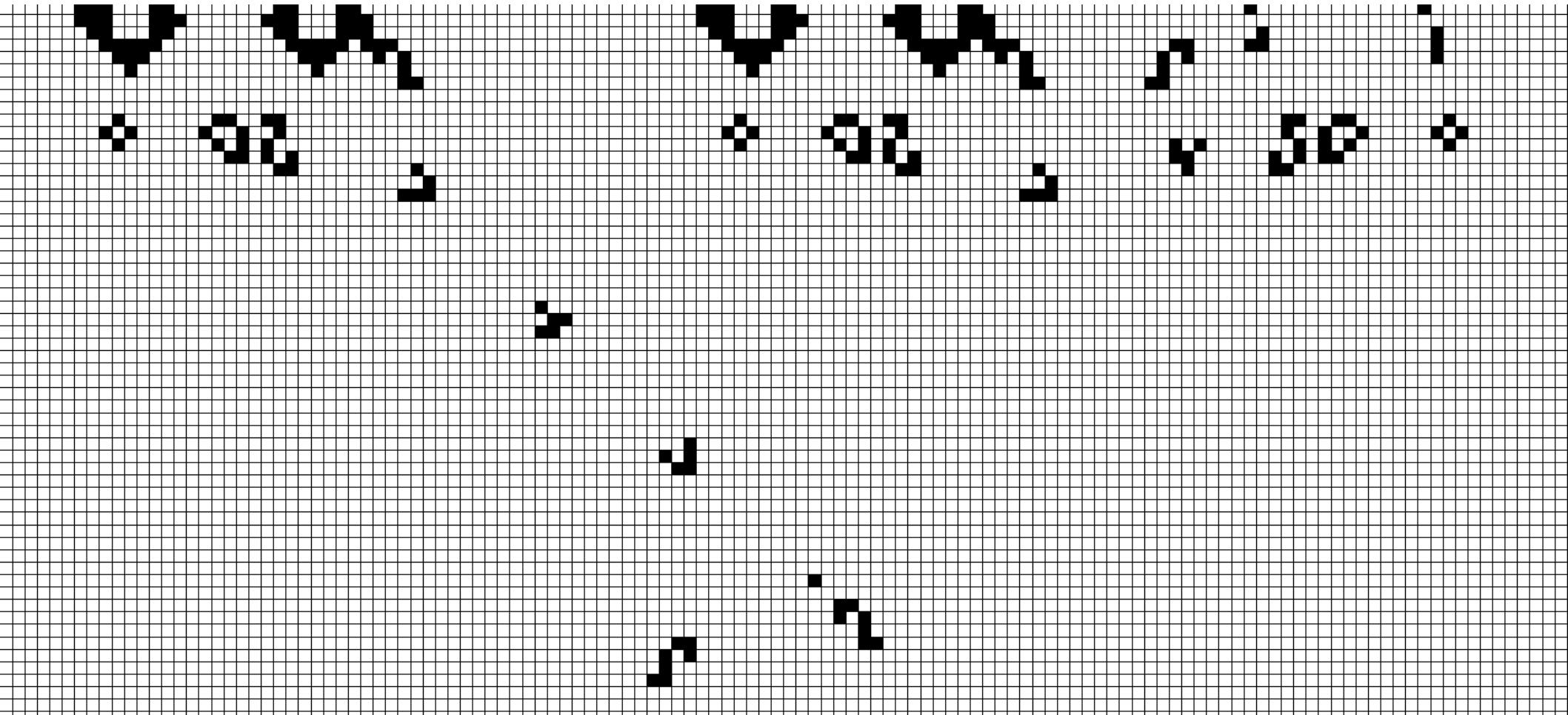
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

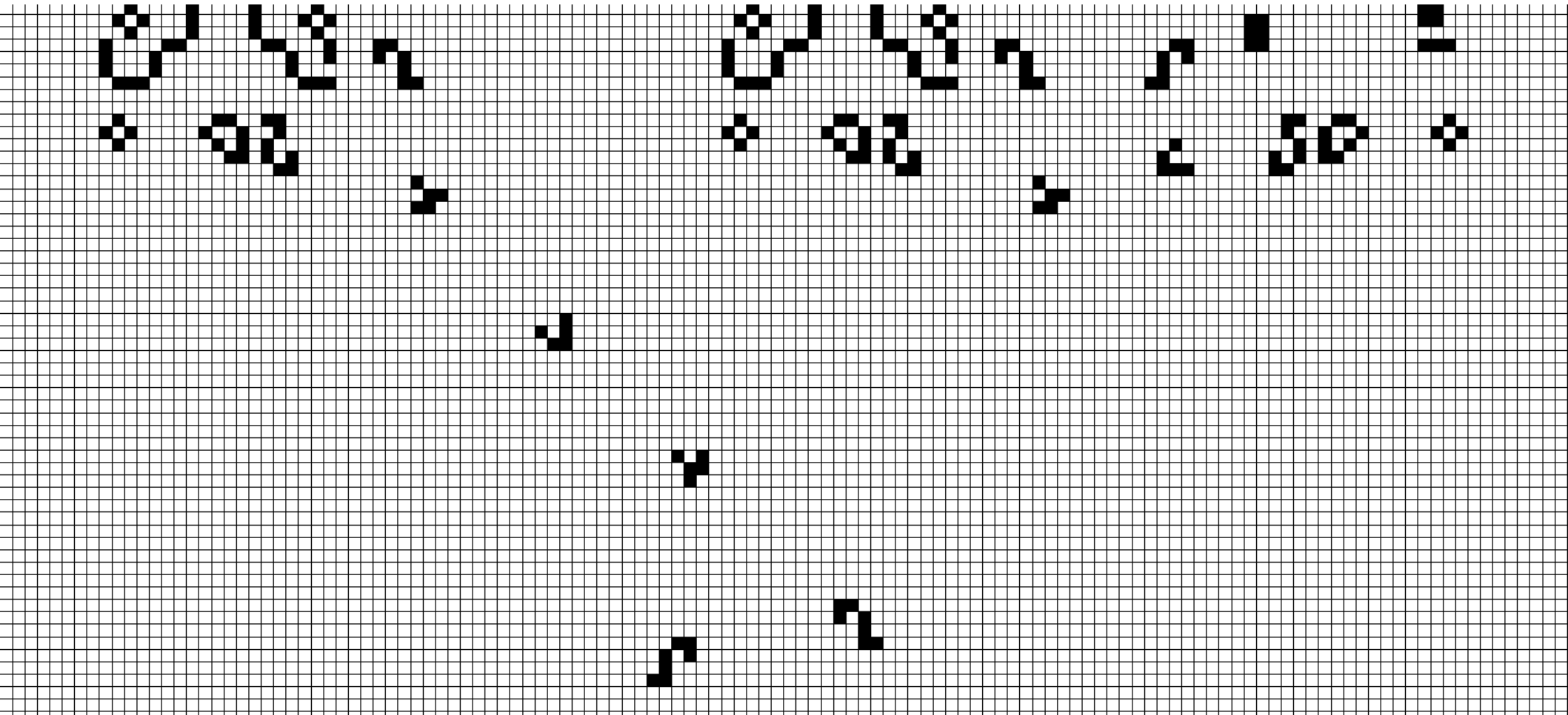
2. Portes logiques



AND : Entrée 1 1

II. Outils nécessaires et composition du filtre

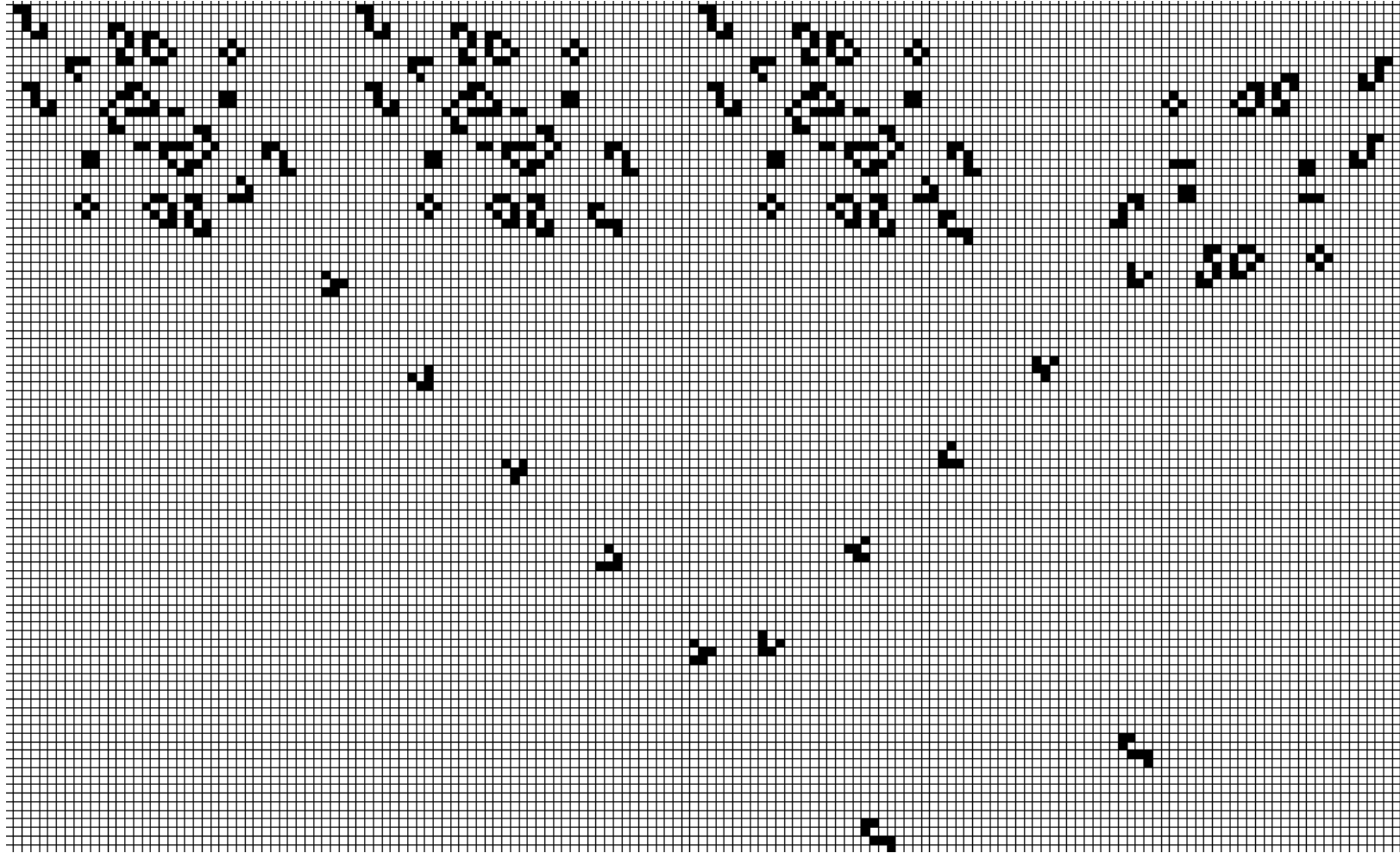
2. Portes logiques



AND : Entrée 1 1

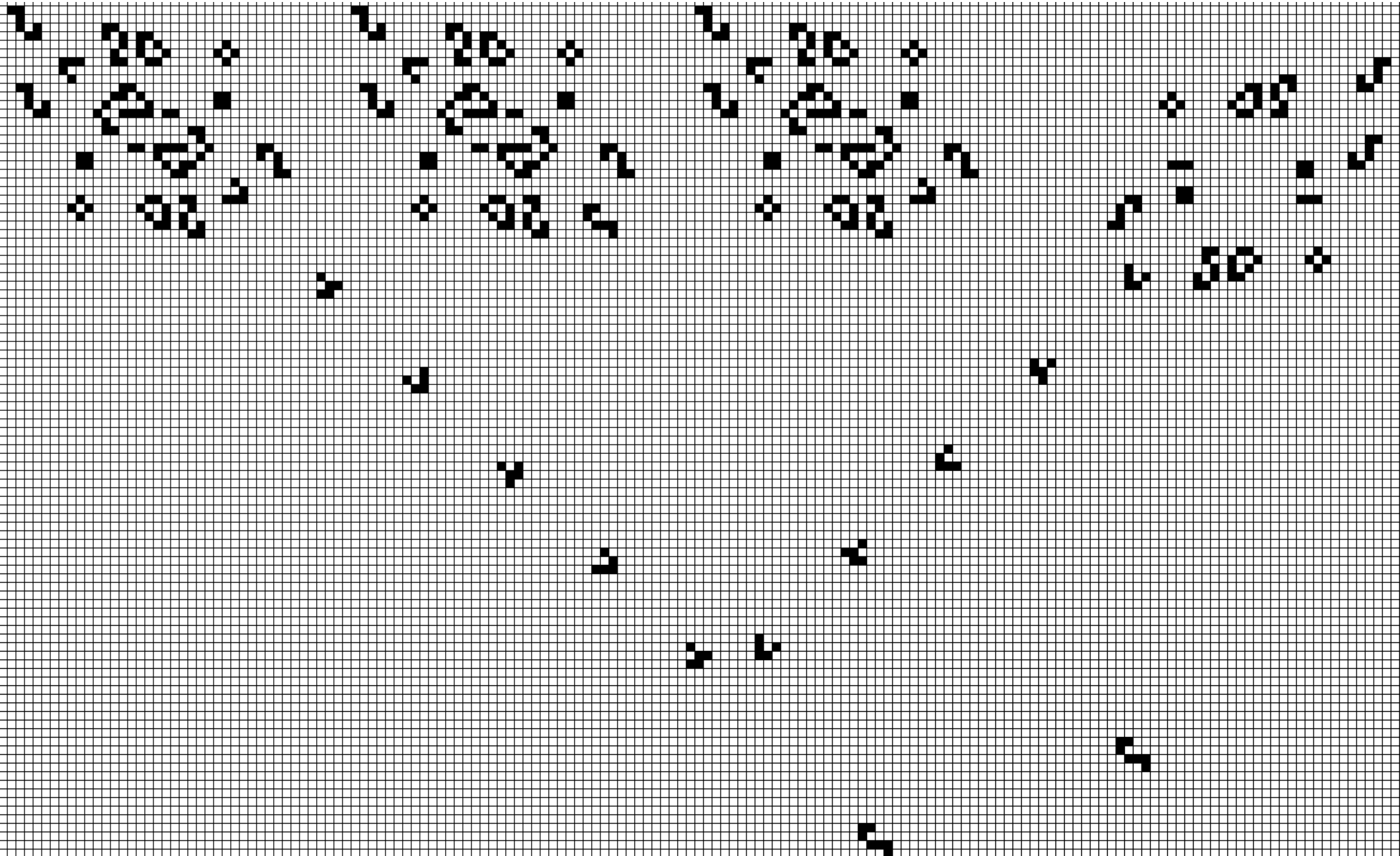
II. Outils nécessaires et composition du filtre

2. Portes logiques



OR : Entrée 0 0

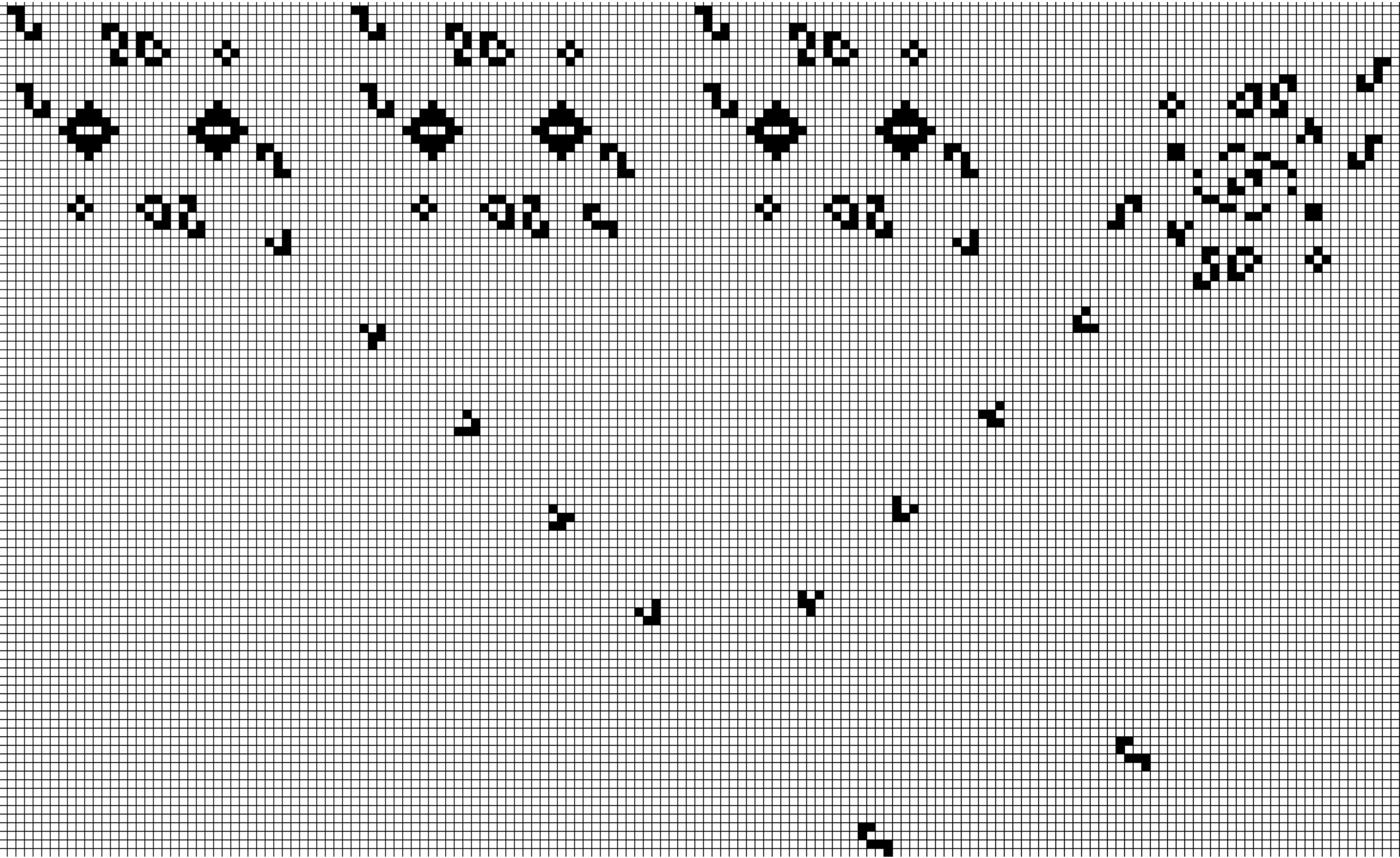
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

250/414

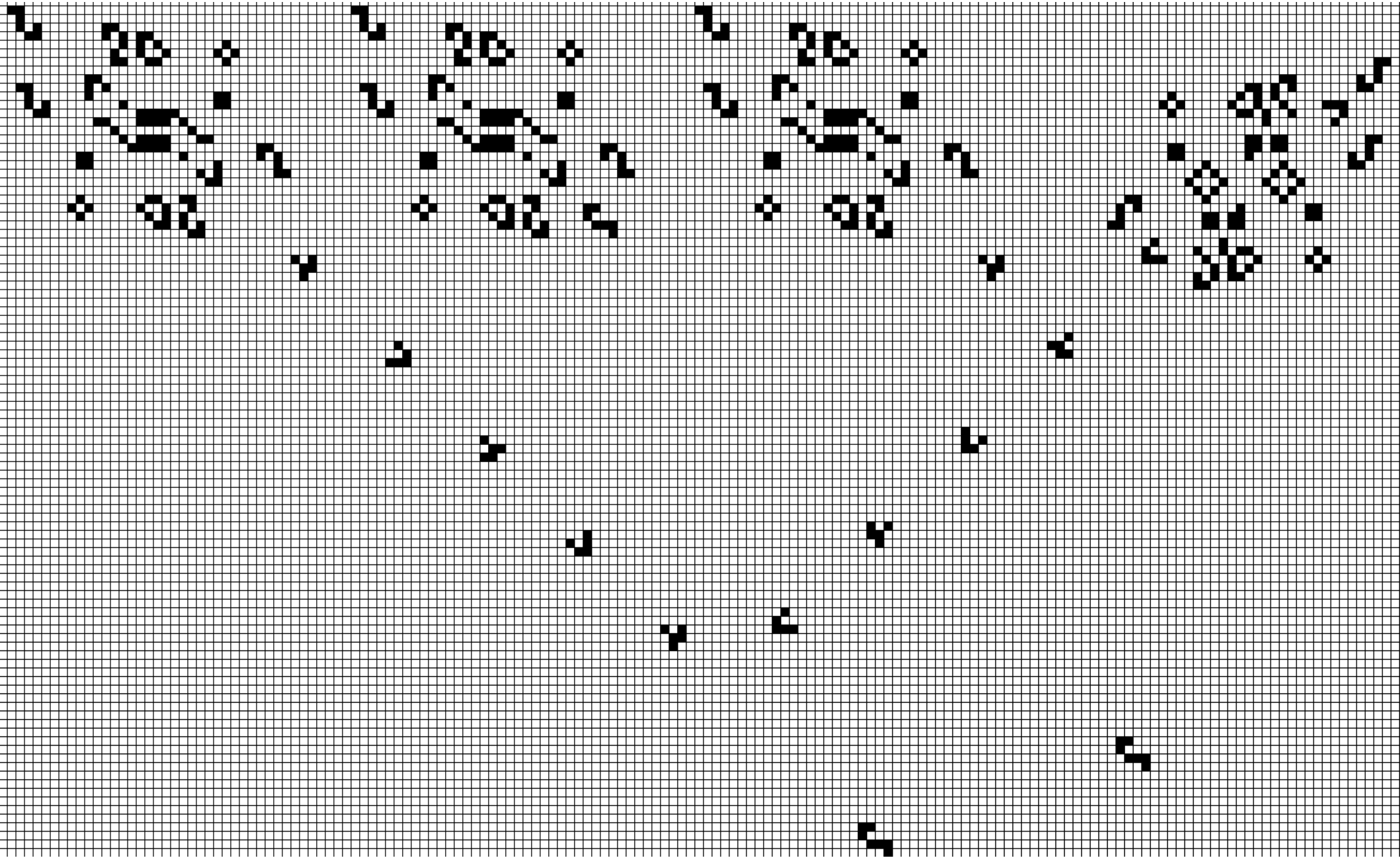
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

251/414

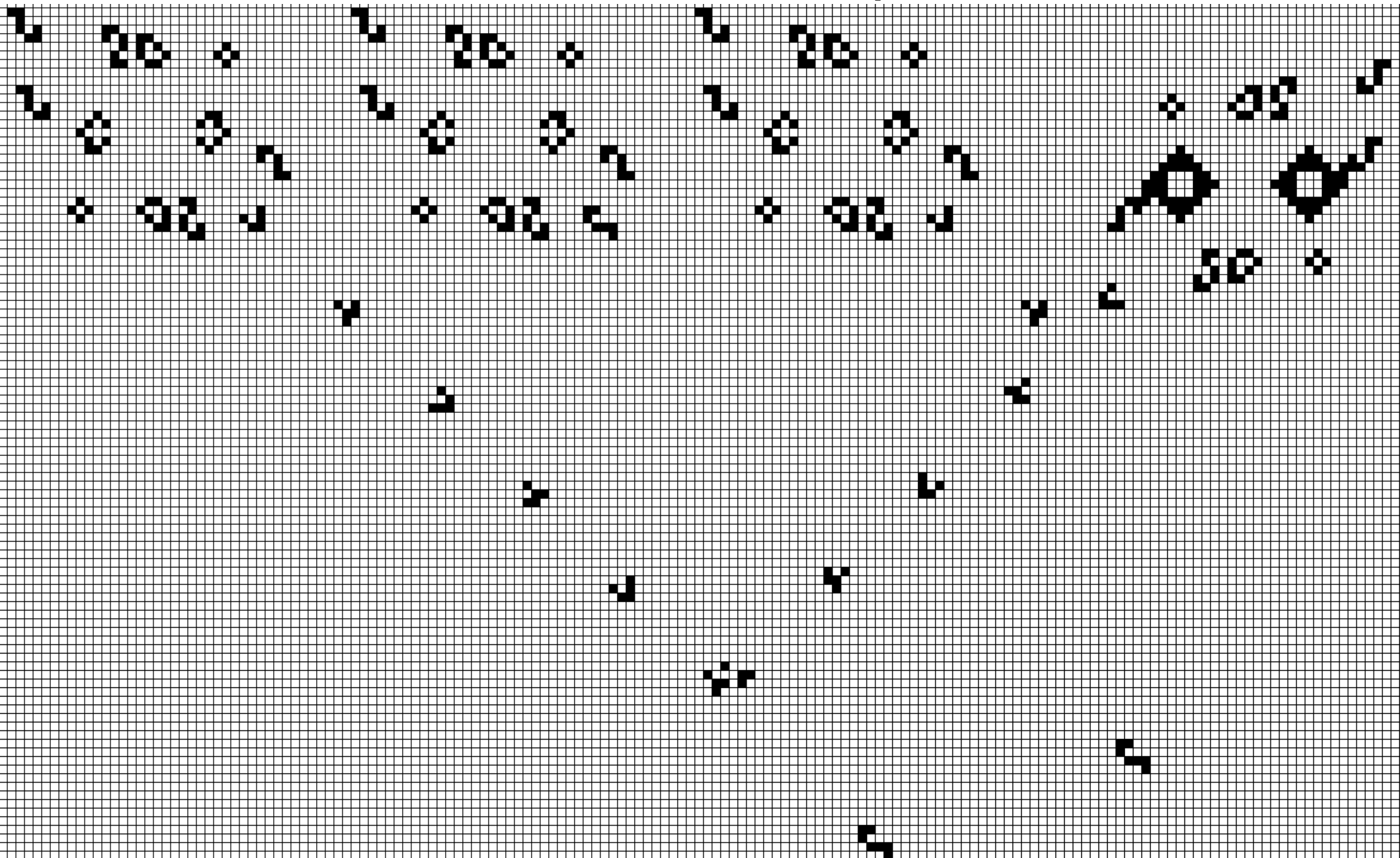
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

252/414

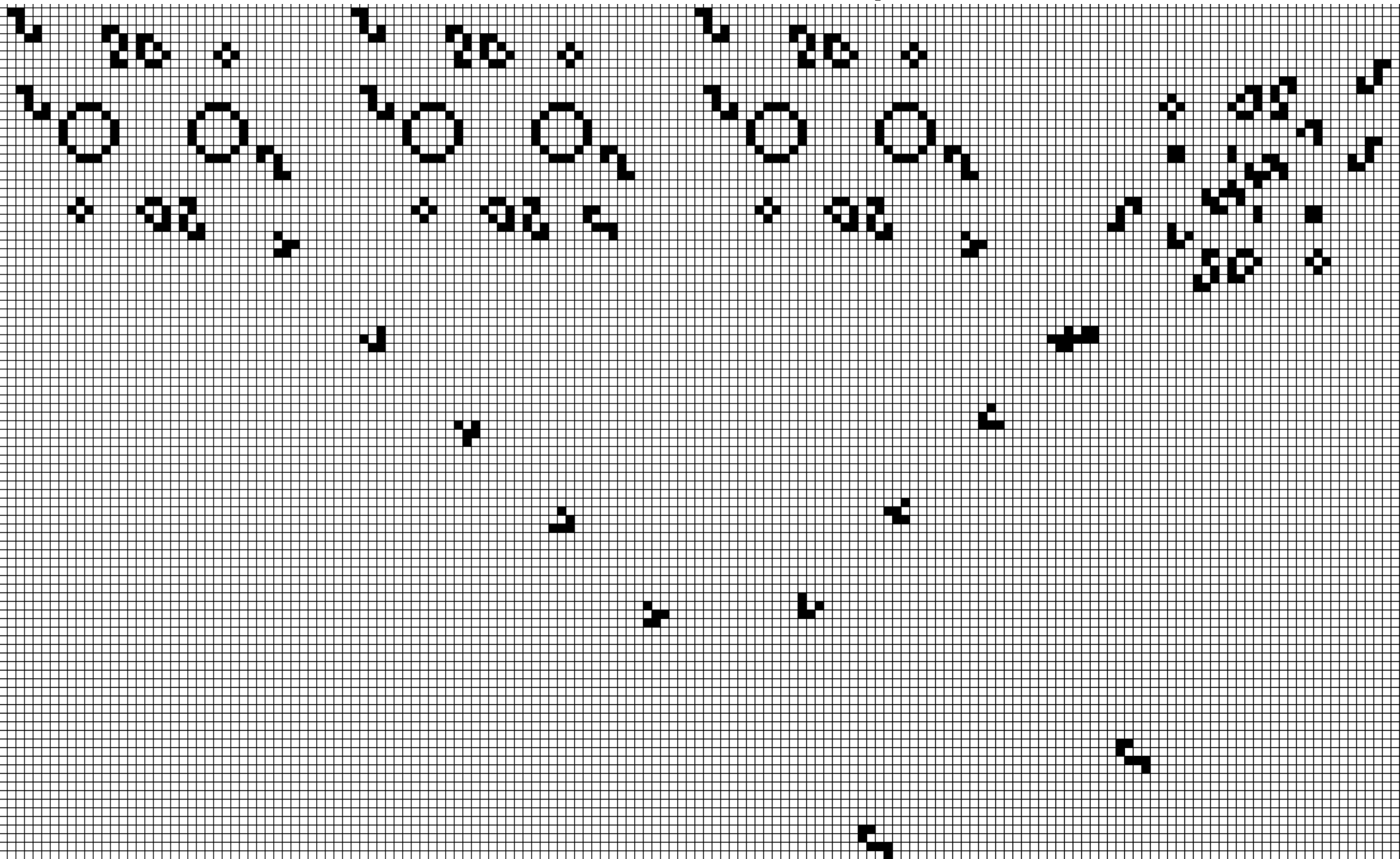
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

253/414

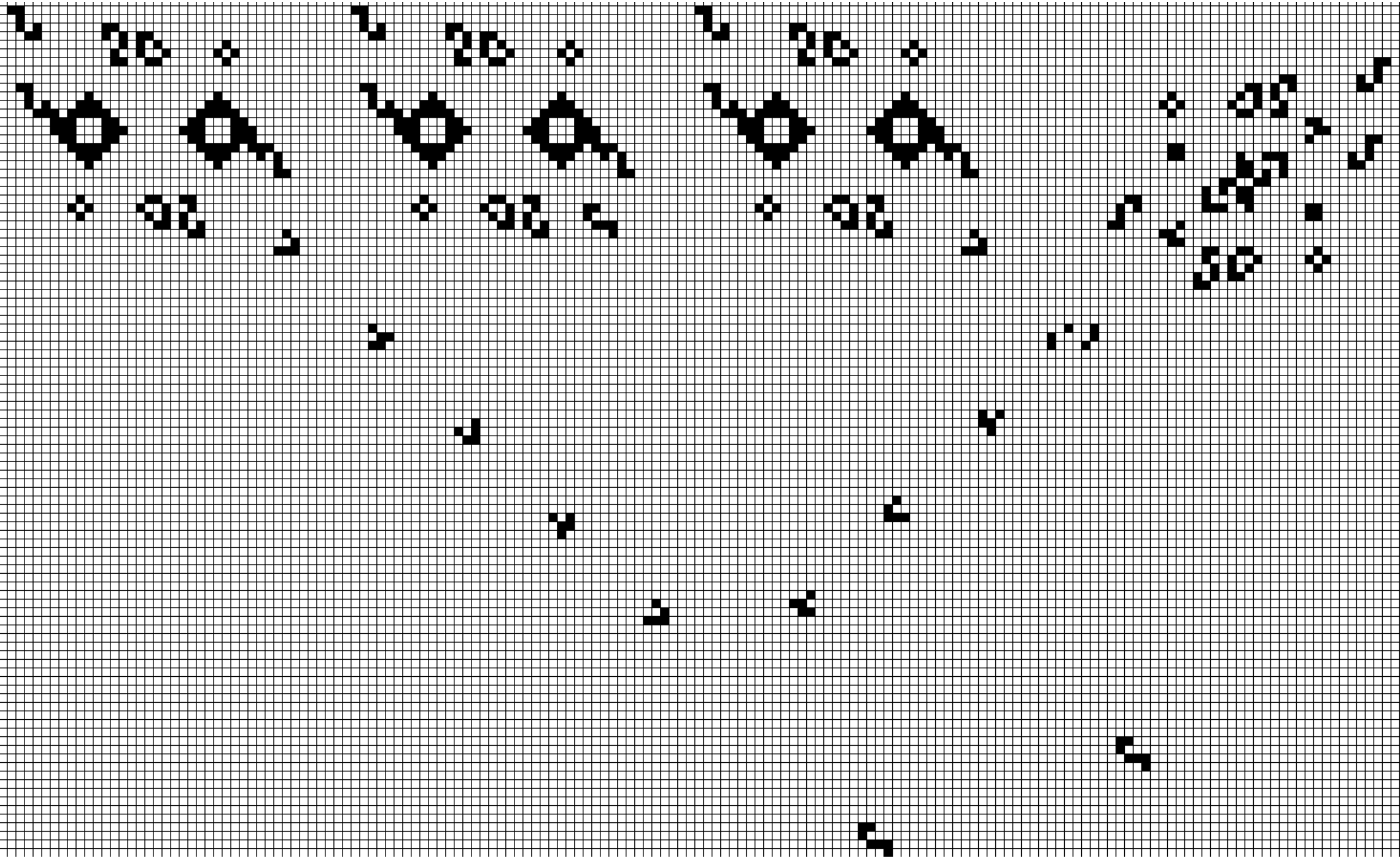
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

254/414

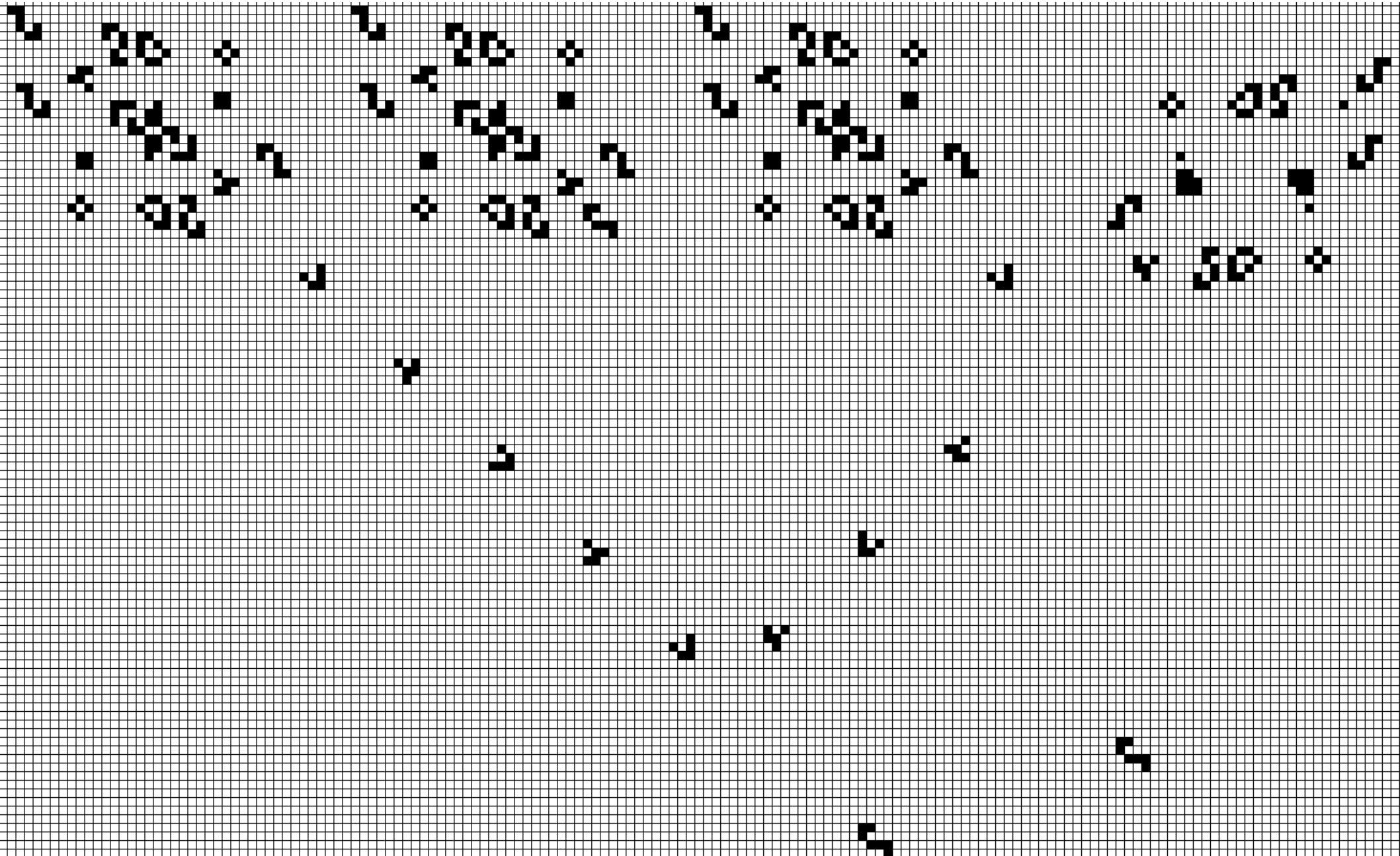
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

255/414

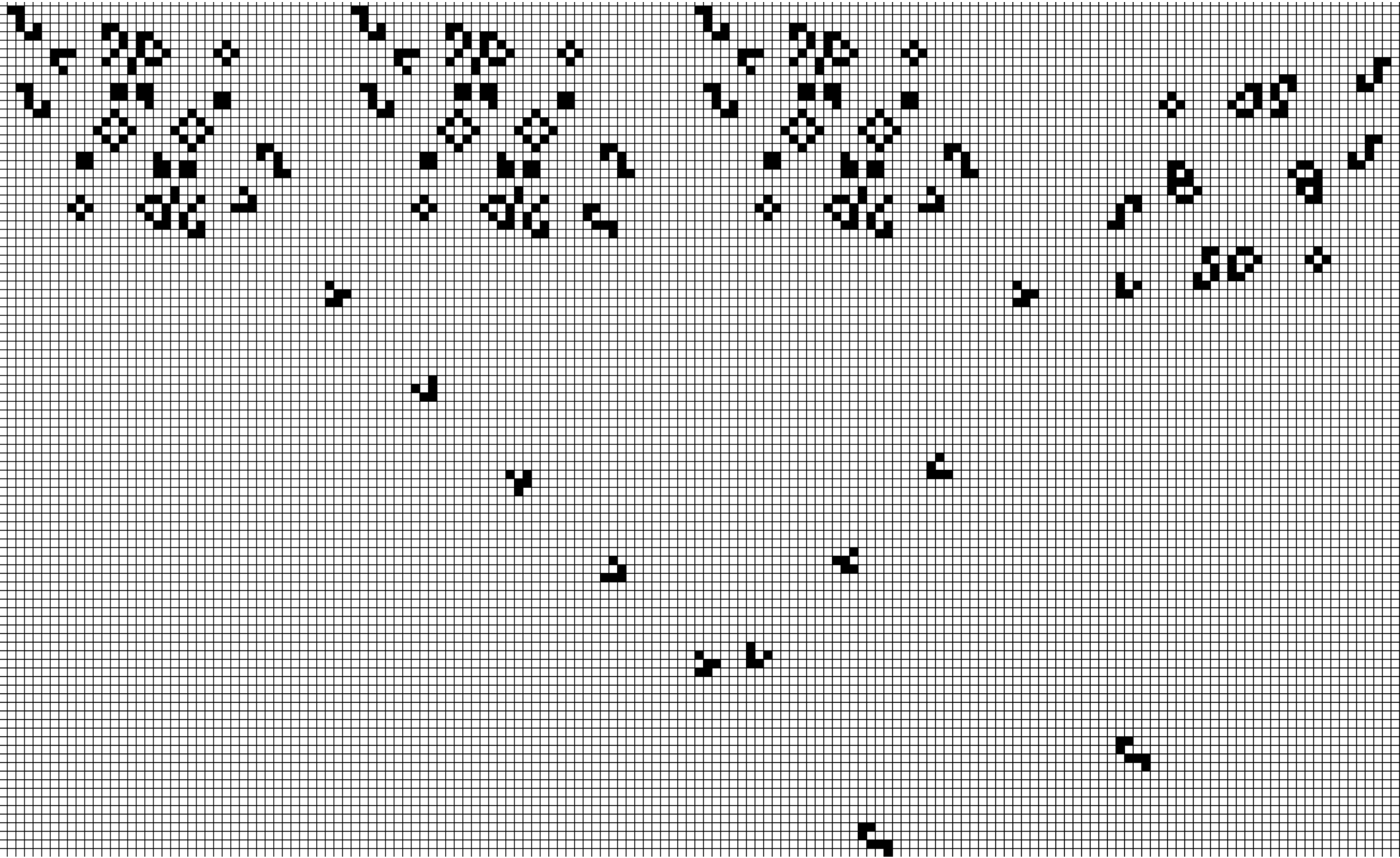
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

256/414

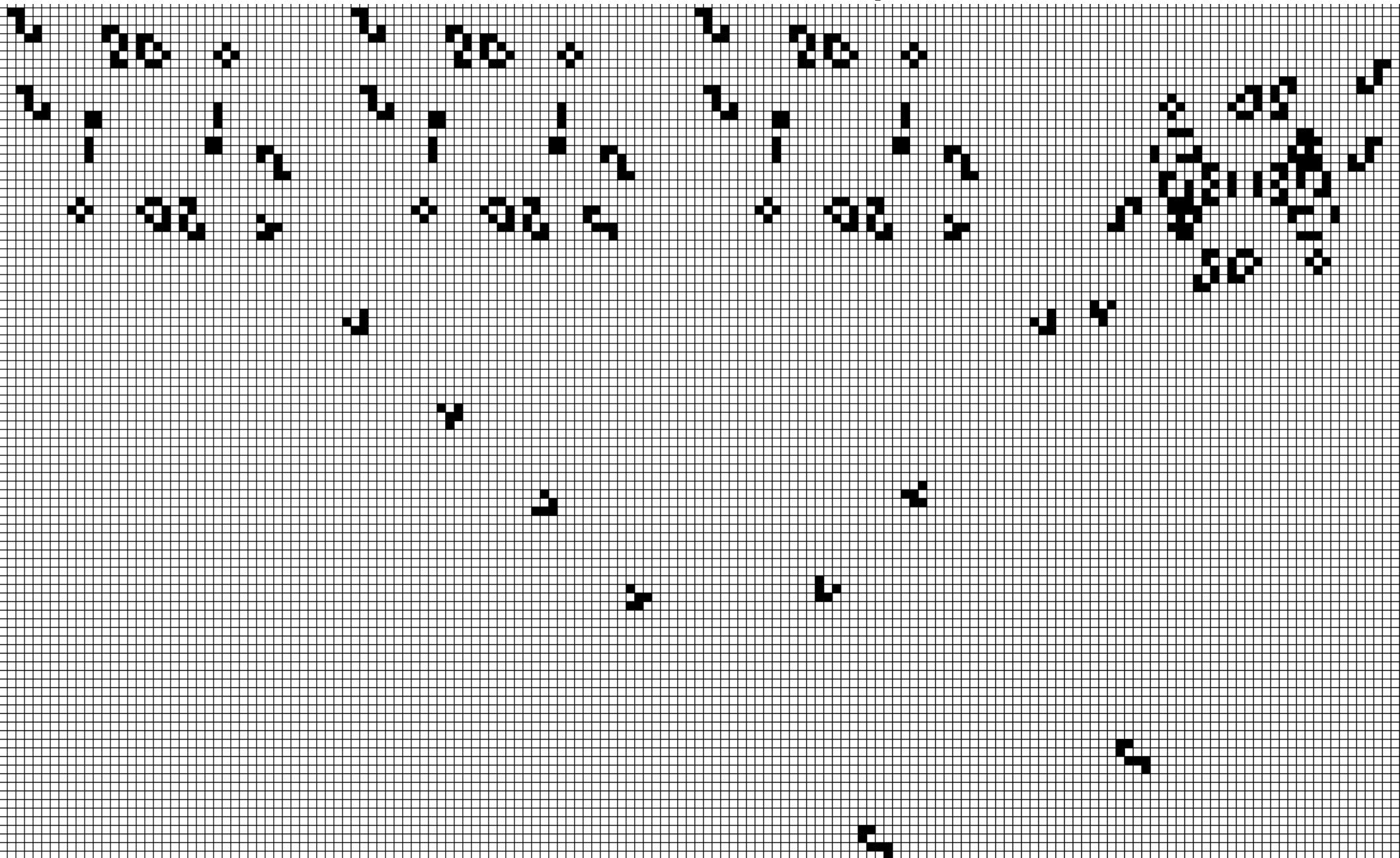
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

257/414

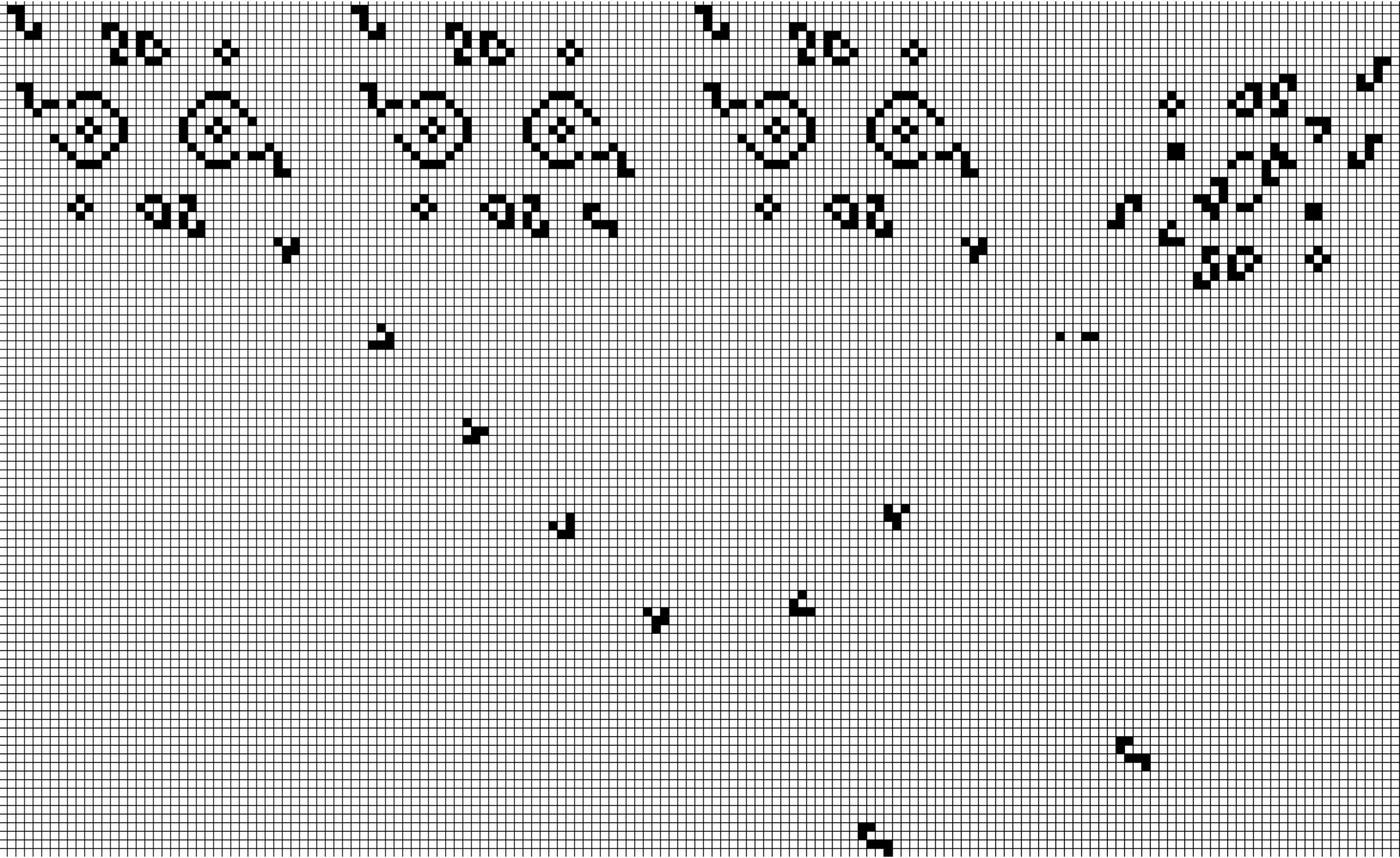
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

258/414

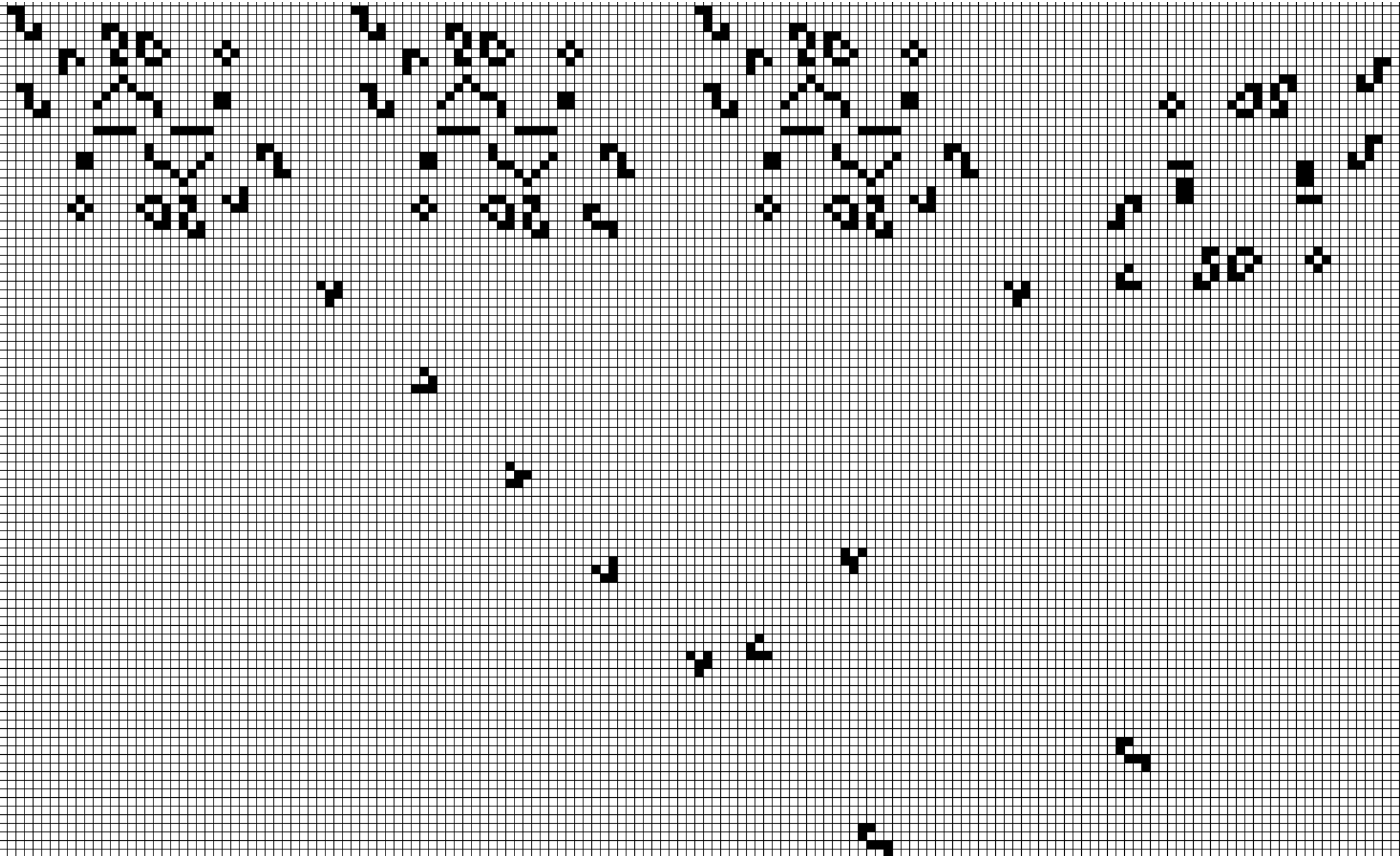
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

259/414

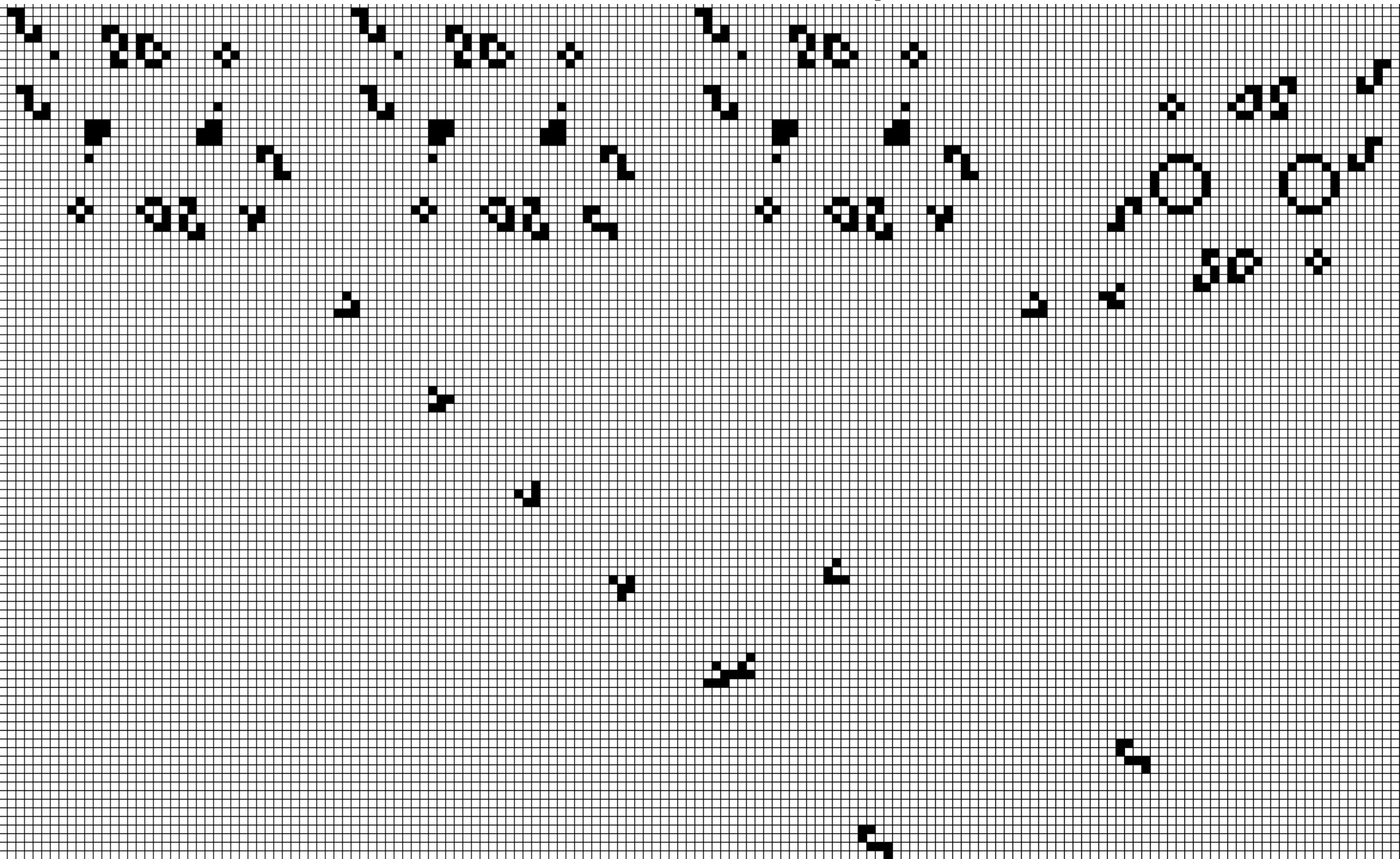
II. Outils nécessaires et composition du filtre



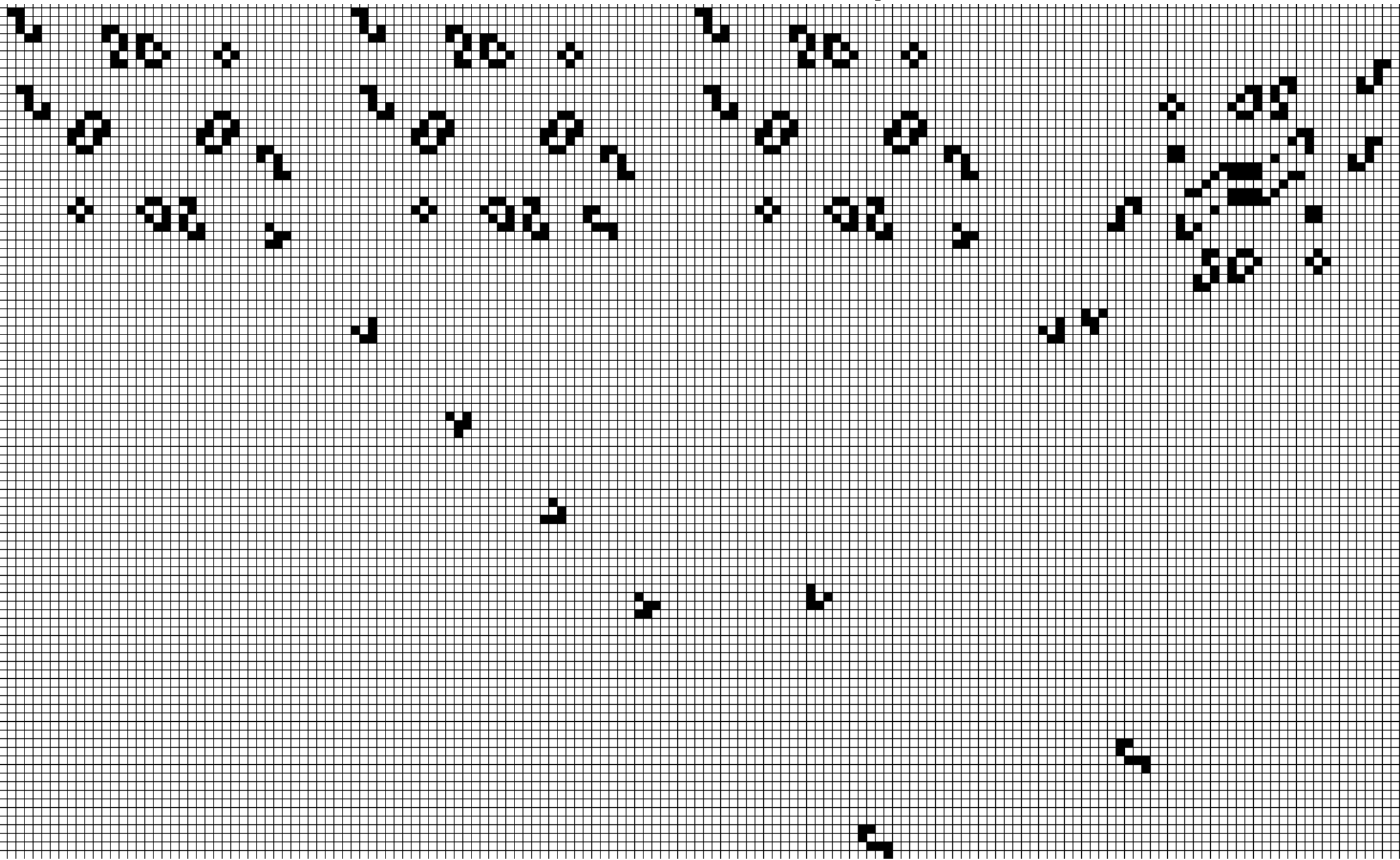
OR : Entrée 0 1

260/414

II. Outils nécessaires et composition du filtre



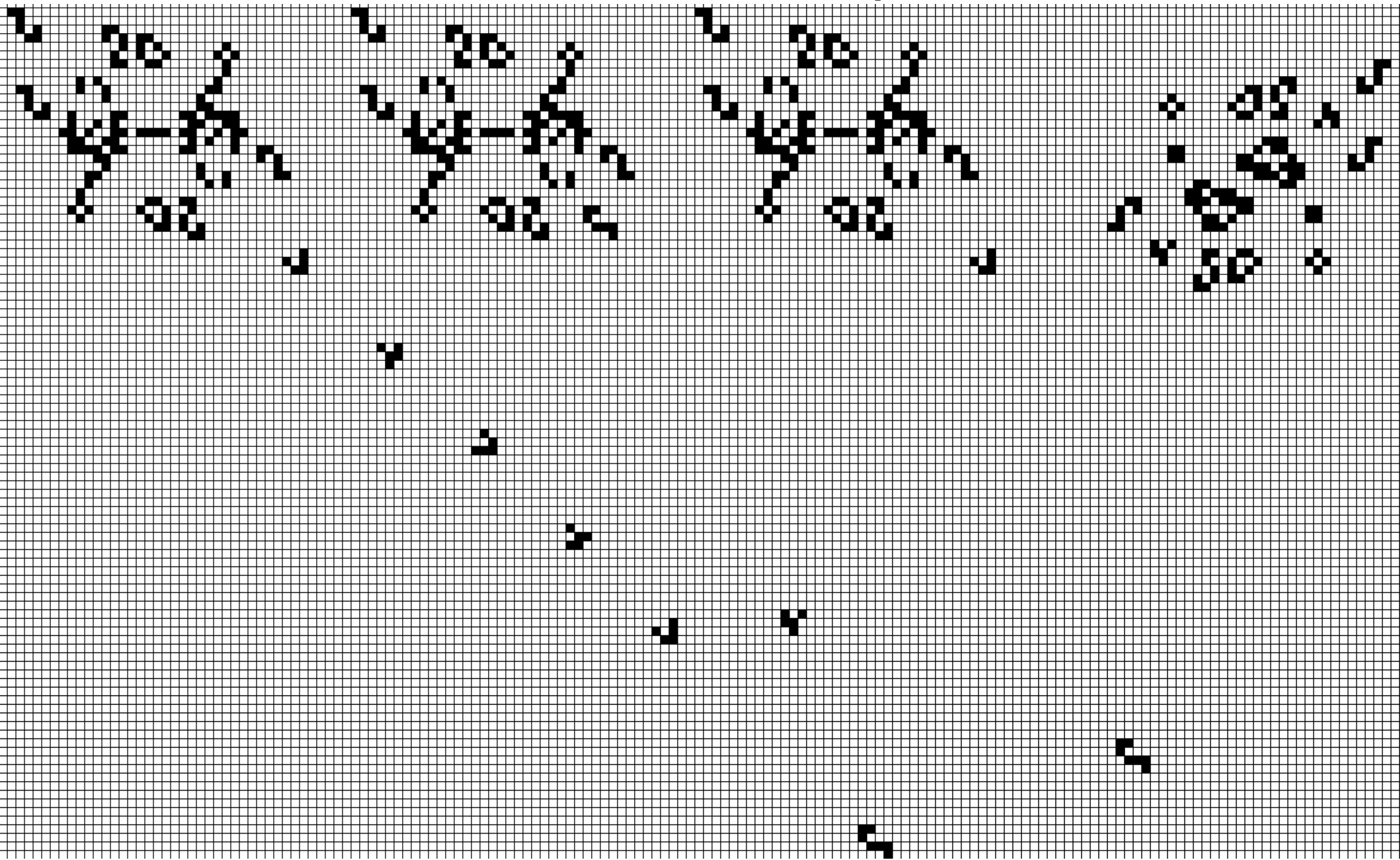
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

262/414

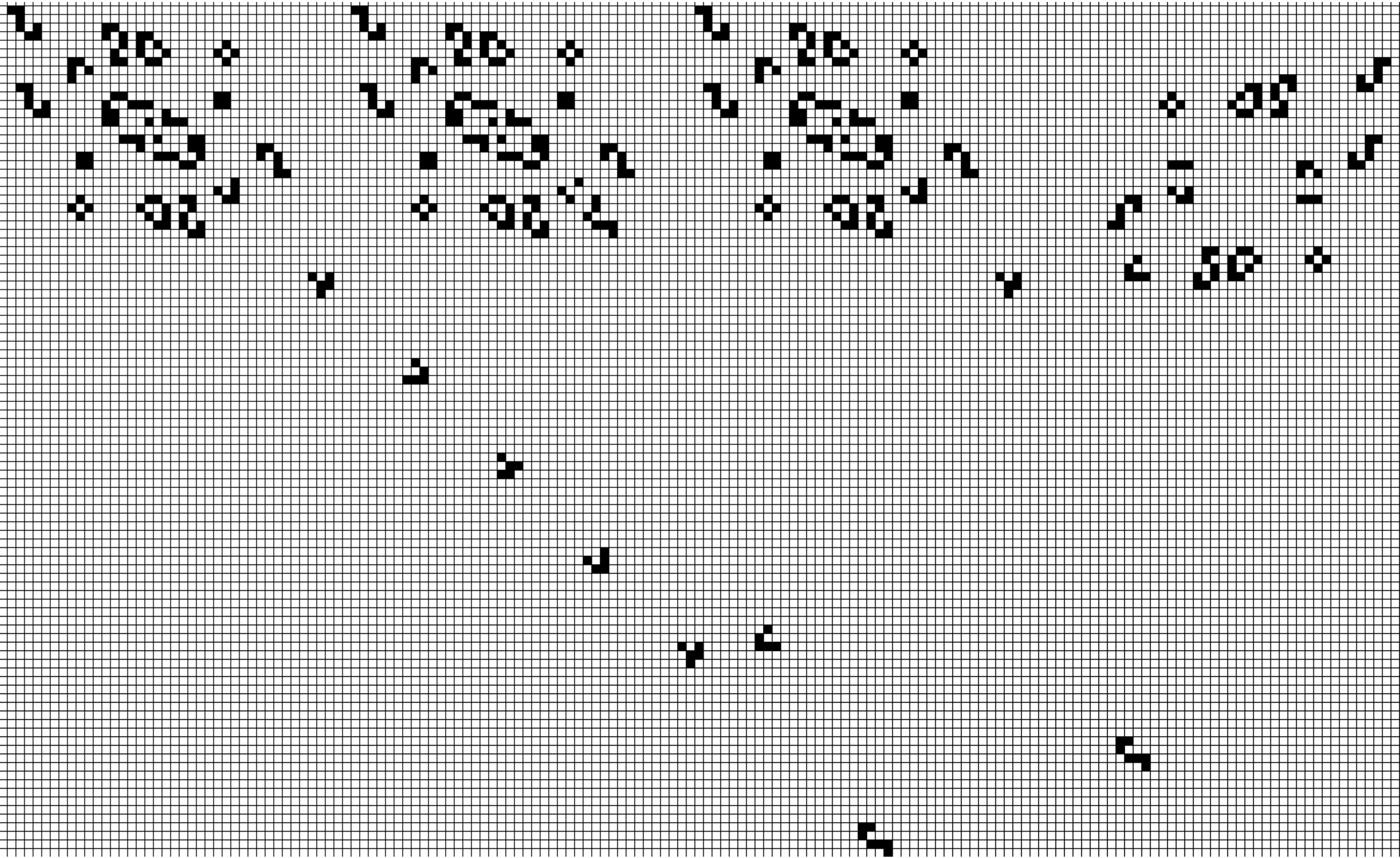
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

263/414

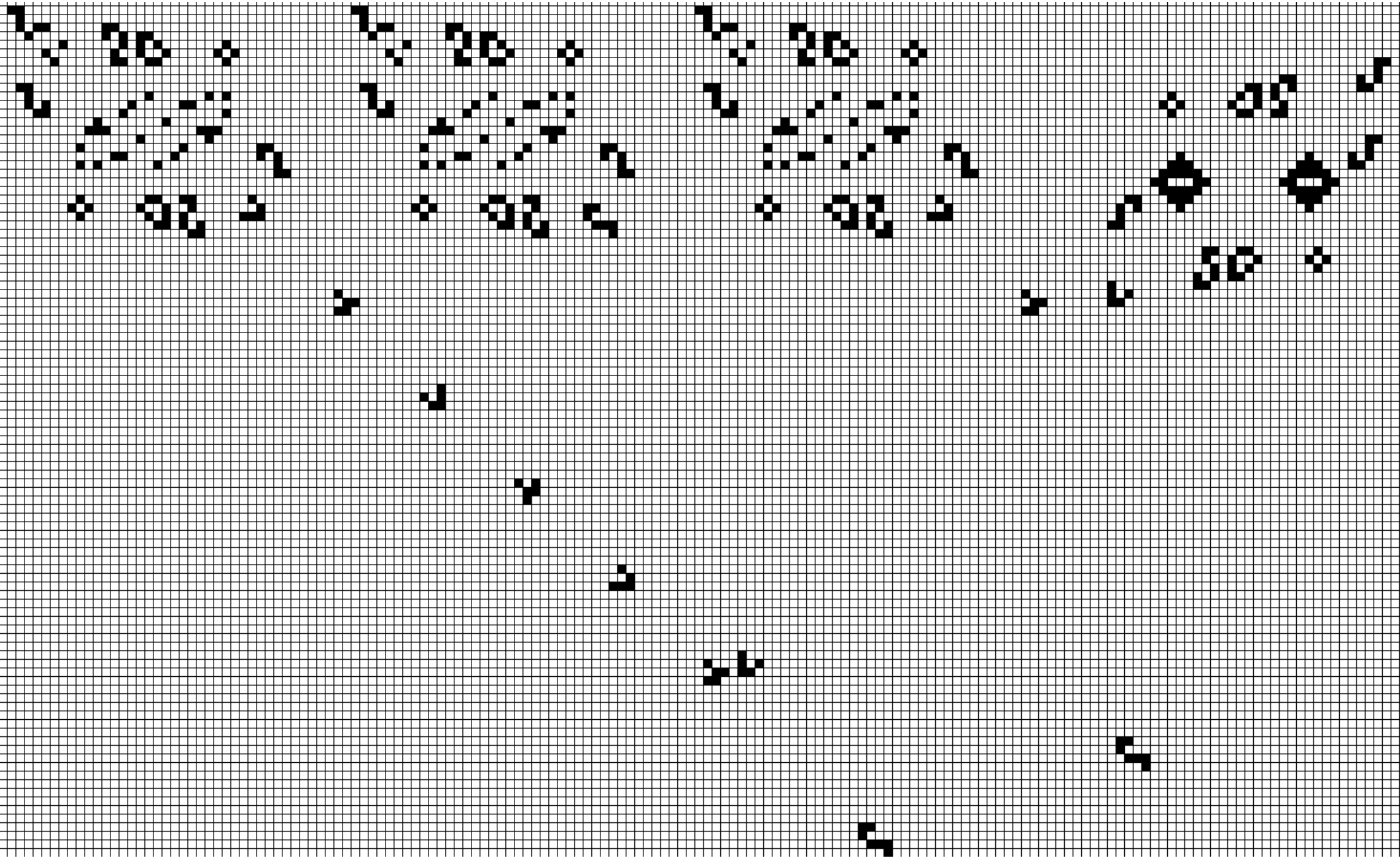
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

264/414

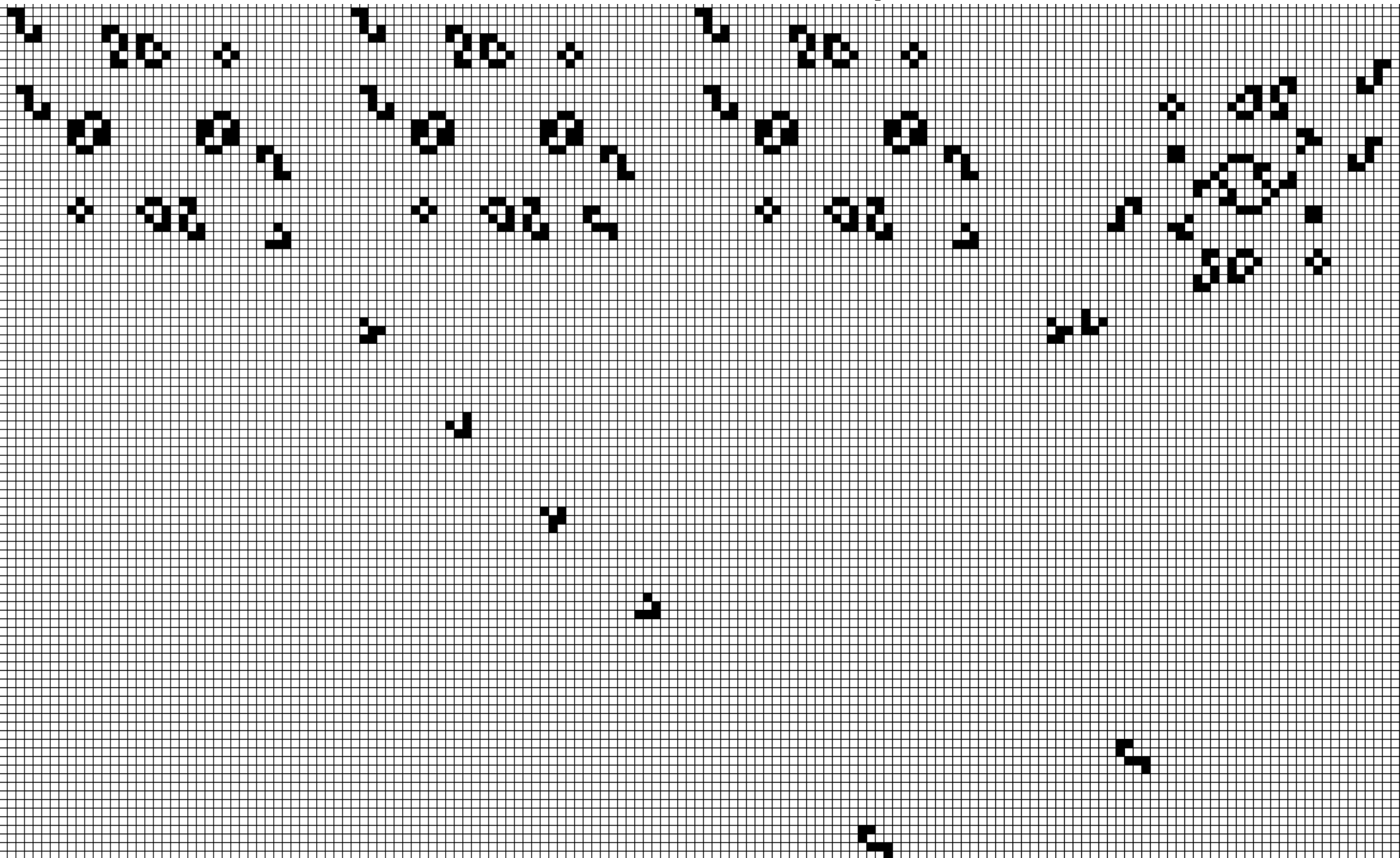
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

265/414

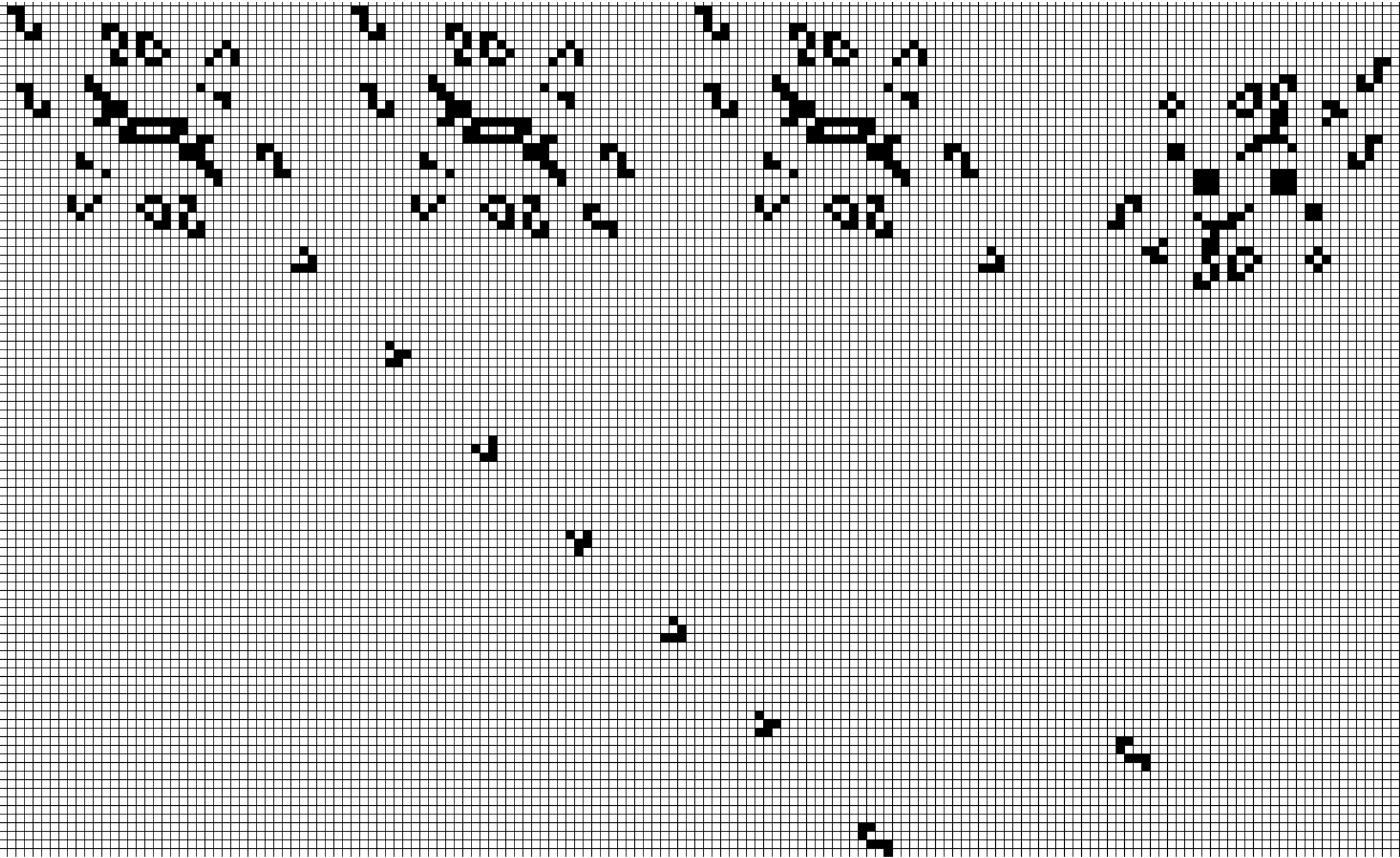
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

266/414

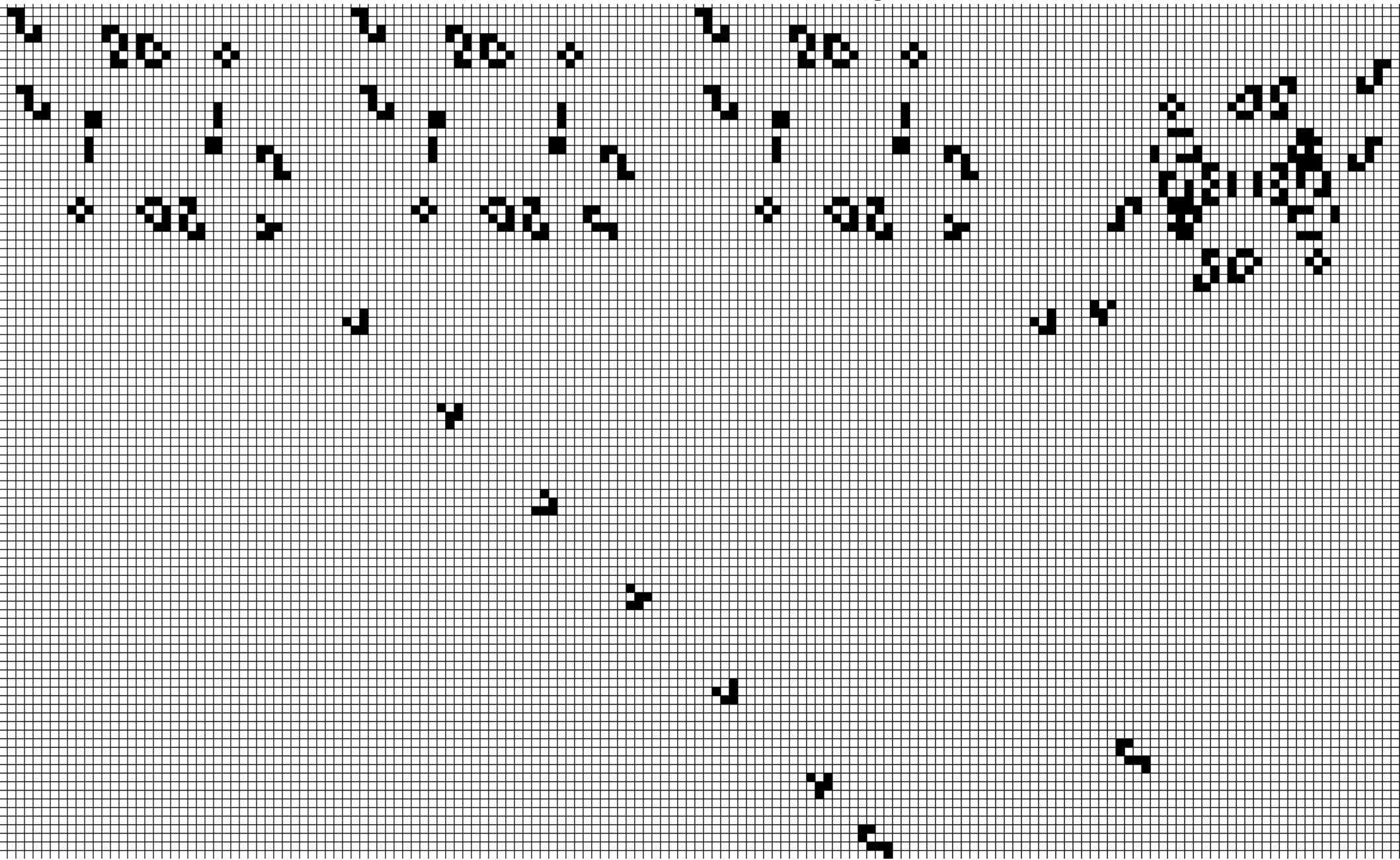
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

267/414

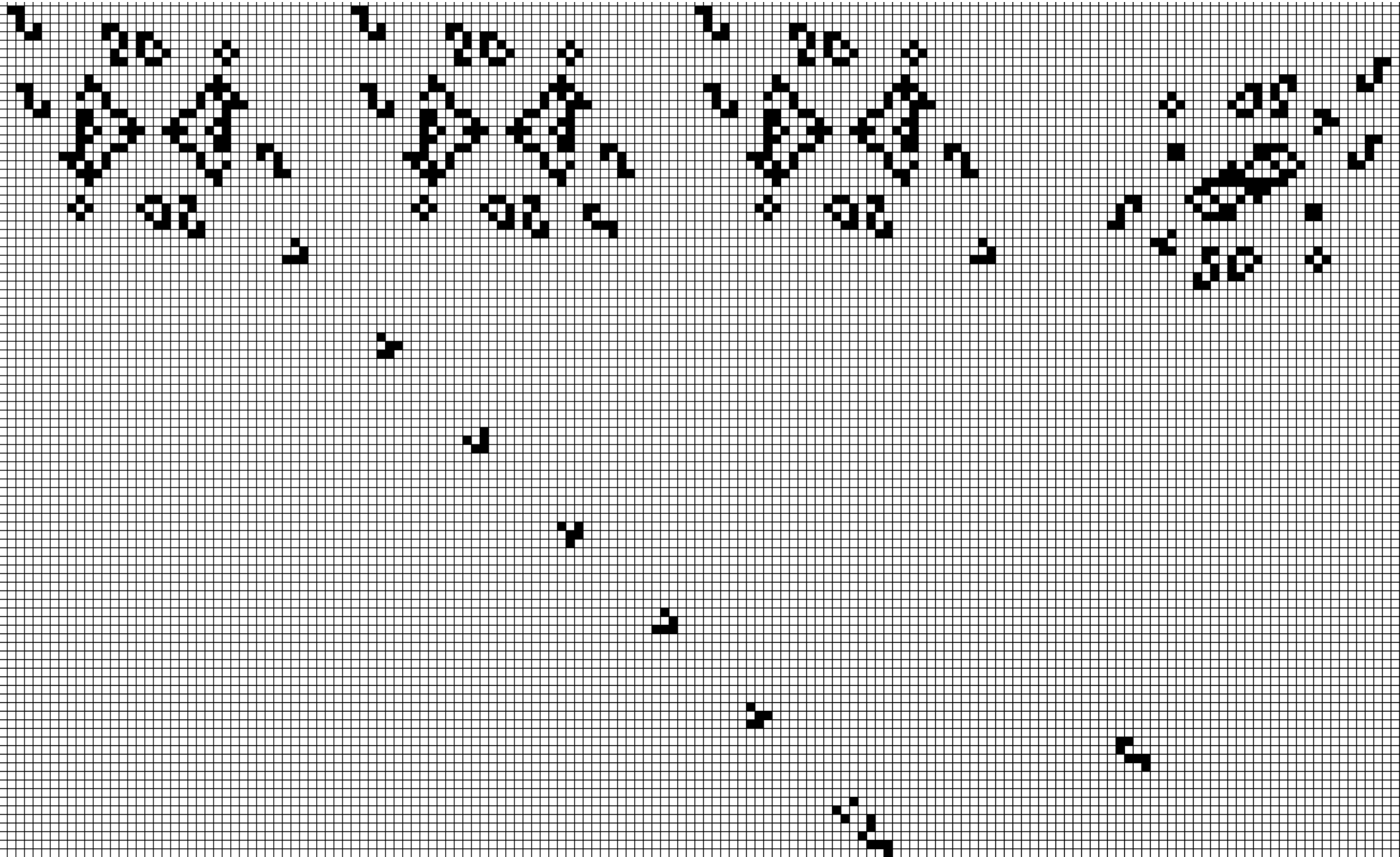
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

268/414

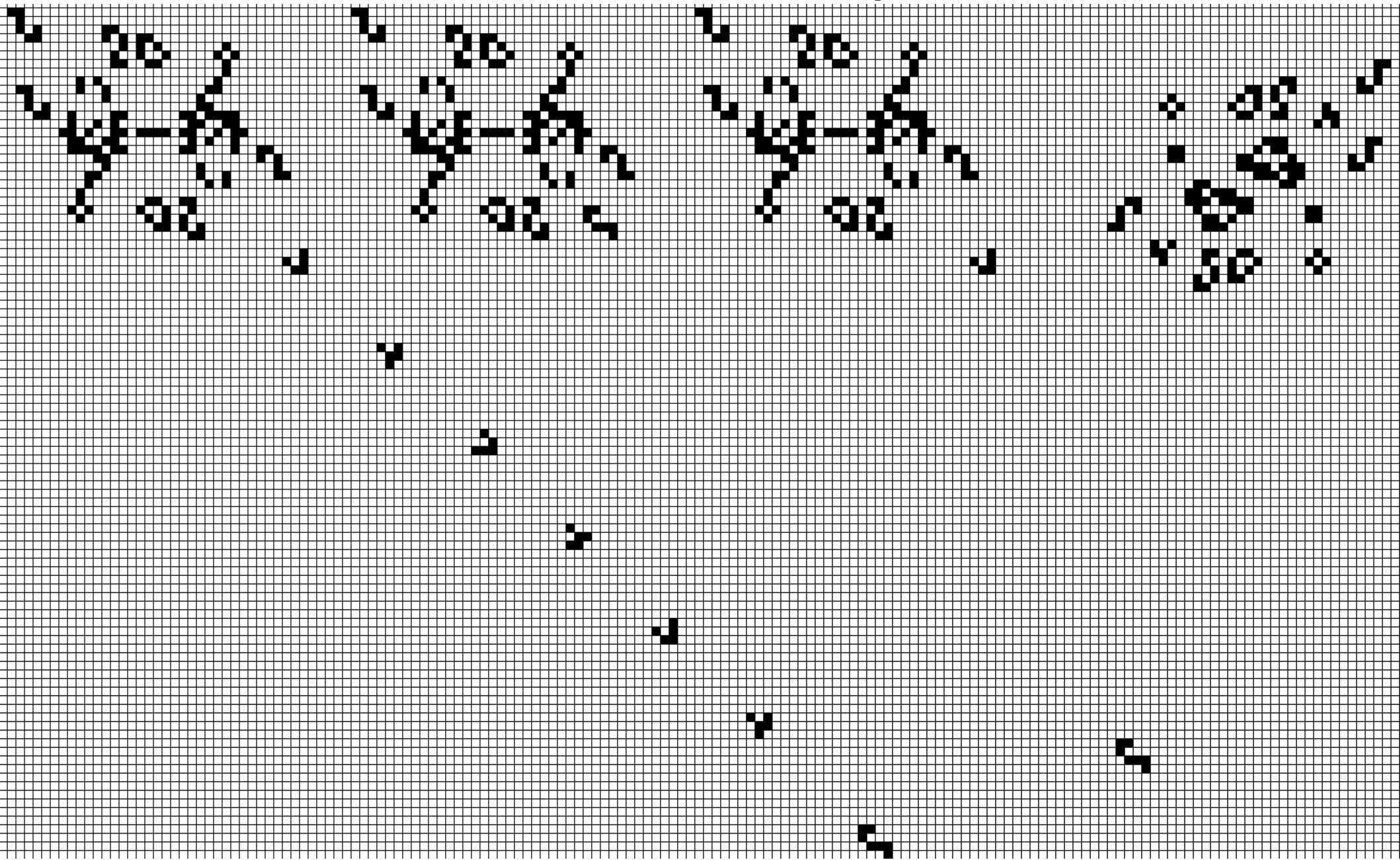
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

269/414

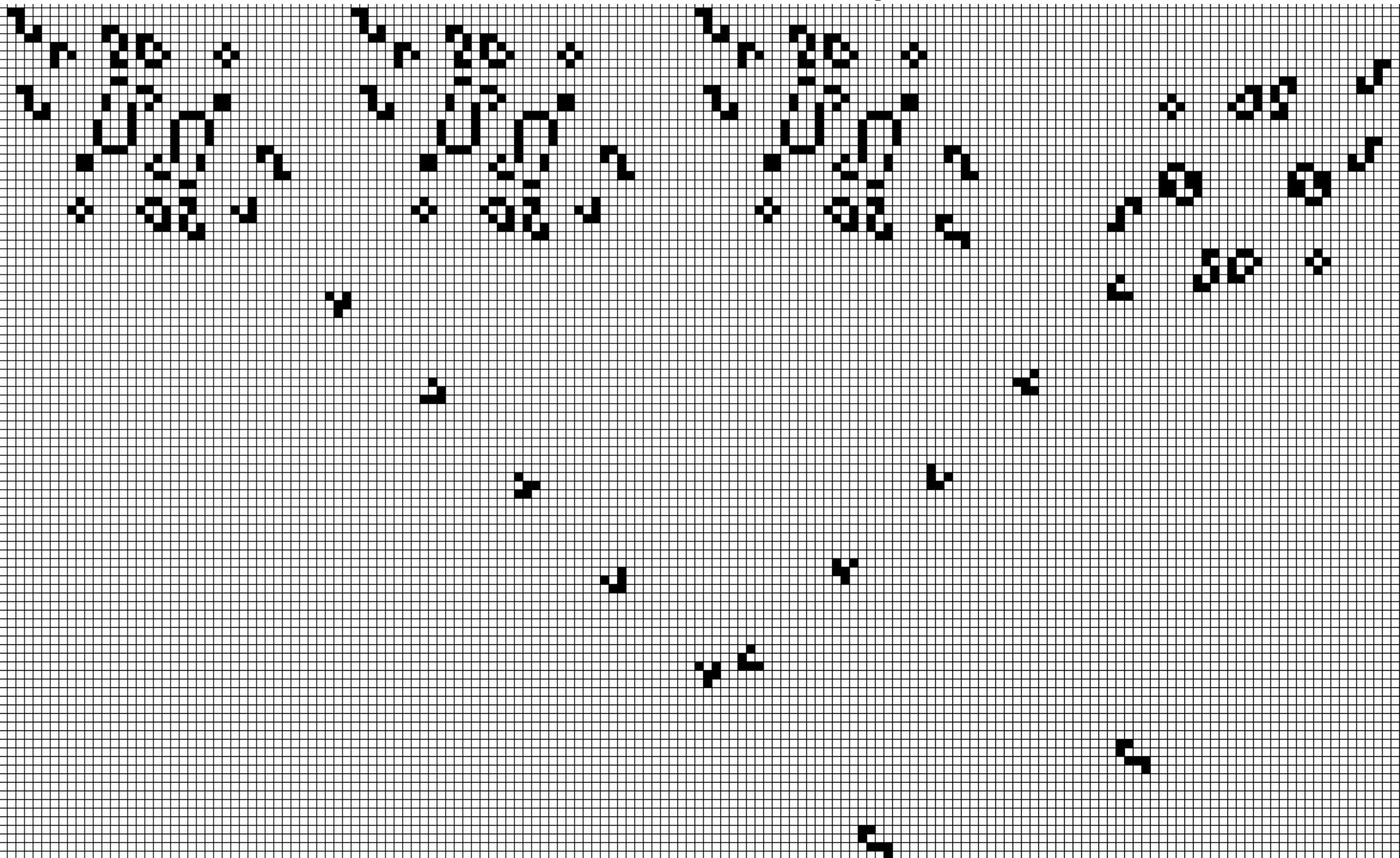
II. Outils nécessaires et composition du filtre



OR : Entrée 0 1

270/414

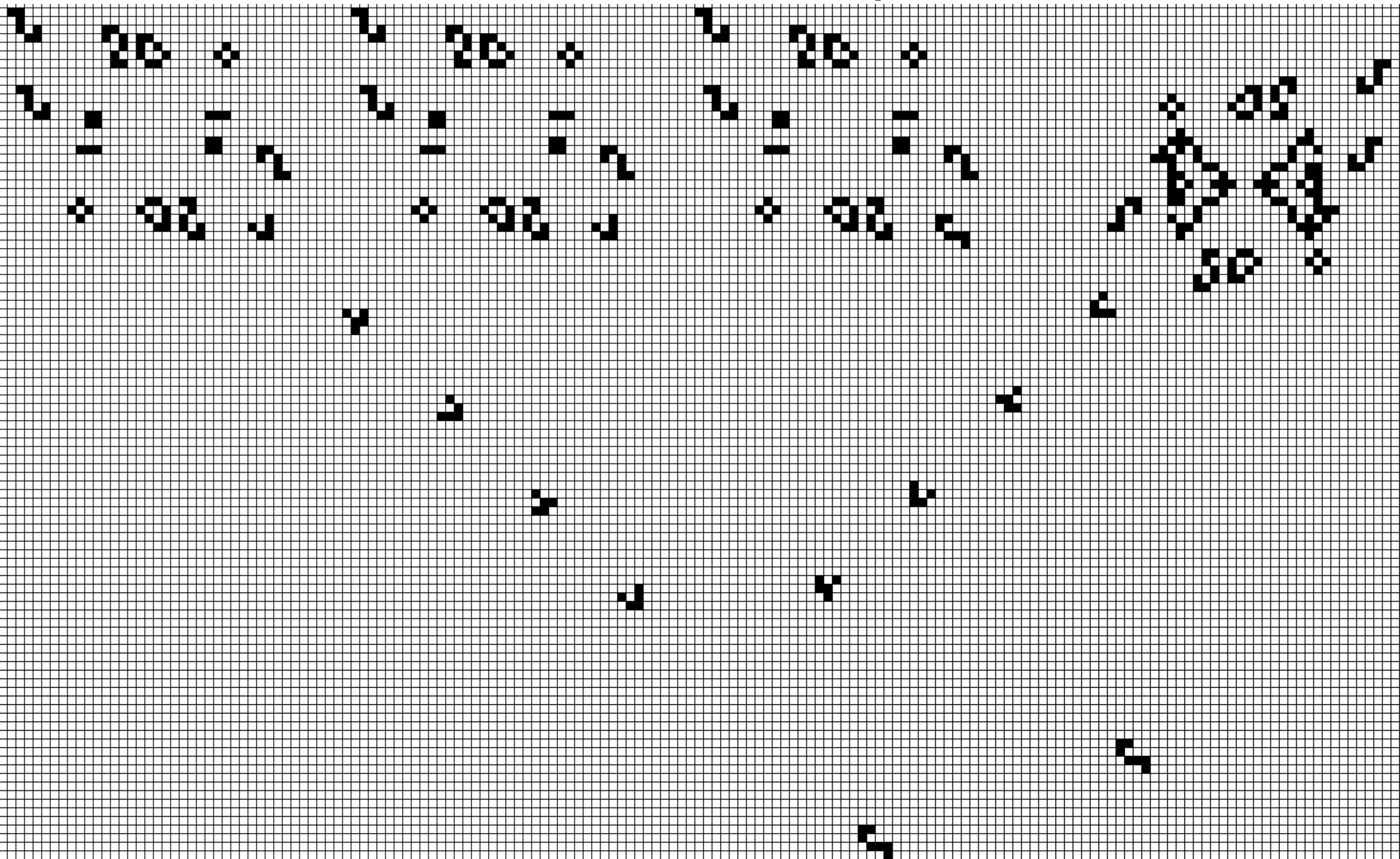
II. Outils nécessaires et composition du filtre



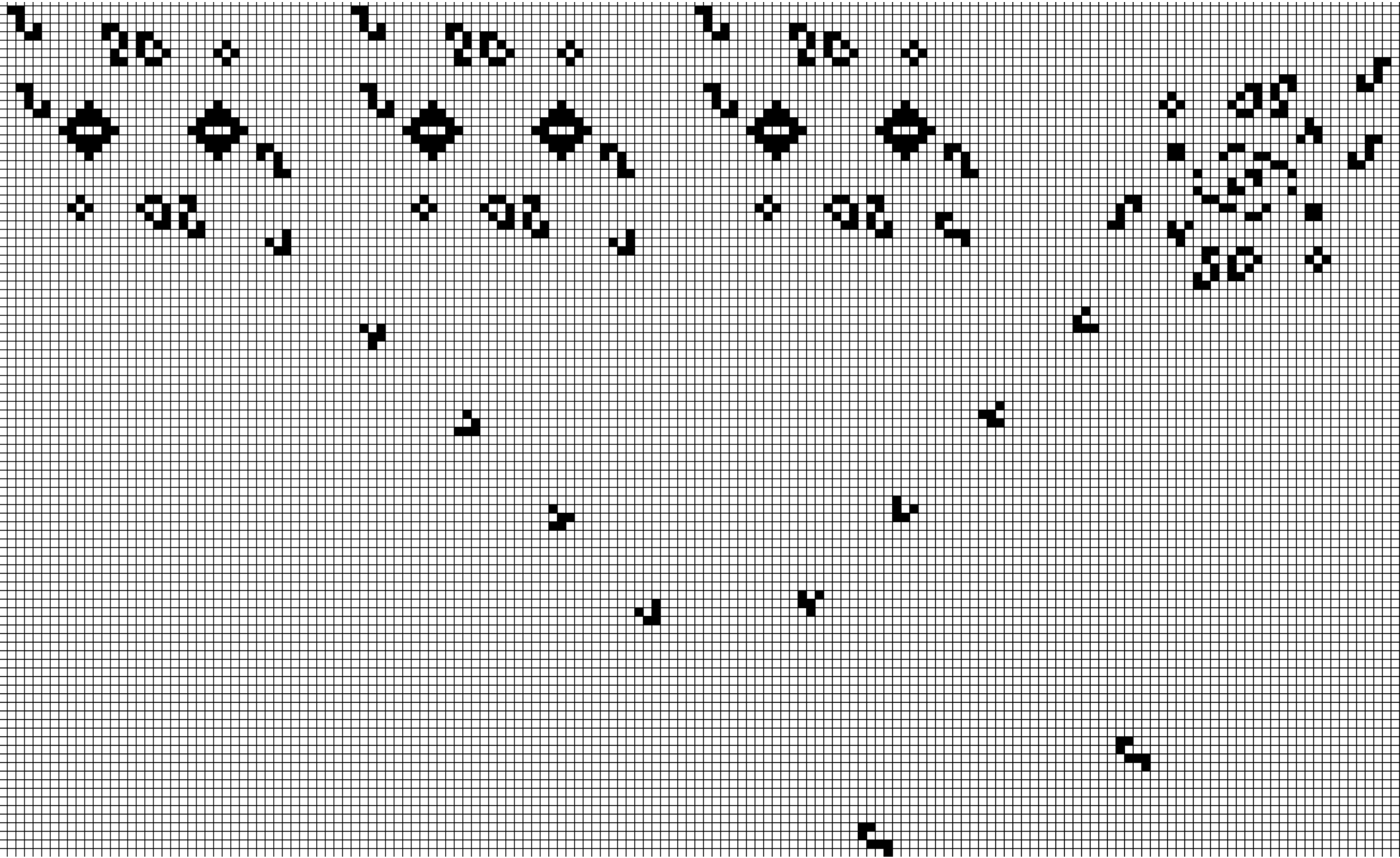
OR : Entrée 1 0

271/414

II. Outils nécessaires et composition du filtre



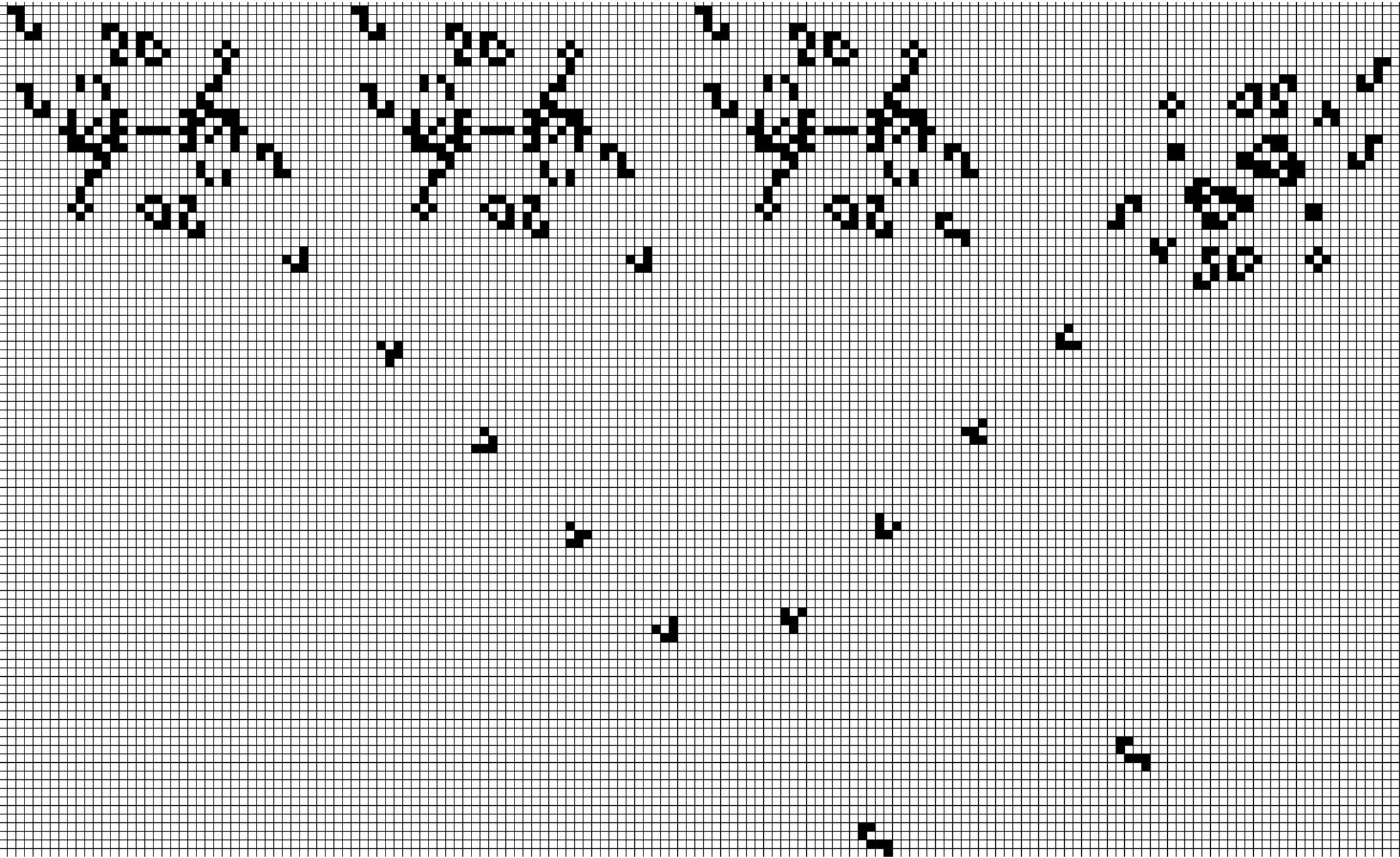
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

273/414

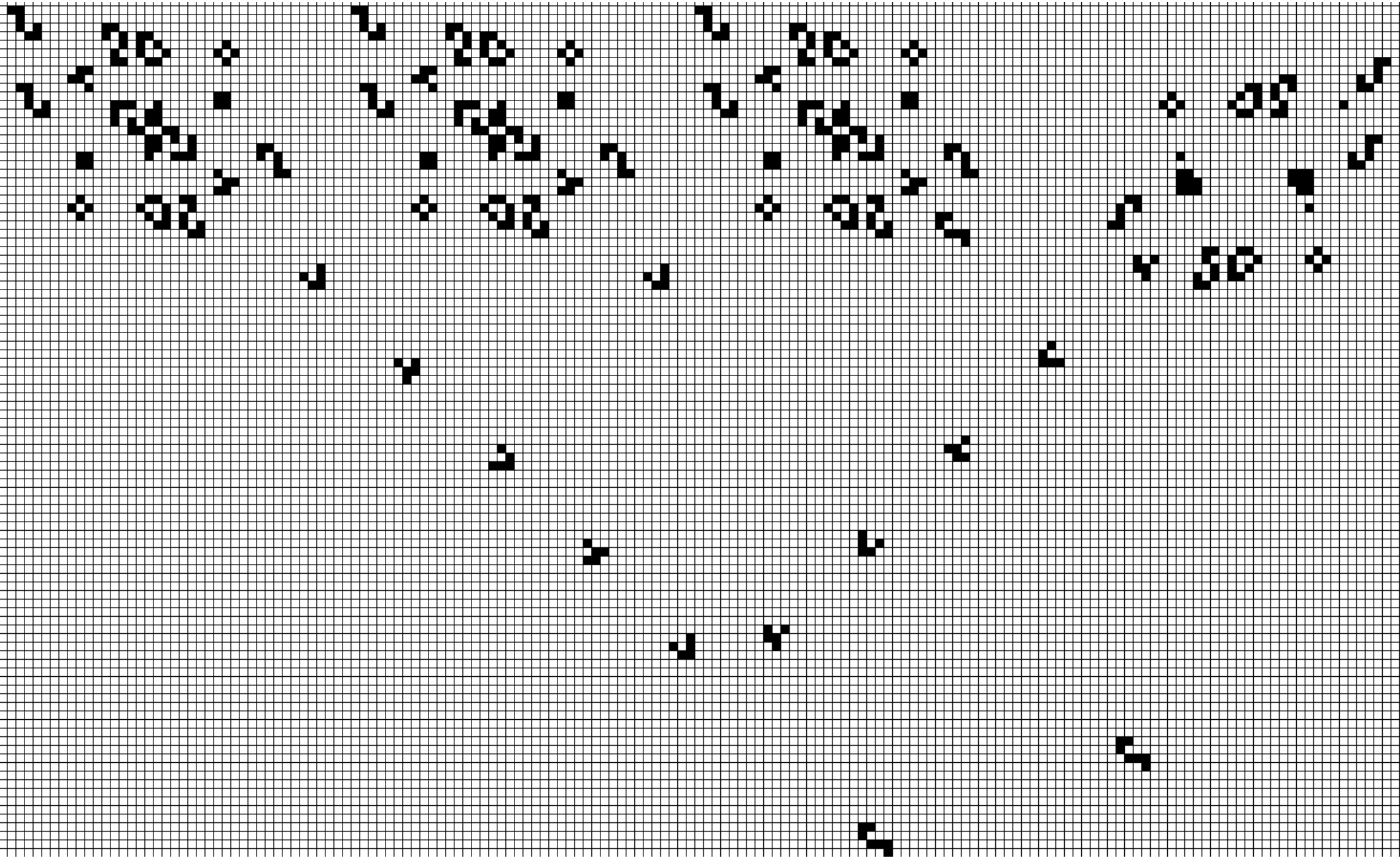
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

274/414

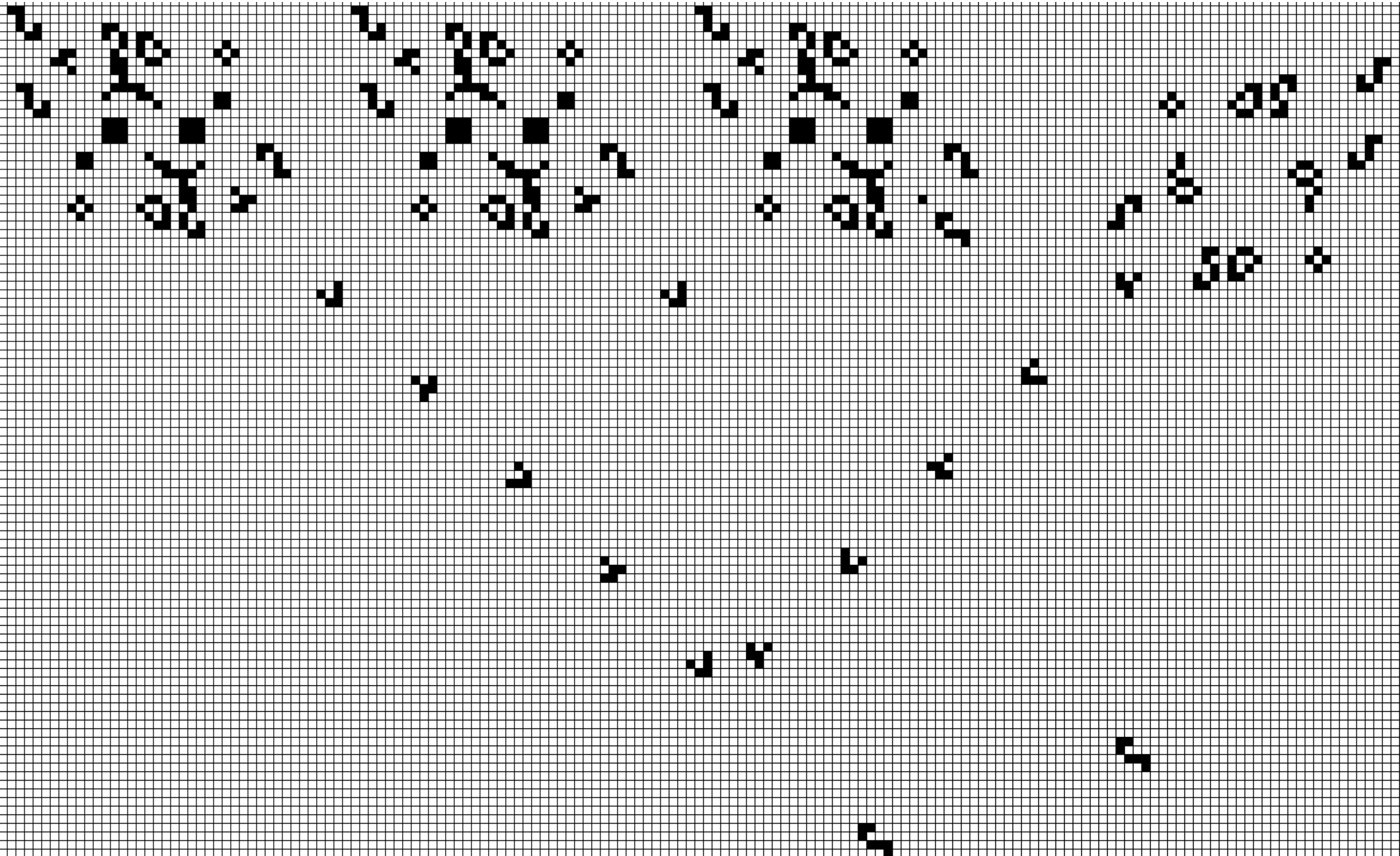
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

275/414

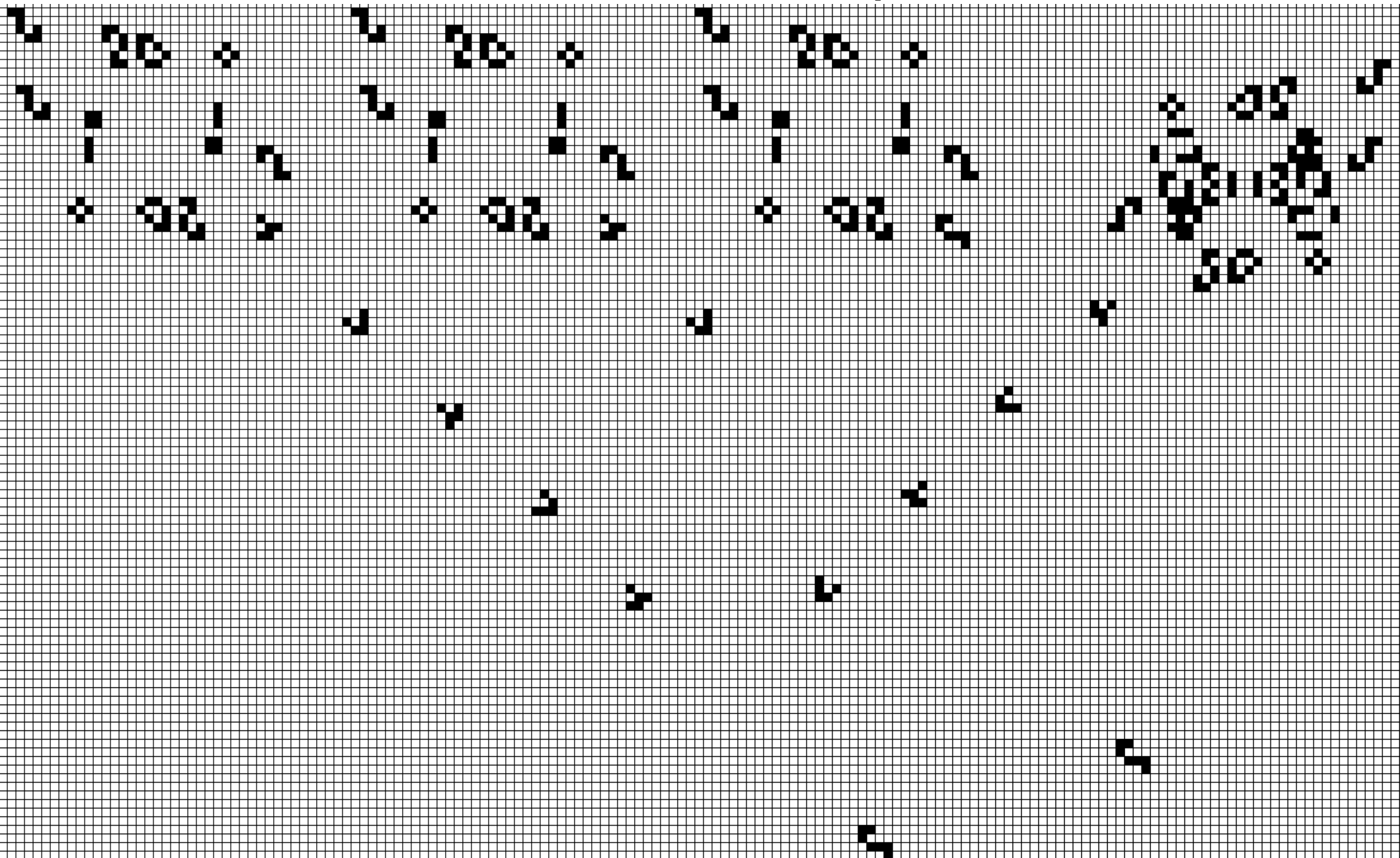
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

276/414

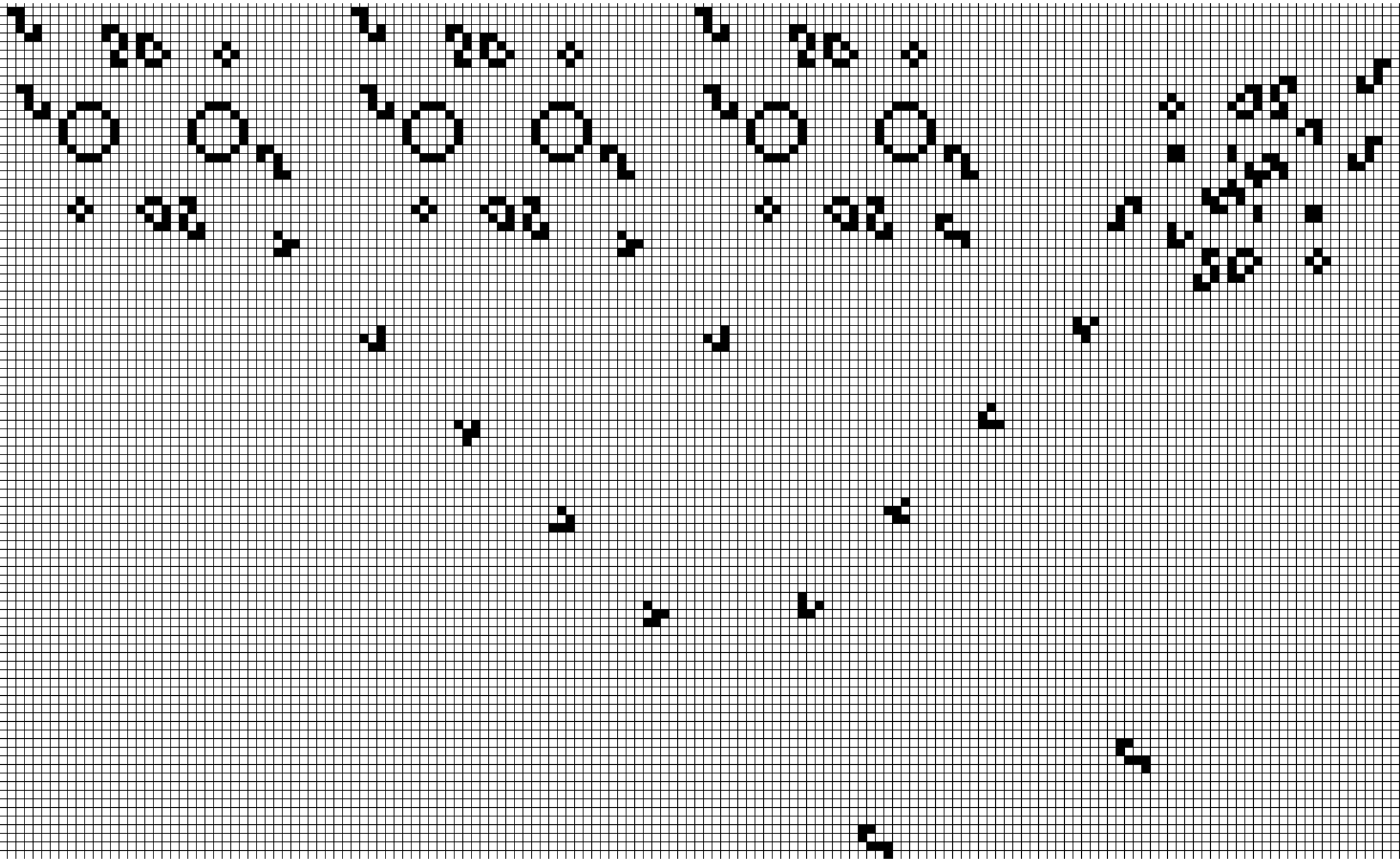
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

277/414

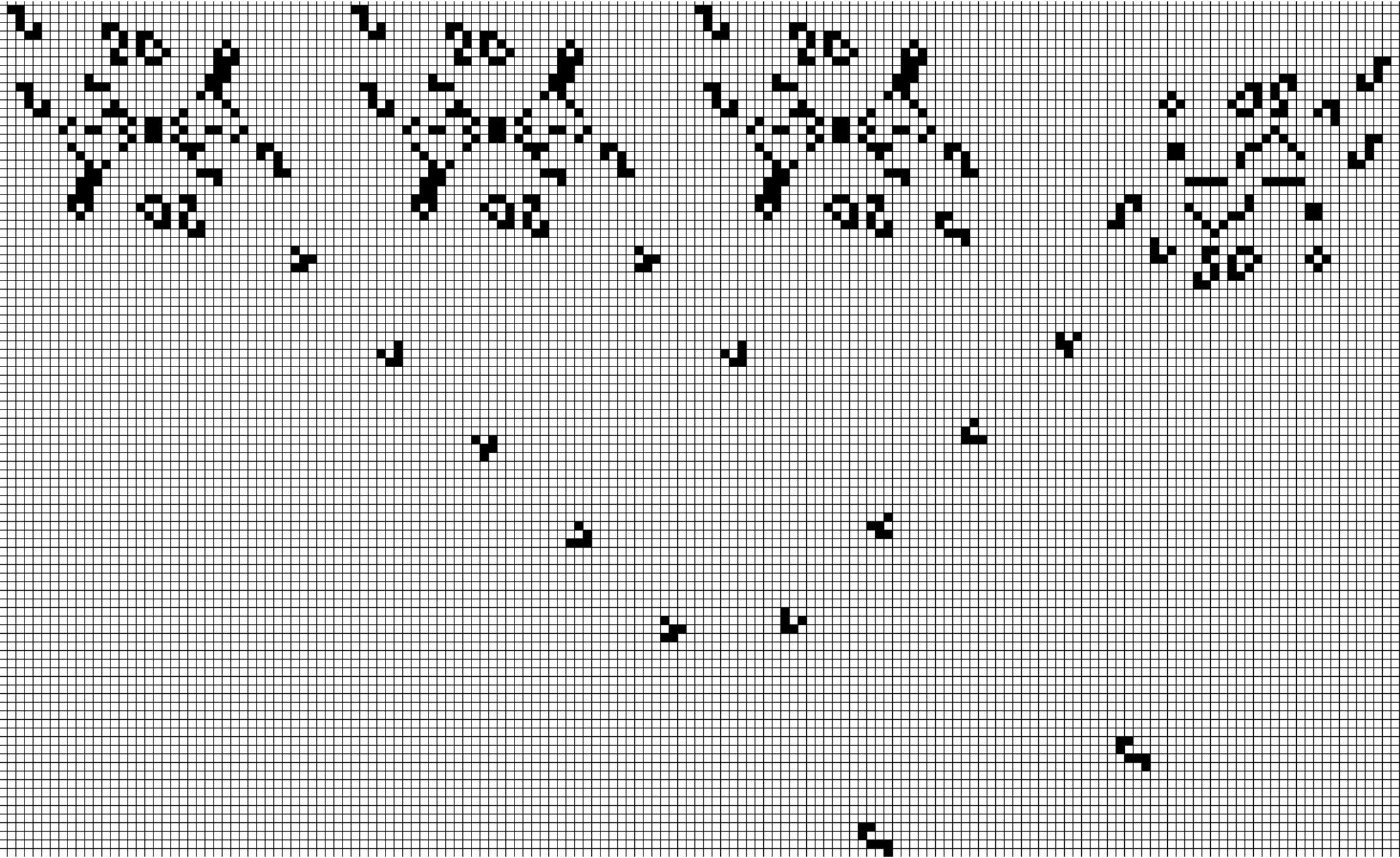
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

278/414

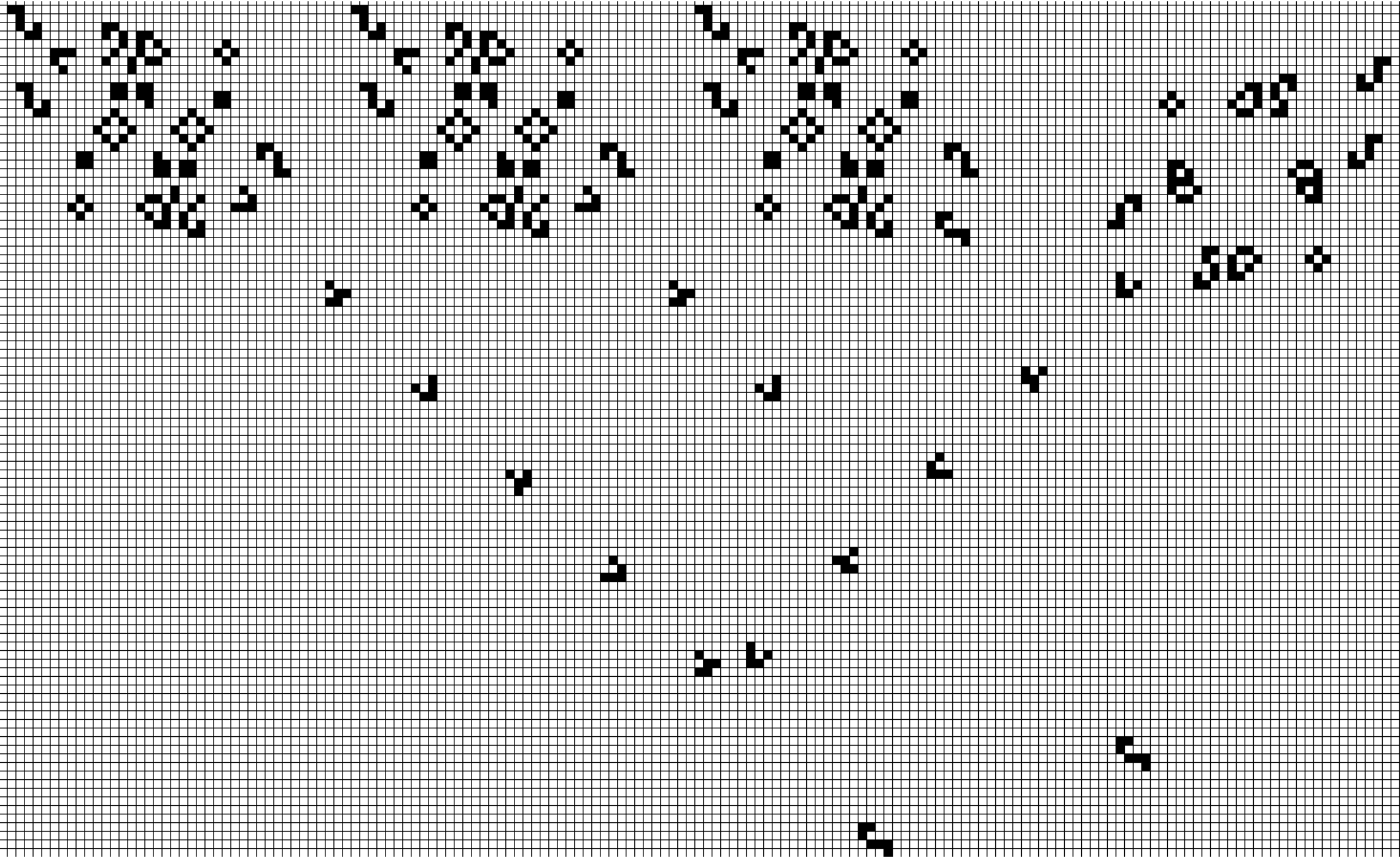
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

279/414

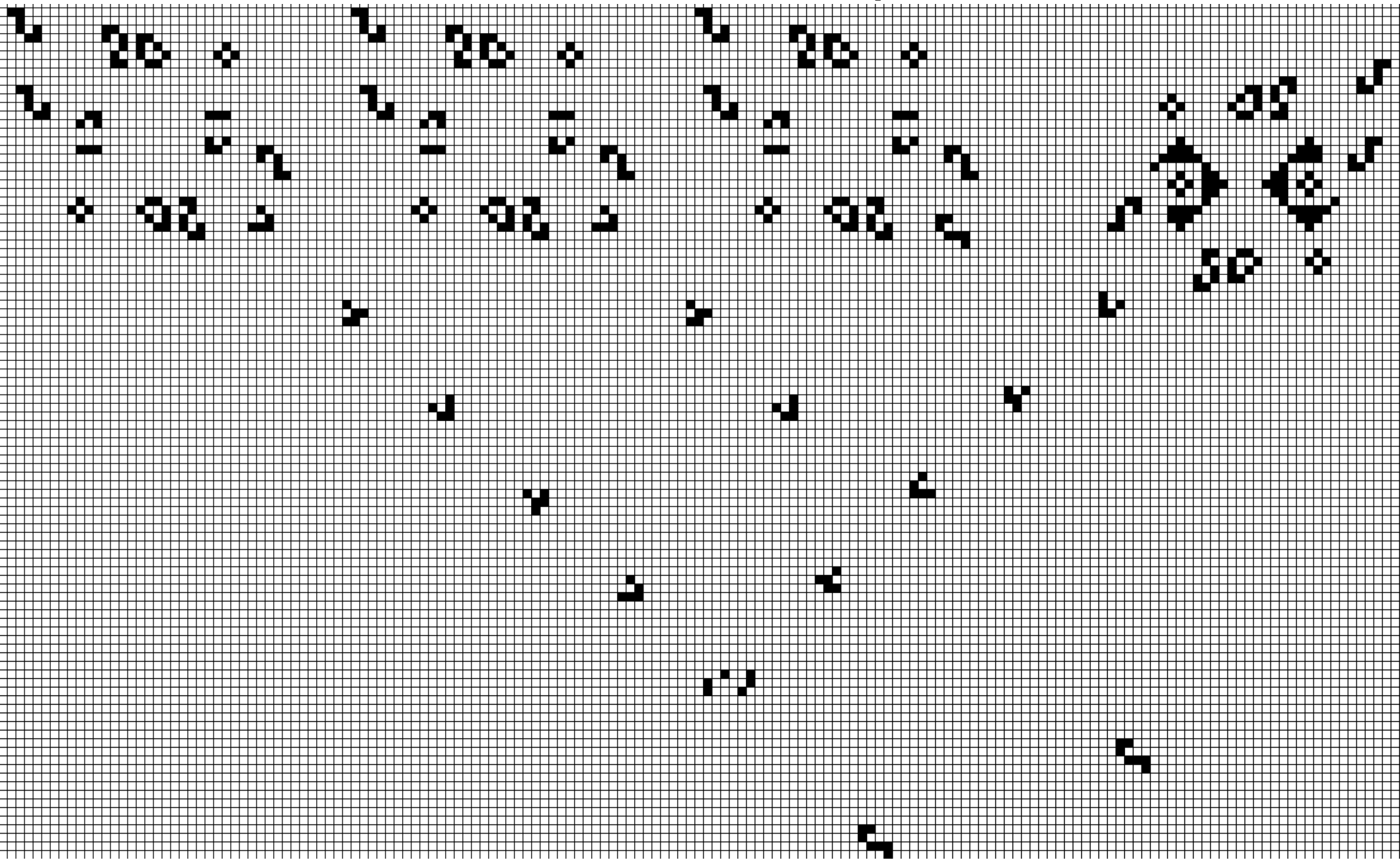
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

280/414

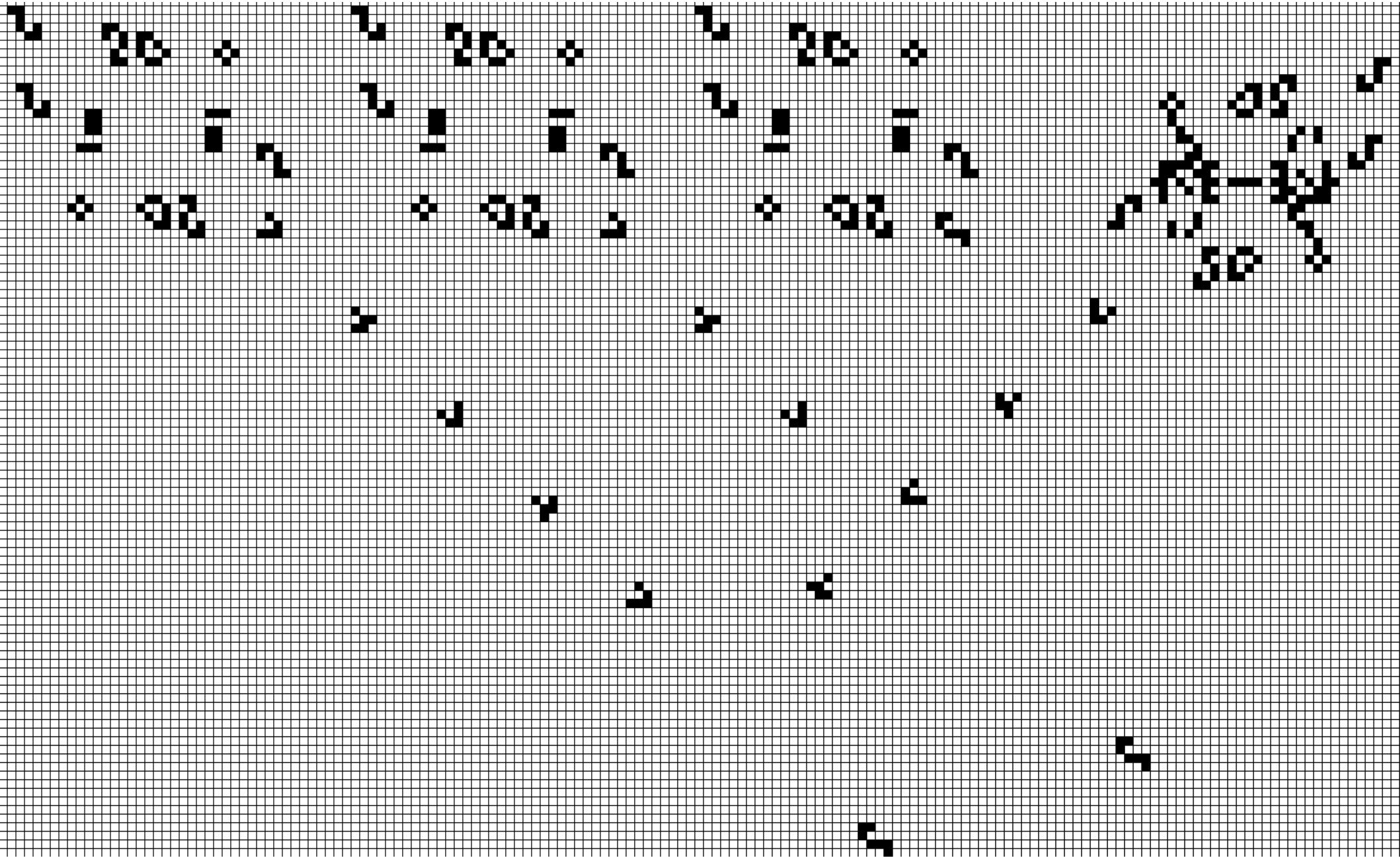
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

281/414

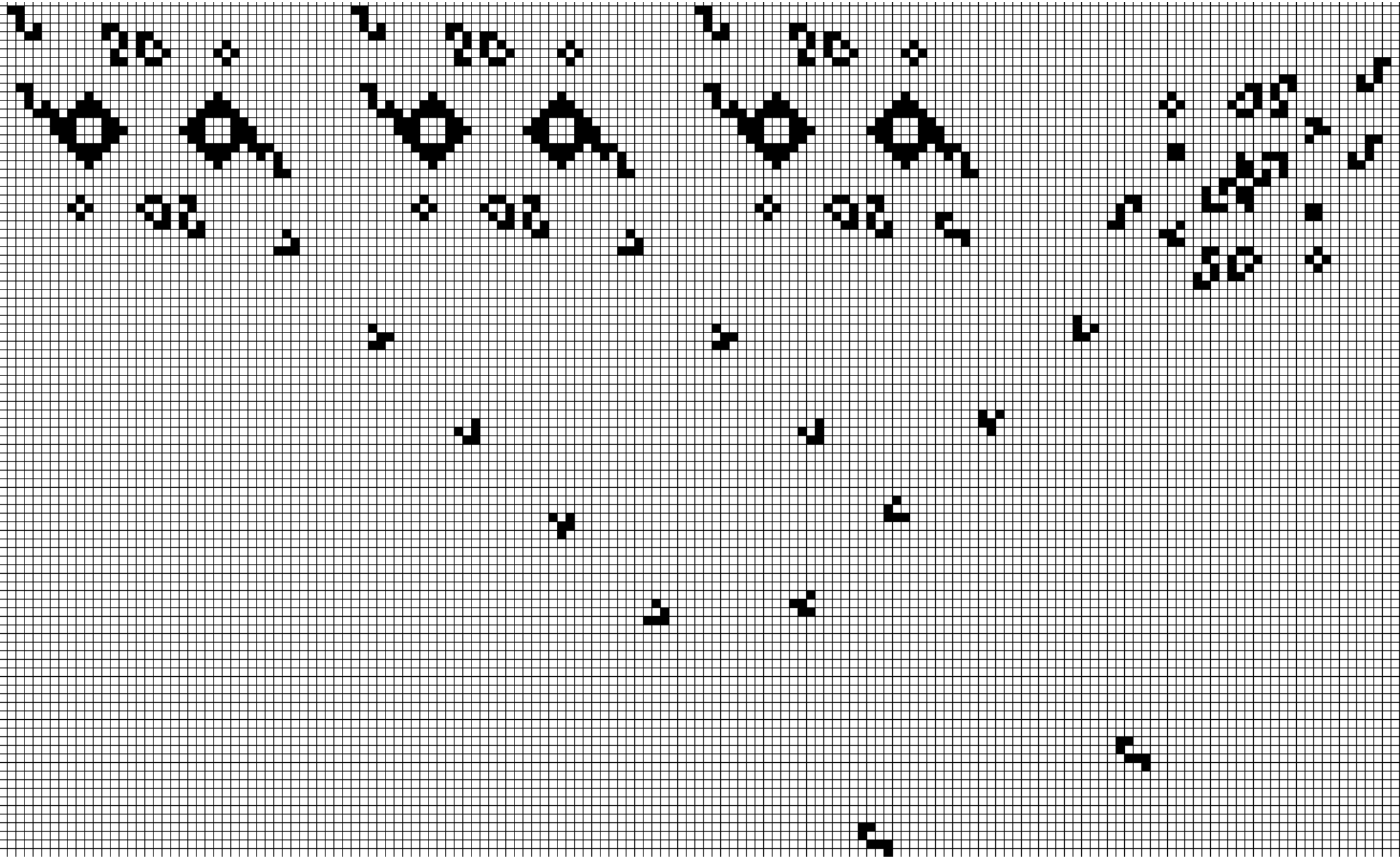
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

282/414

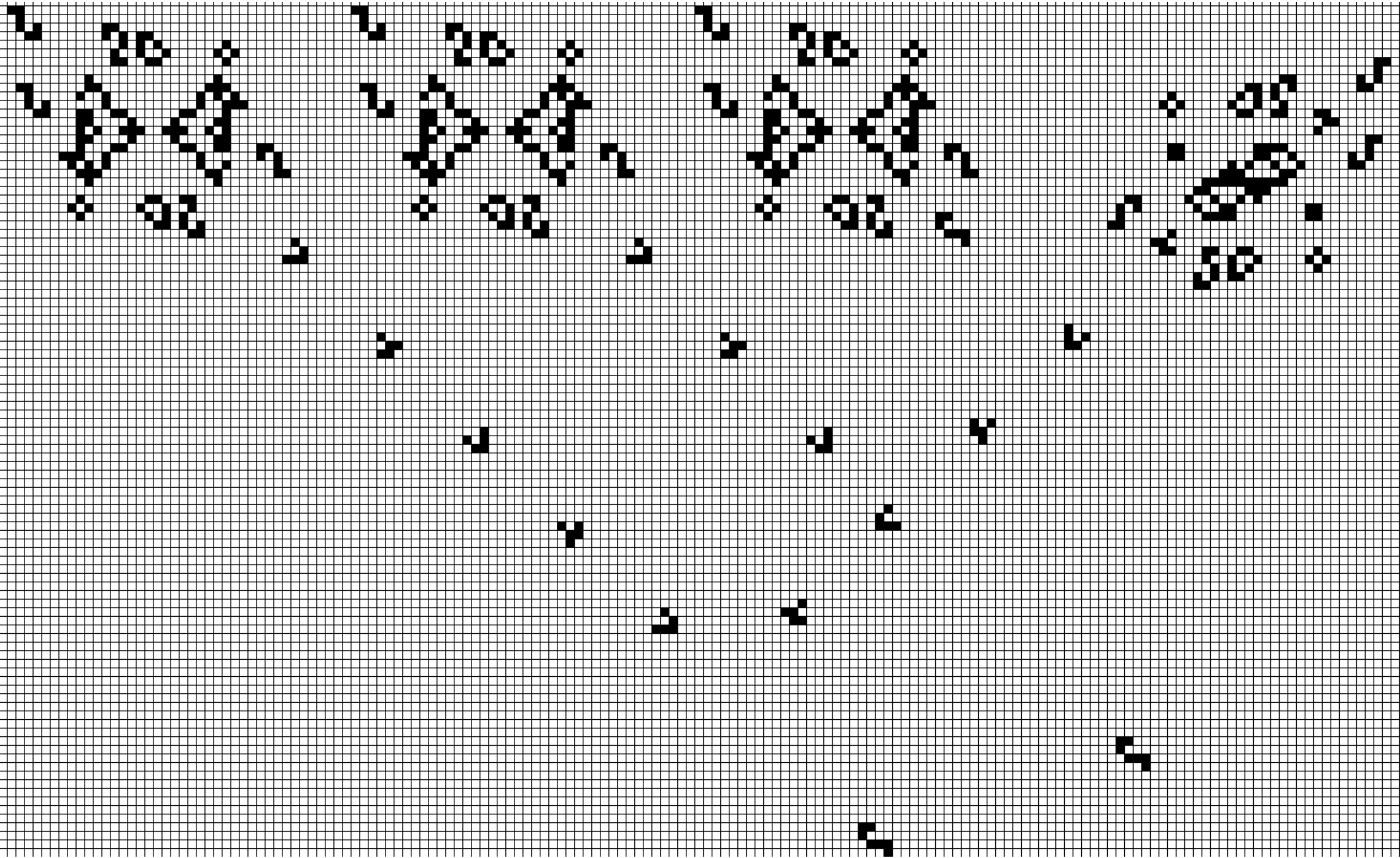
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

283/414

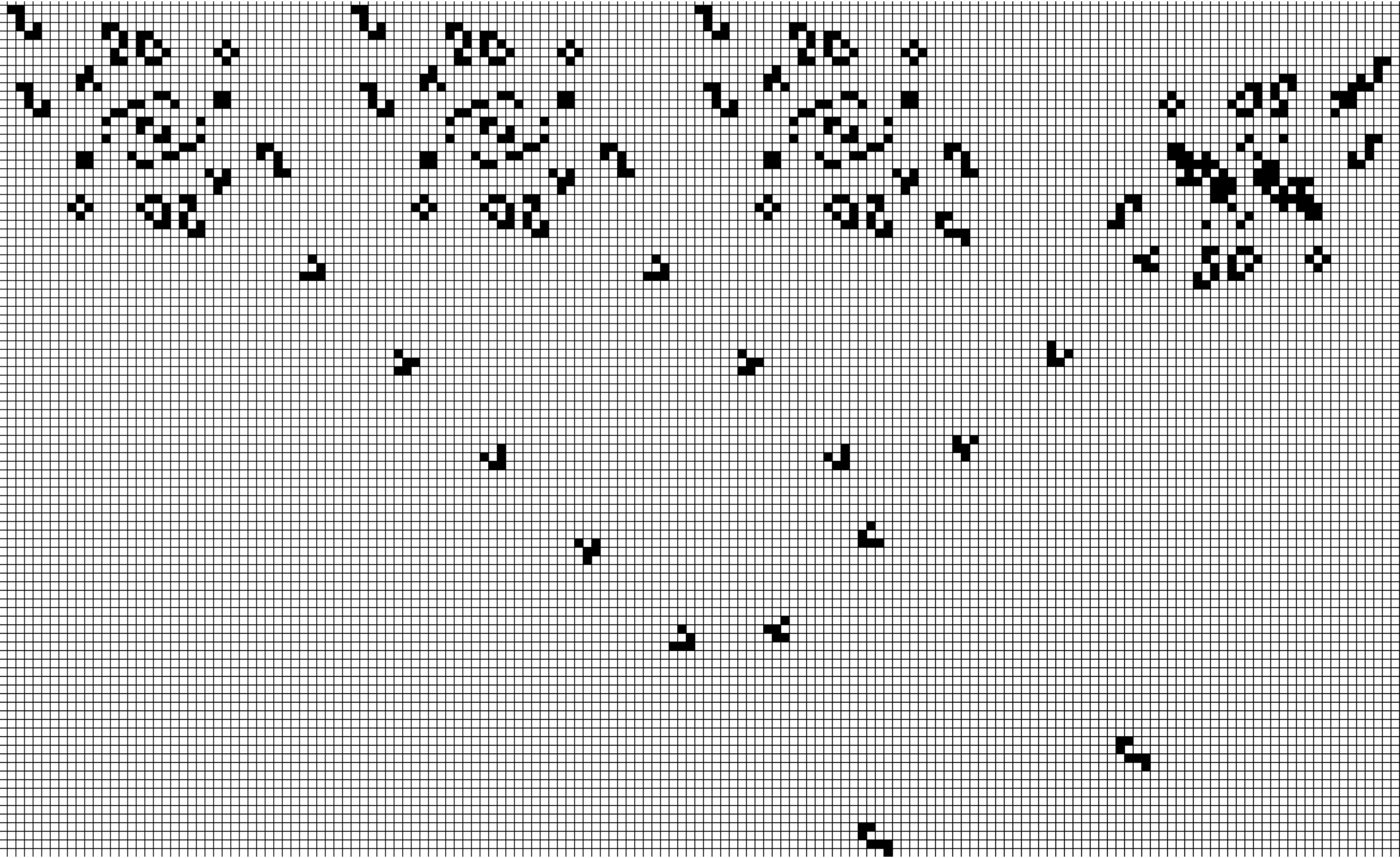
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

284/414

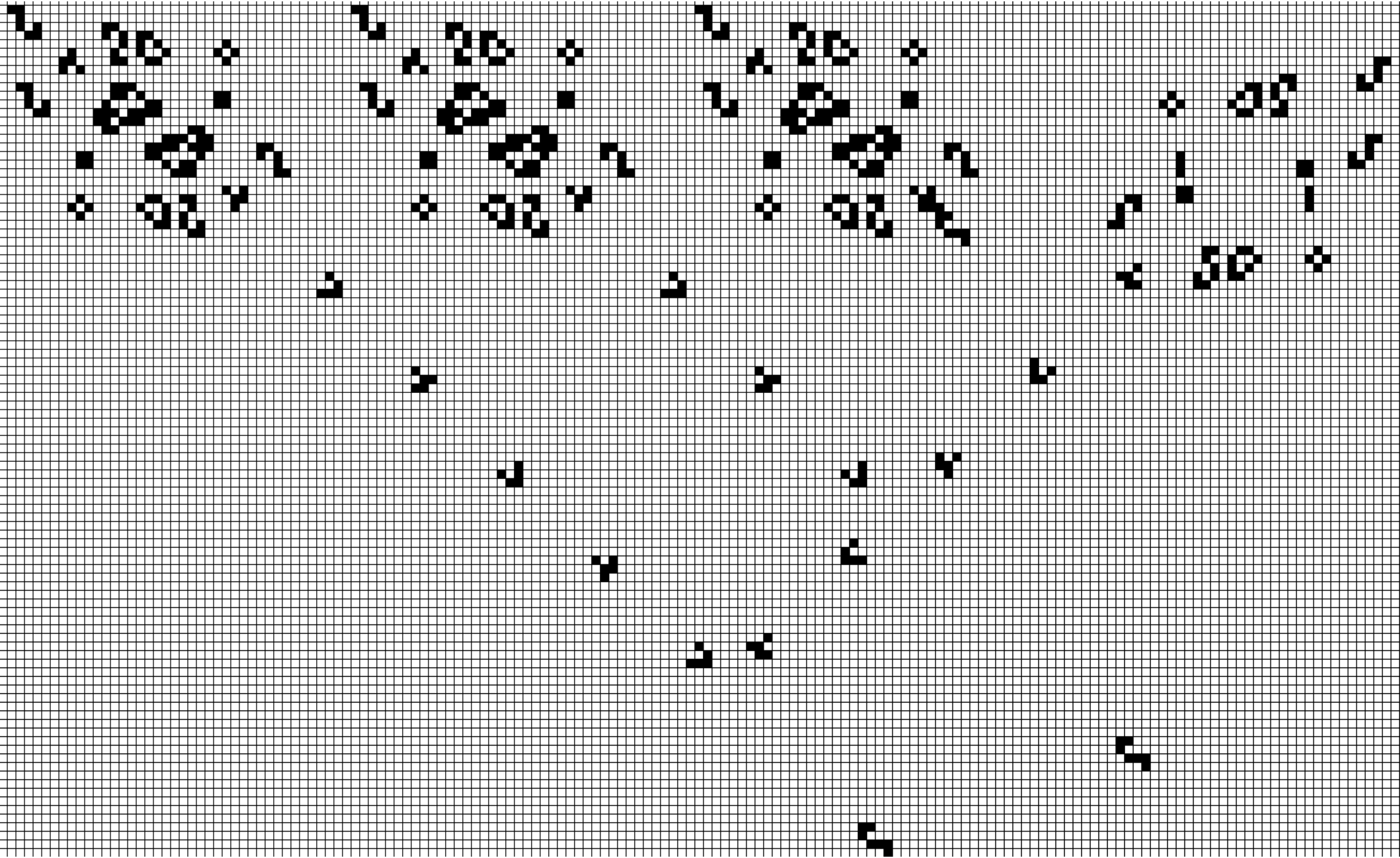
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

285/414

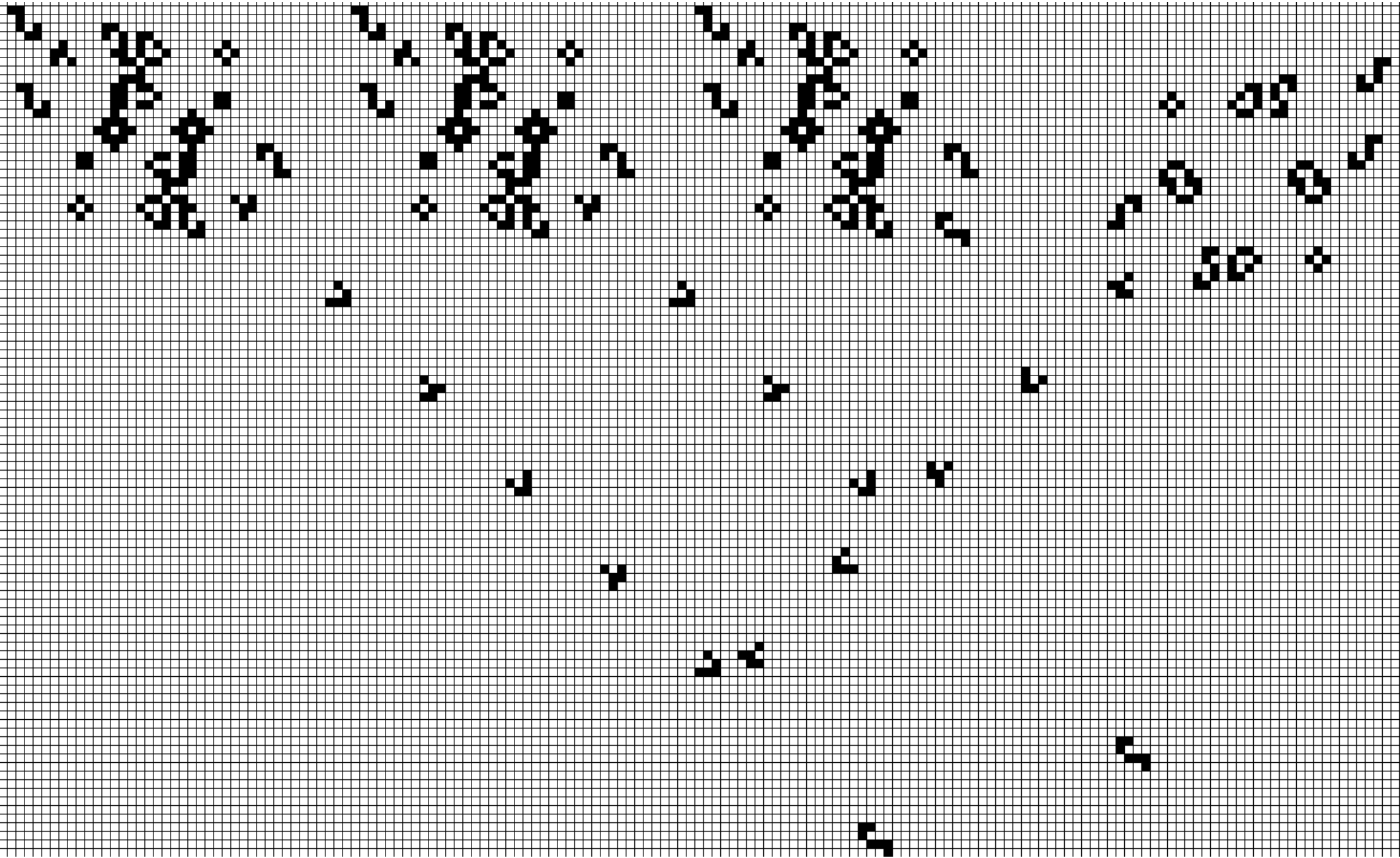
II. Outils nécessaires et composition du filtre



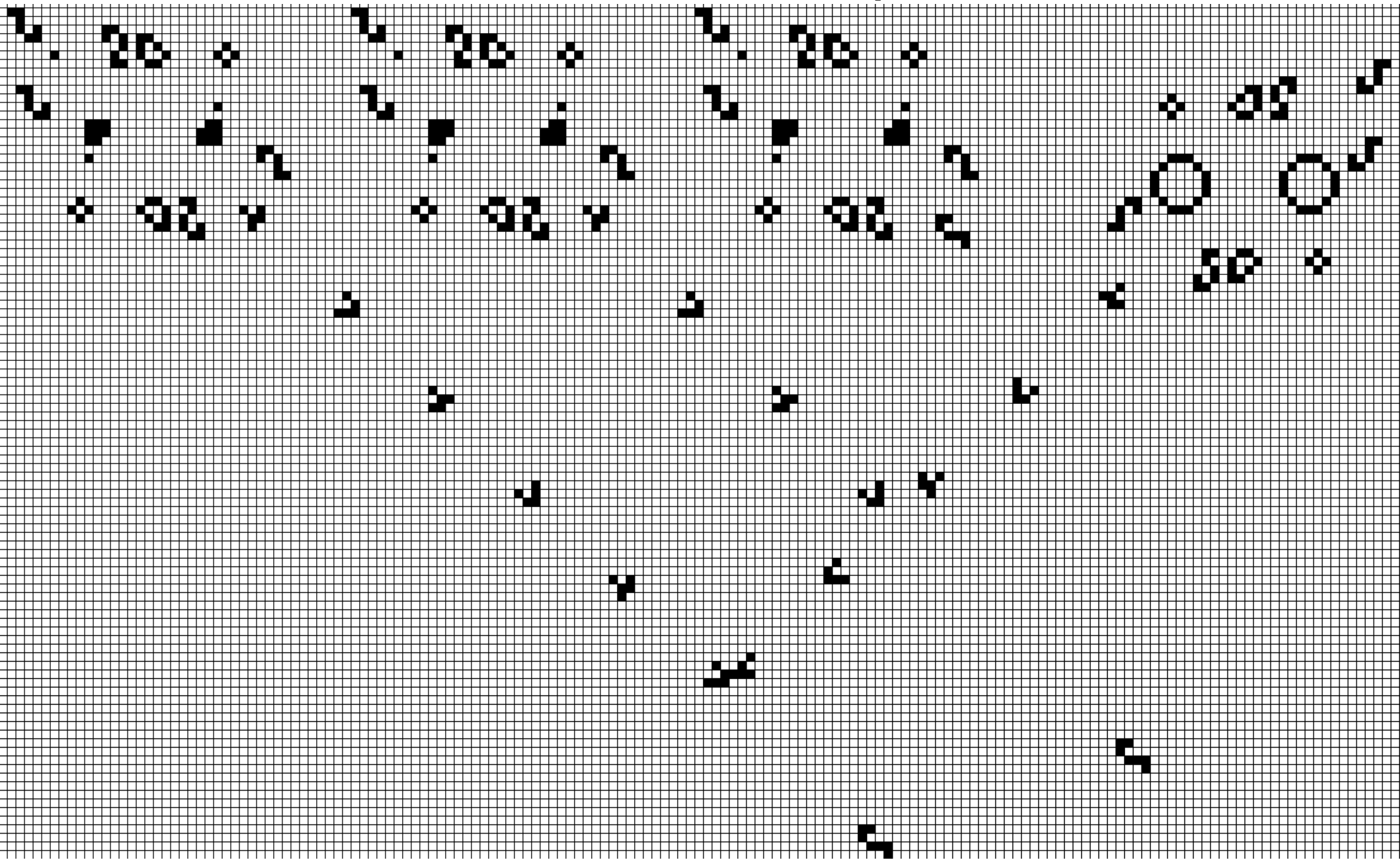
OR : Entrée 1 0

286/414

II. Outils nécessaires et composition du filtre



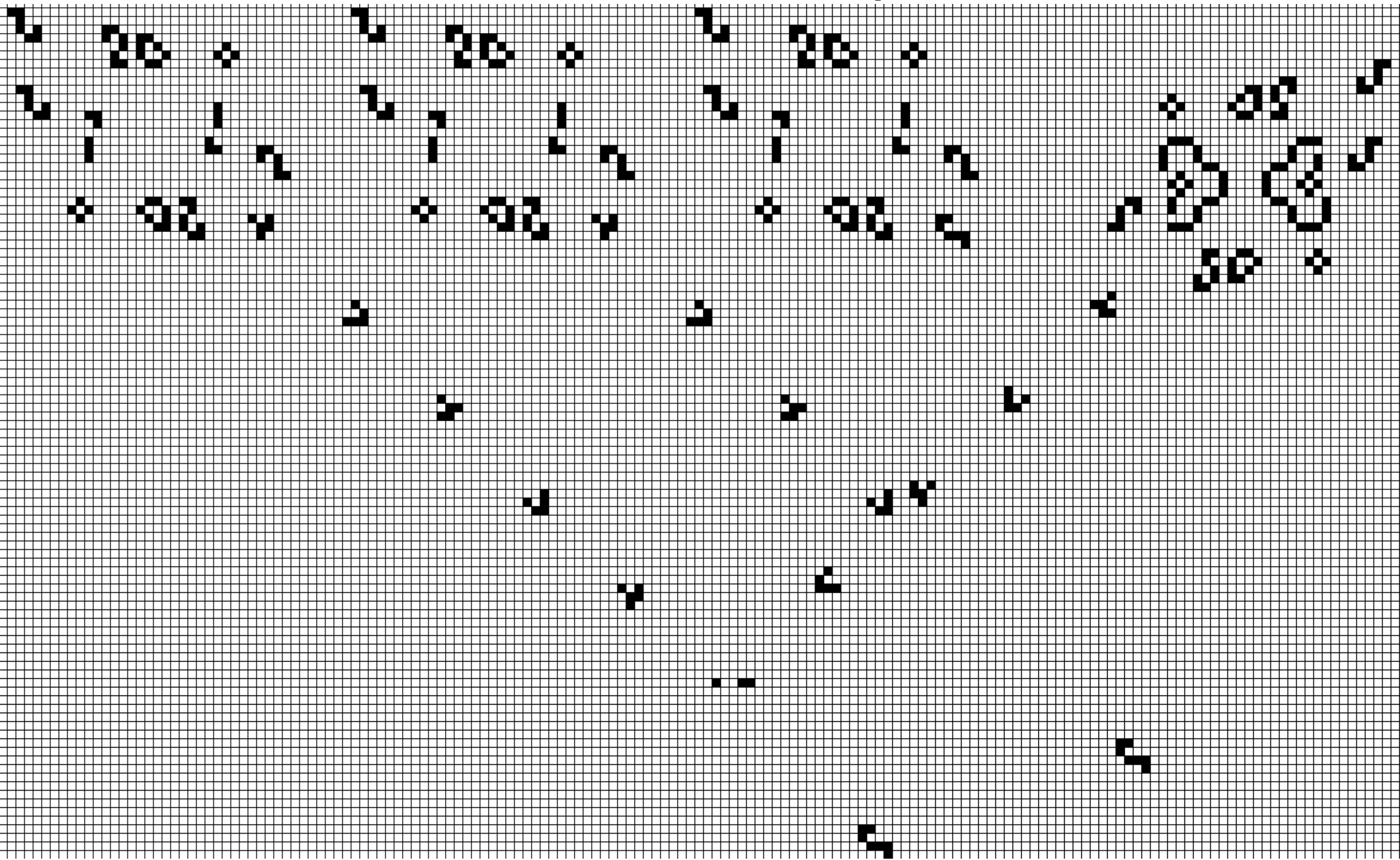
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

288/414

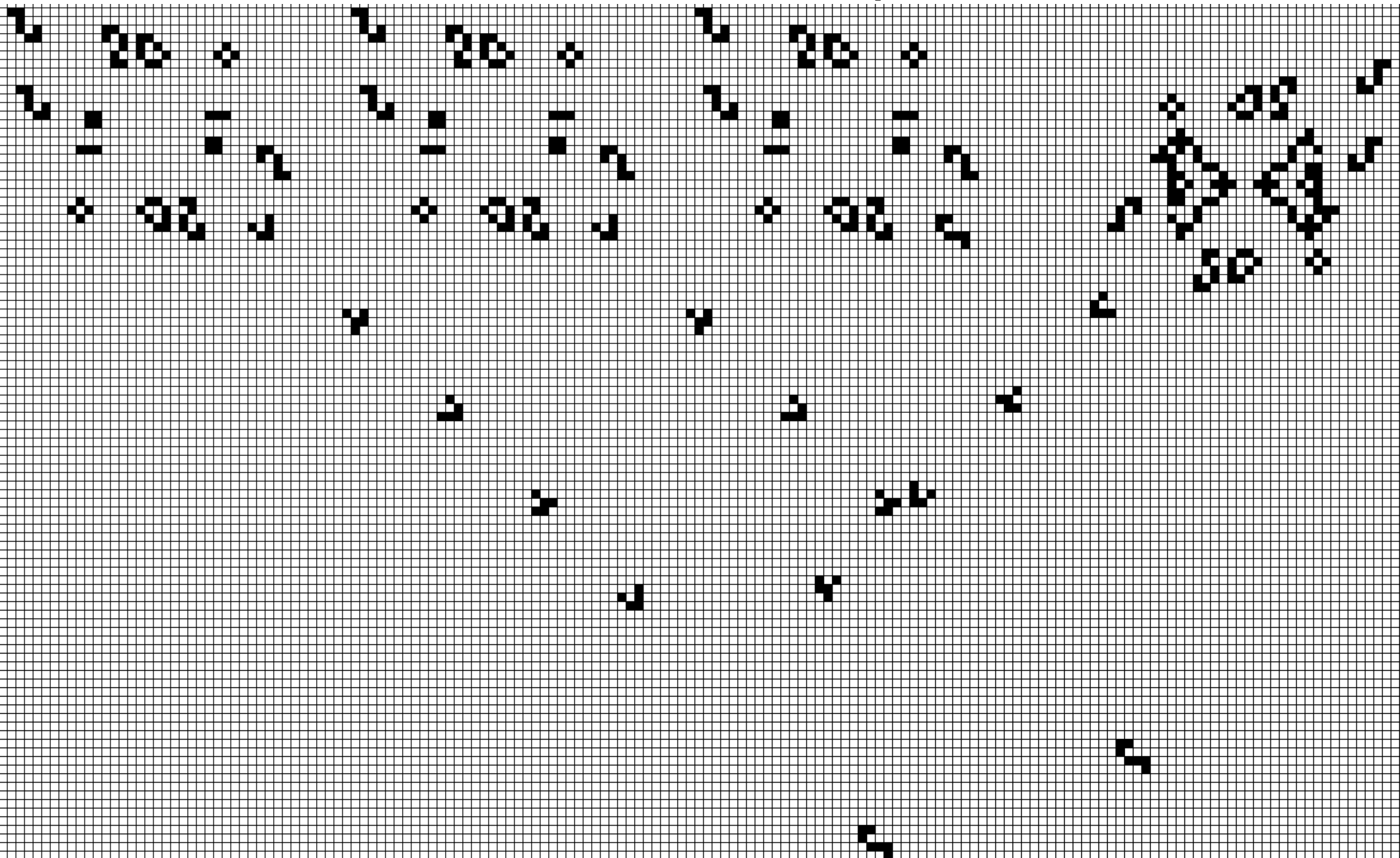
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

289/414

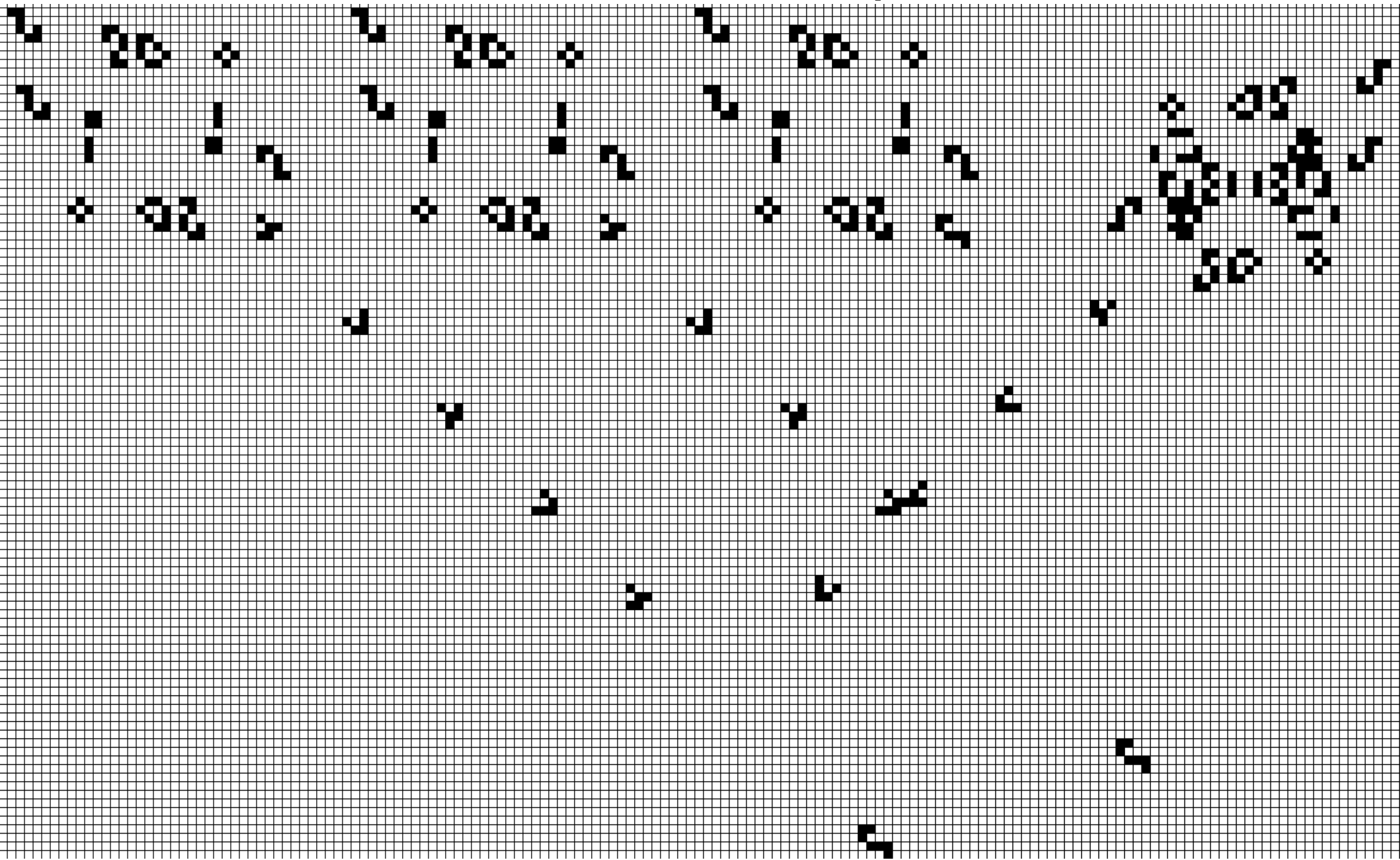
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

290/414

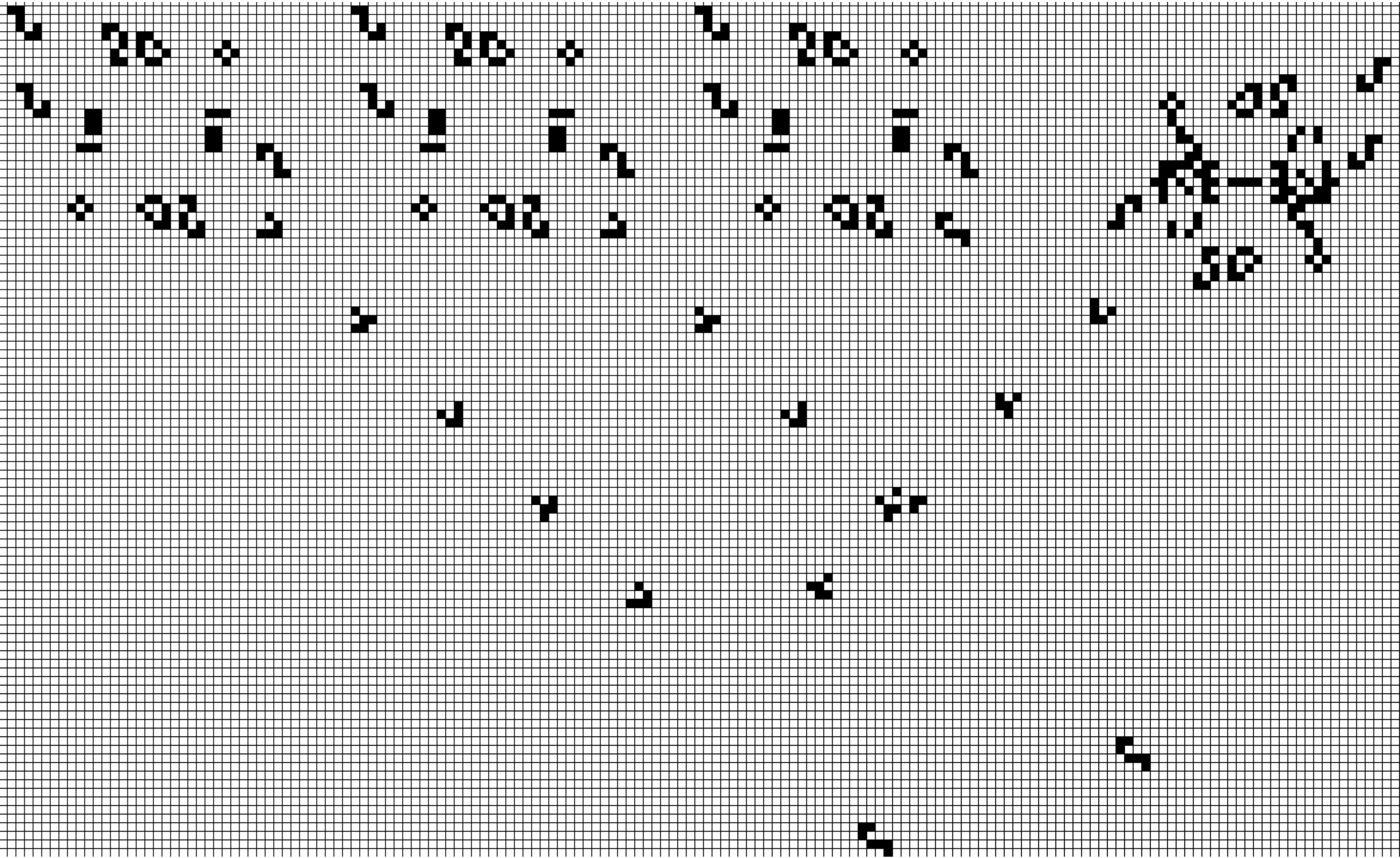
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

291/414

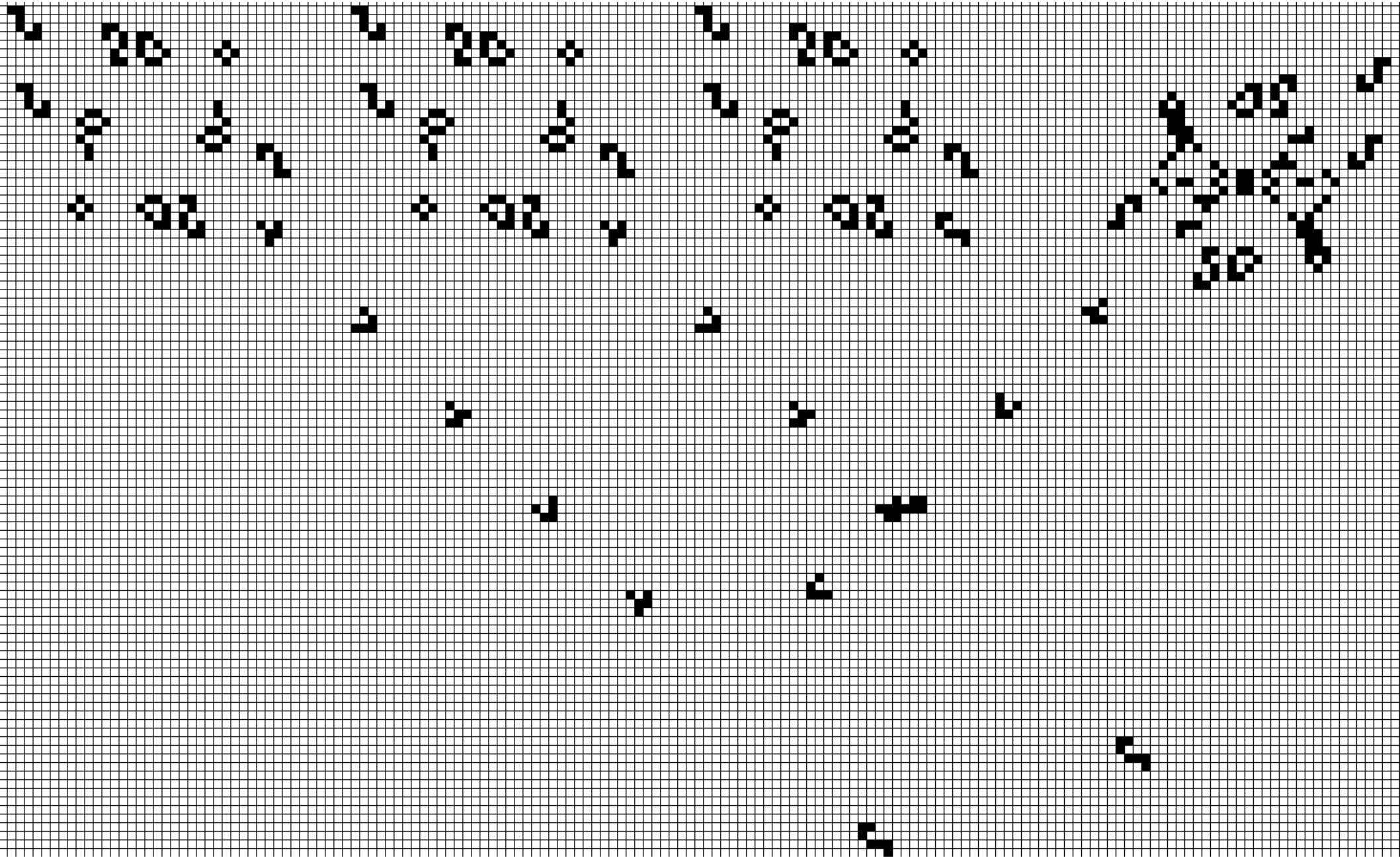
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

292/414

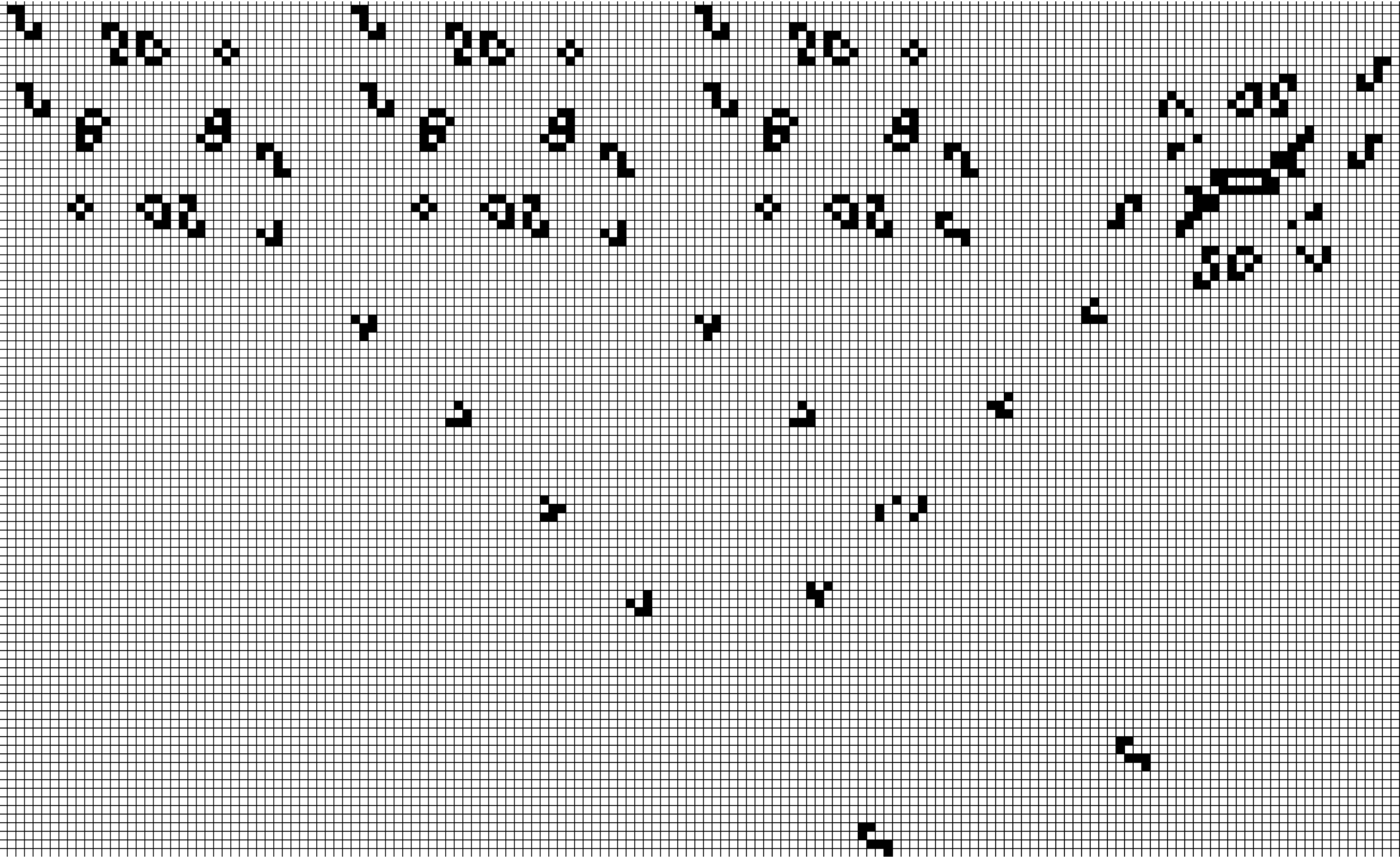
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

293/414

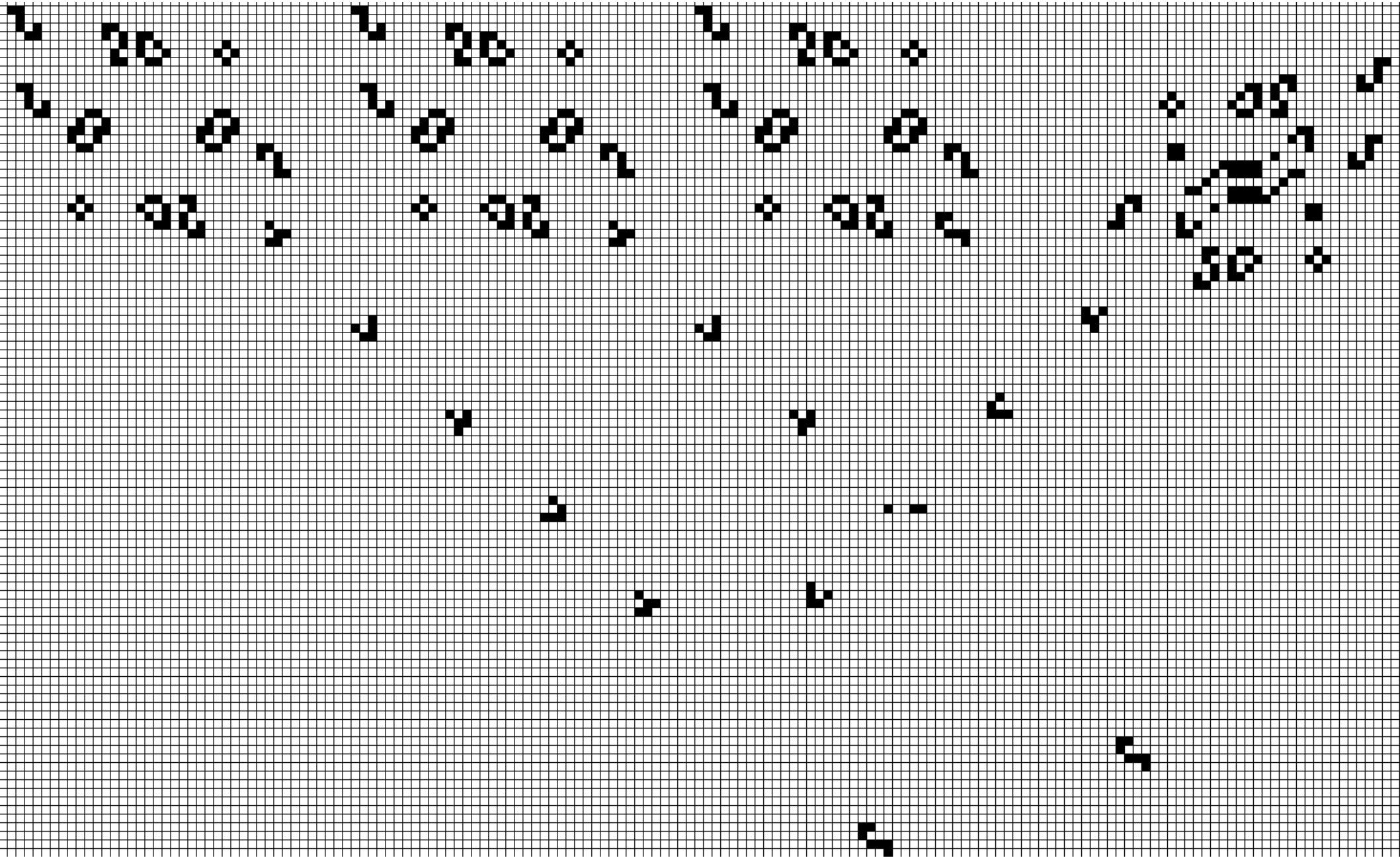
II. Outils nécessaires et composition du filtre



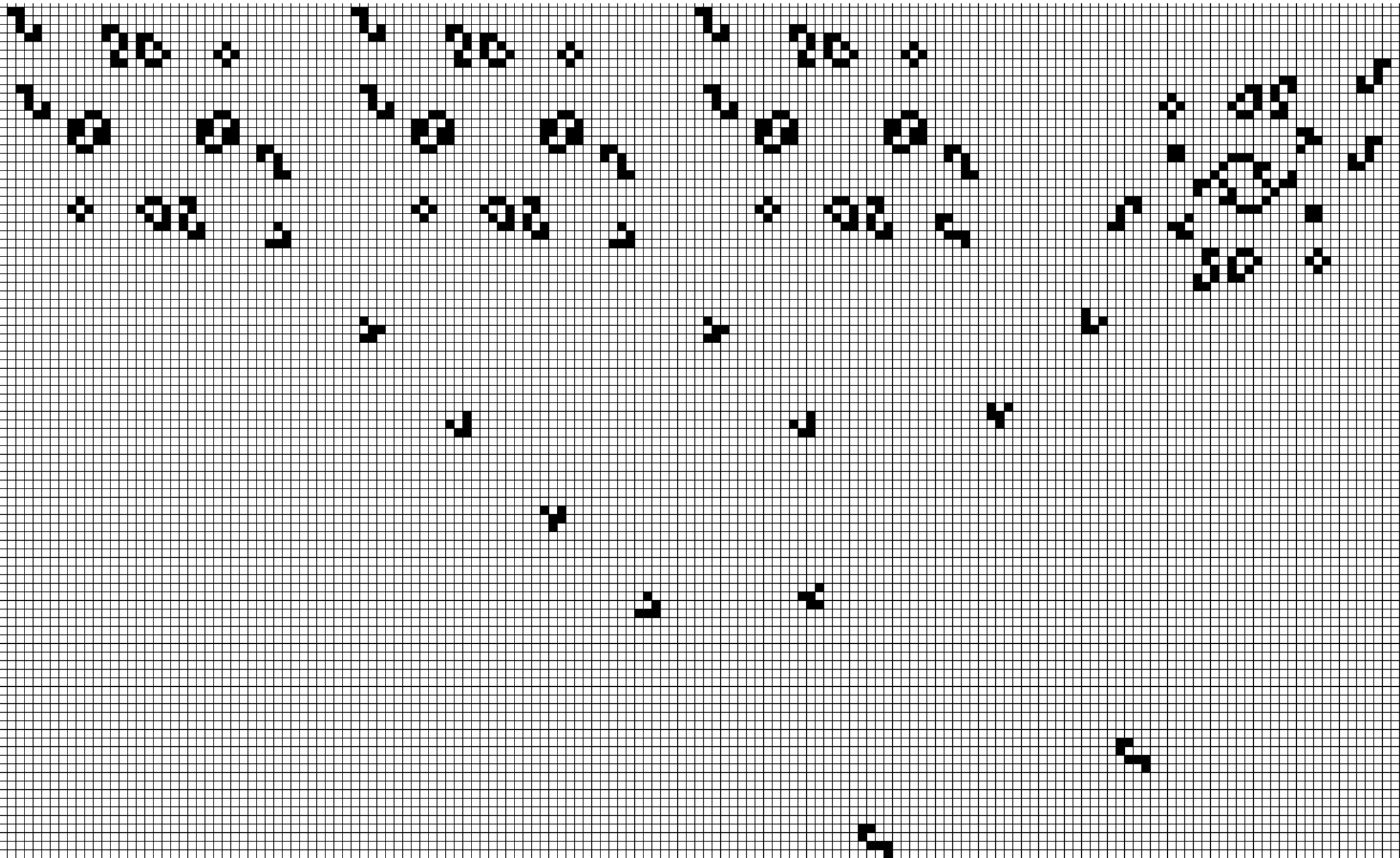
OR : Entrée 1 0

294/414

II. Outils nécessaires et composition du filtre



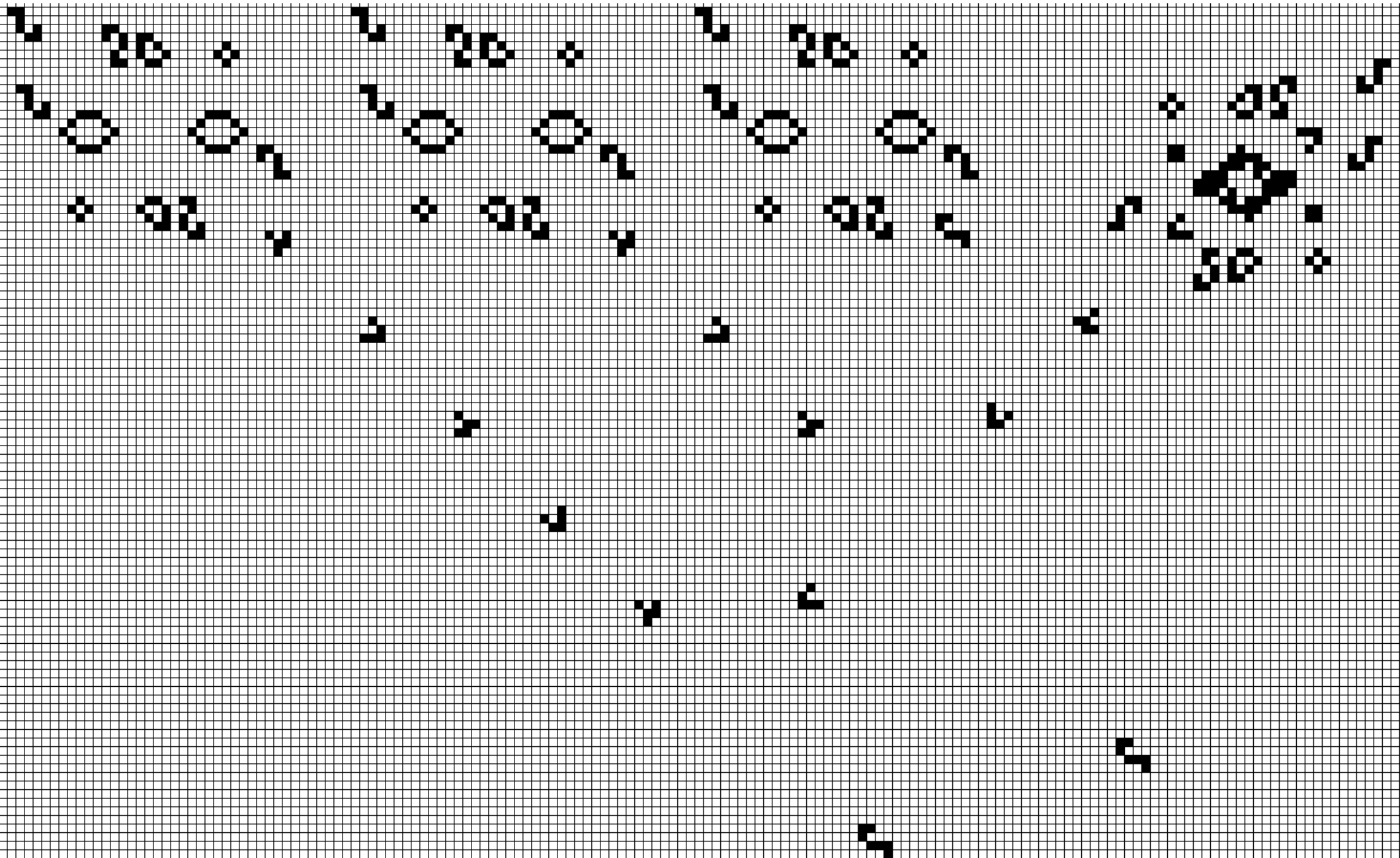
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

296/414

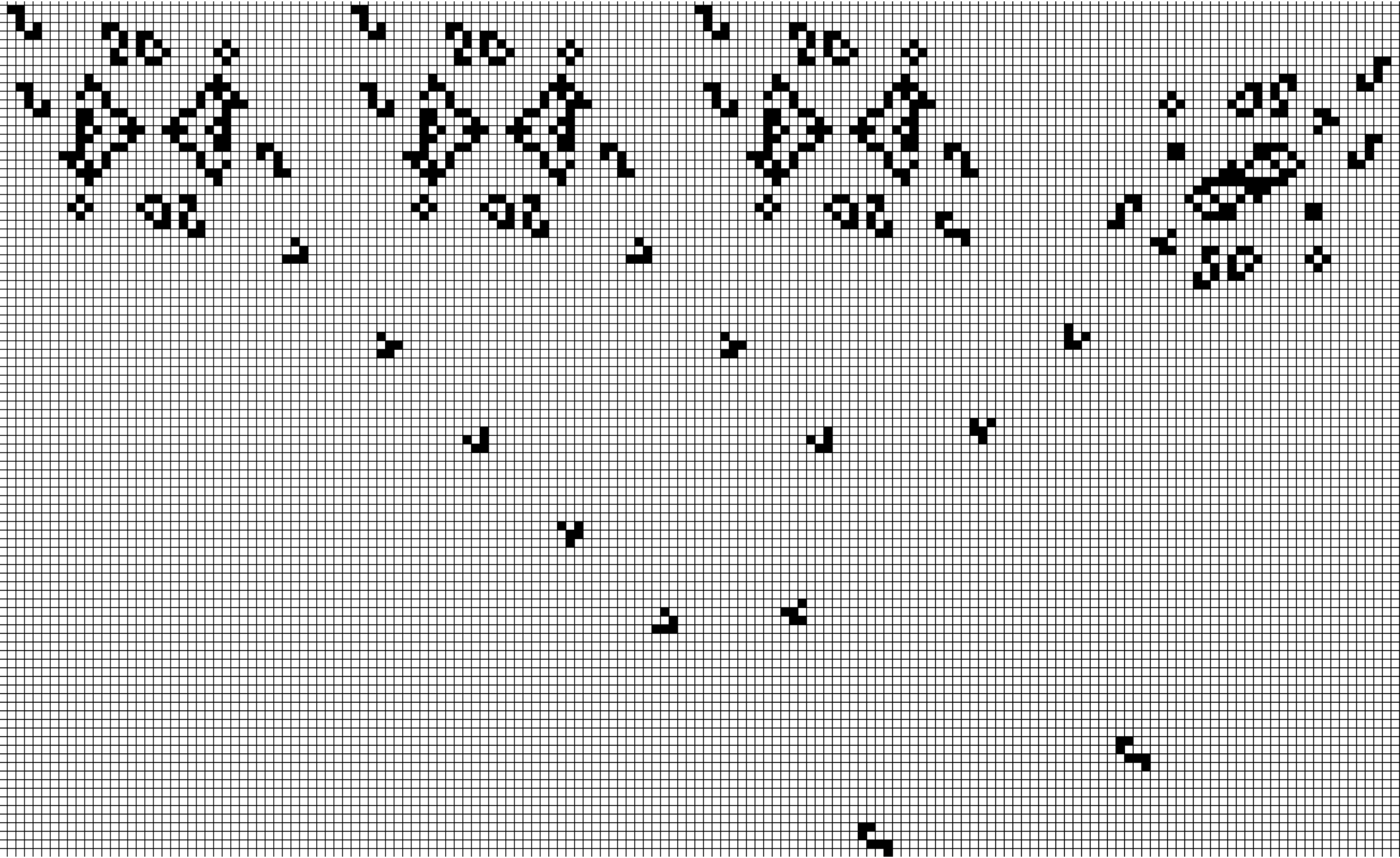
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

297/414

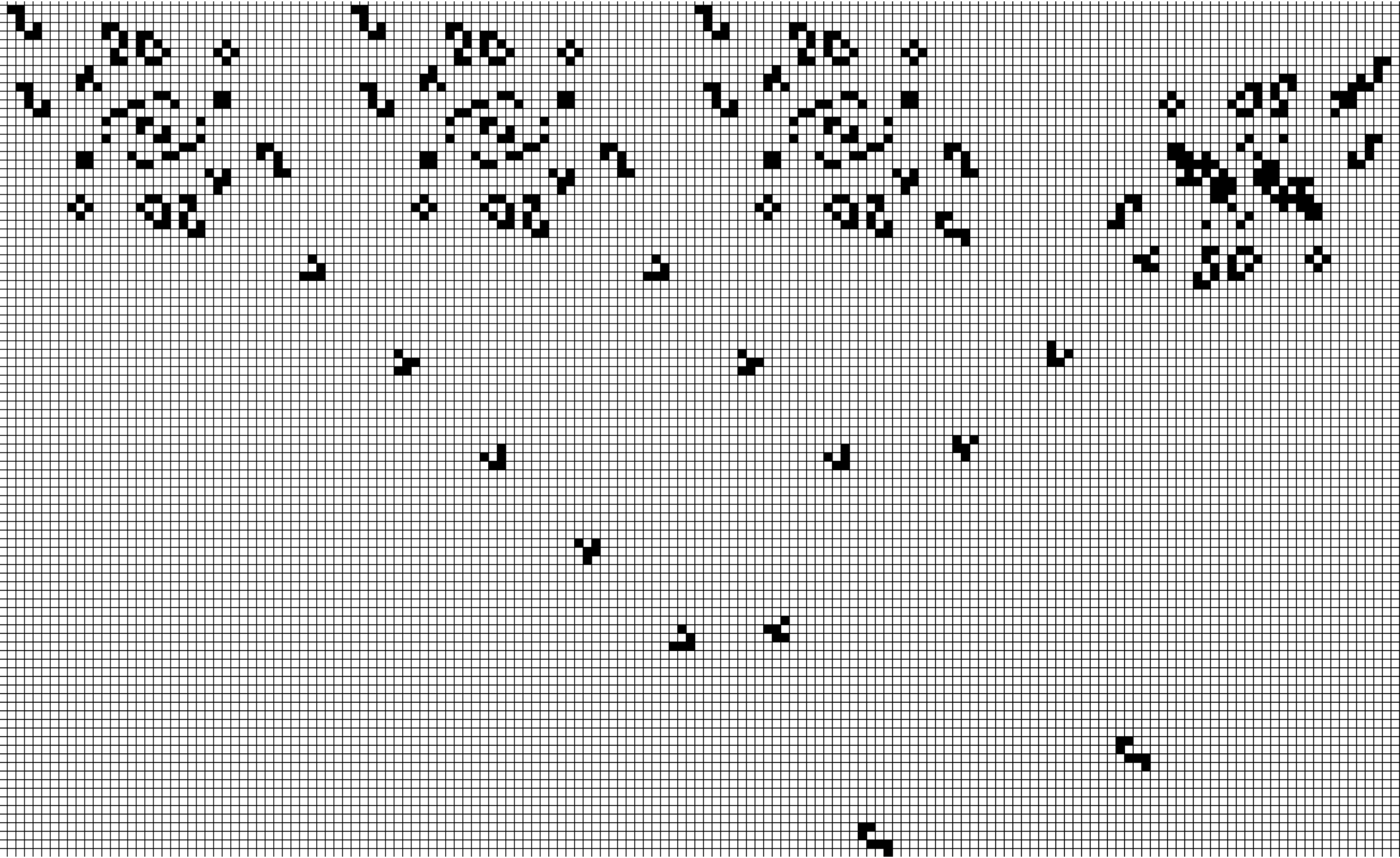
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

298/414

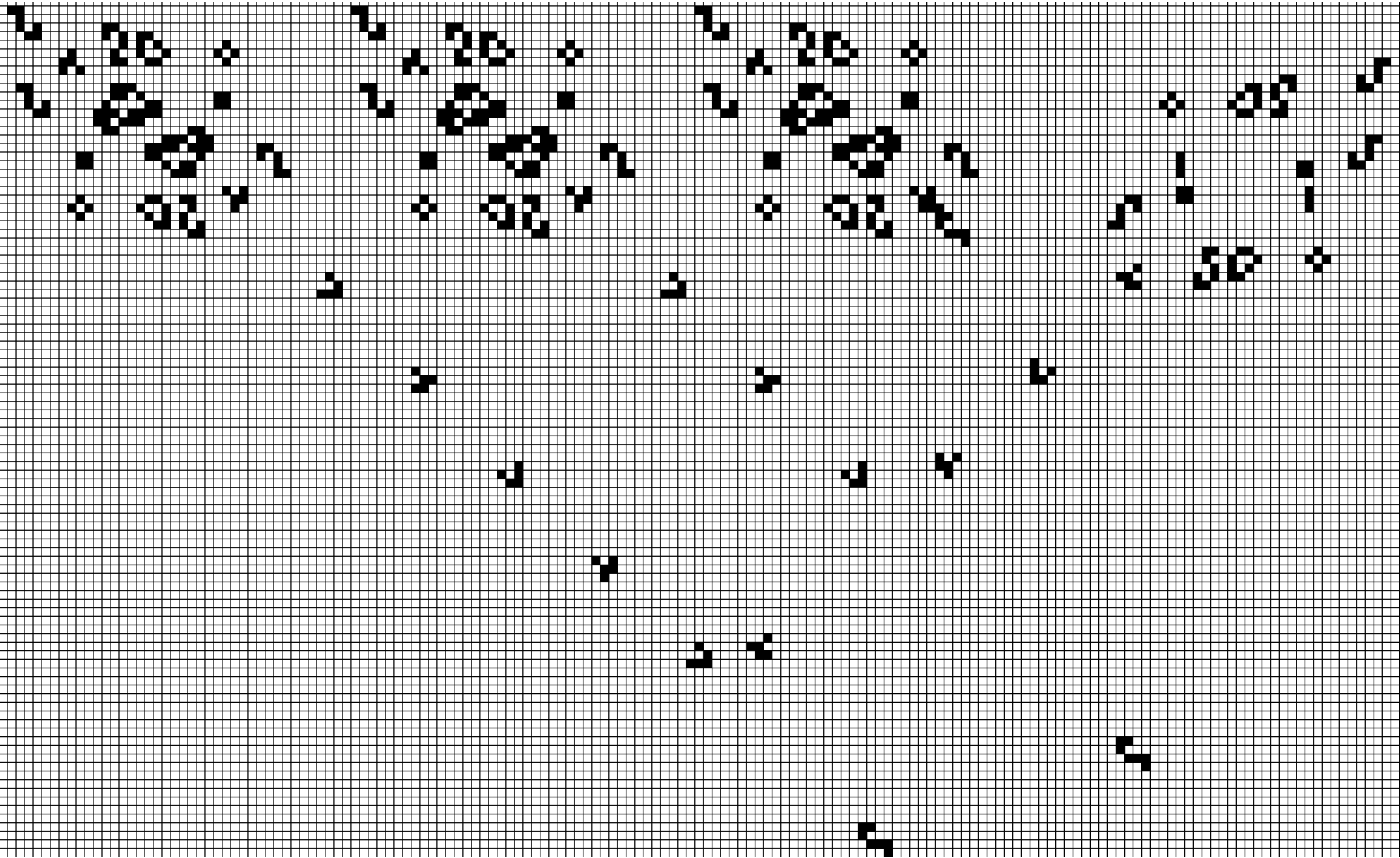
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

299/414

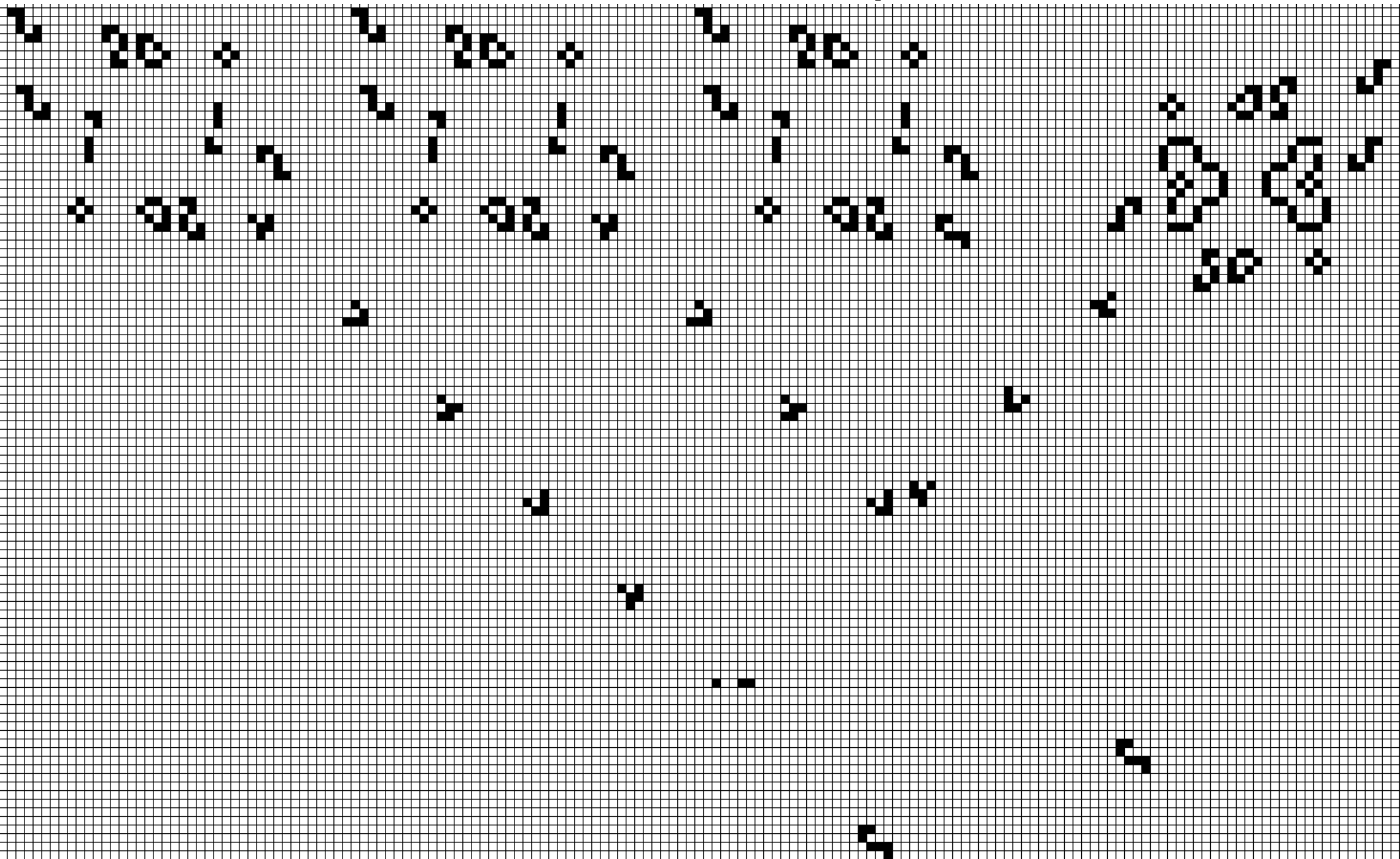
II. Outils nécessaires et composition du filtre



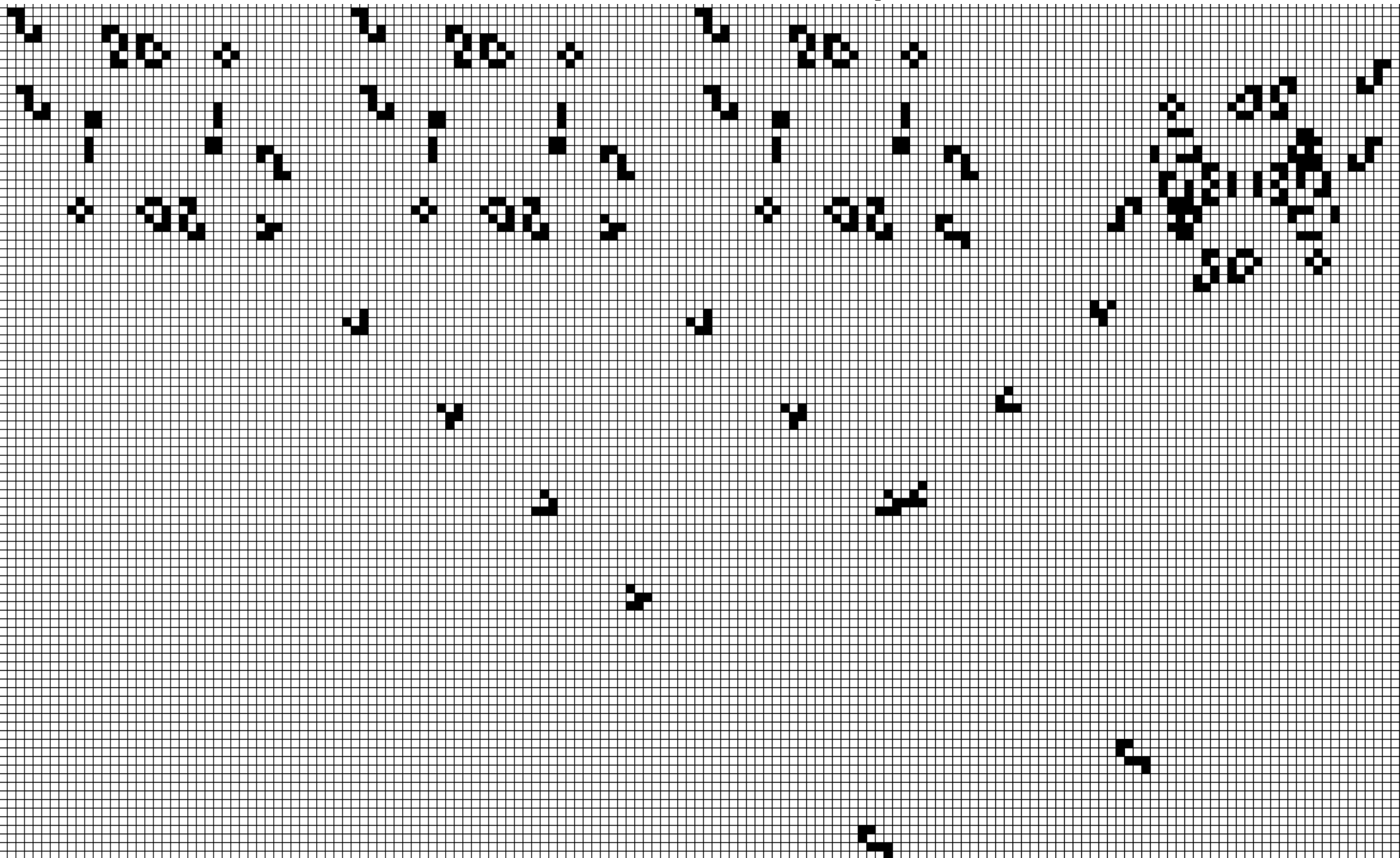
OR : Entrée 1 0

300/414

II. Outils nécessaires et composition du filtre



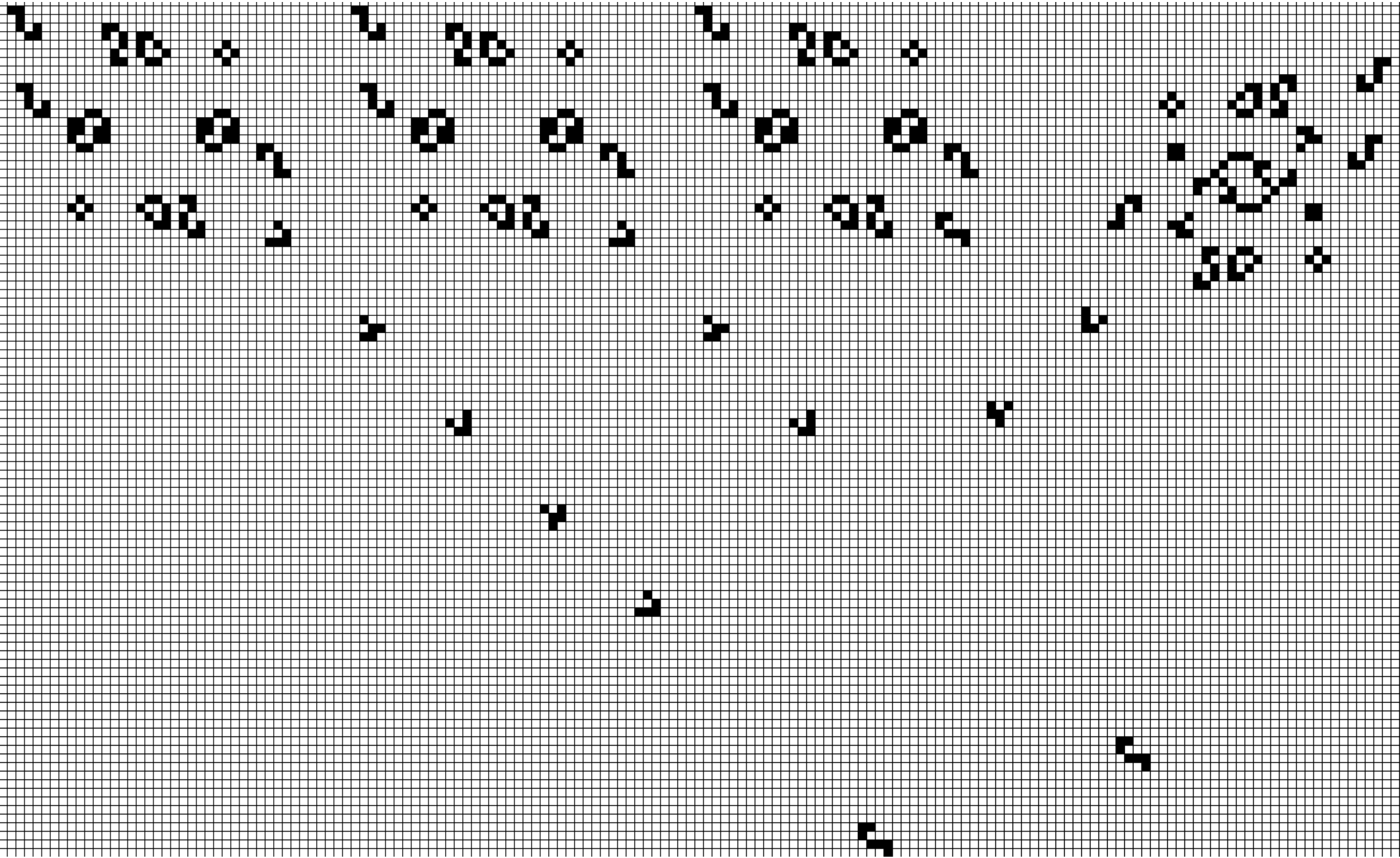
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

302/414

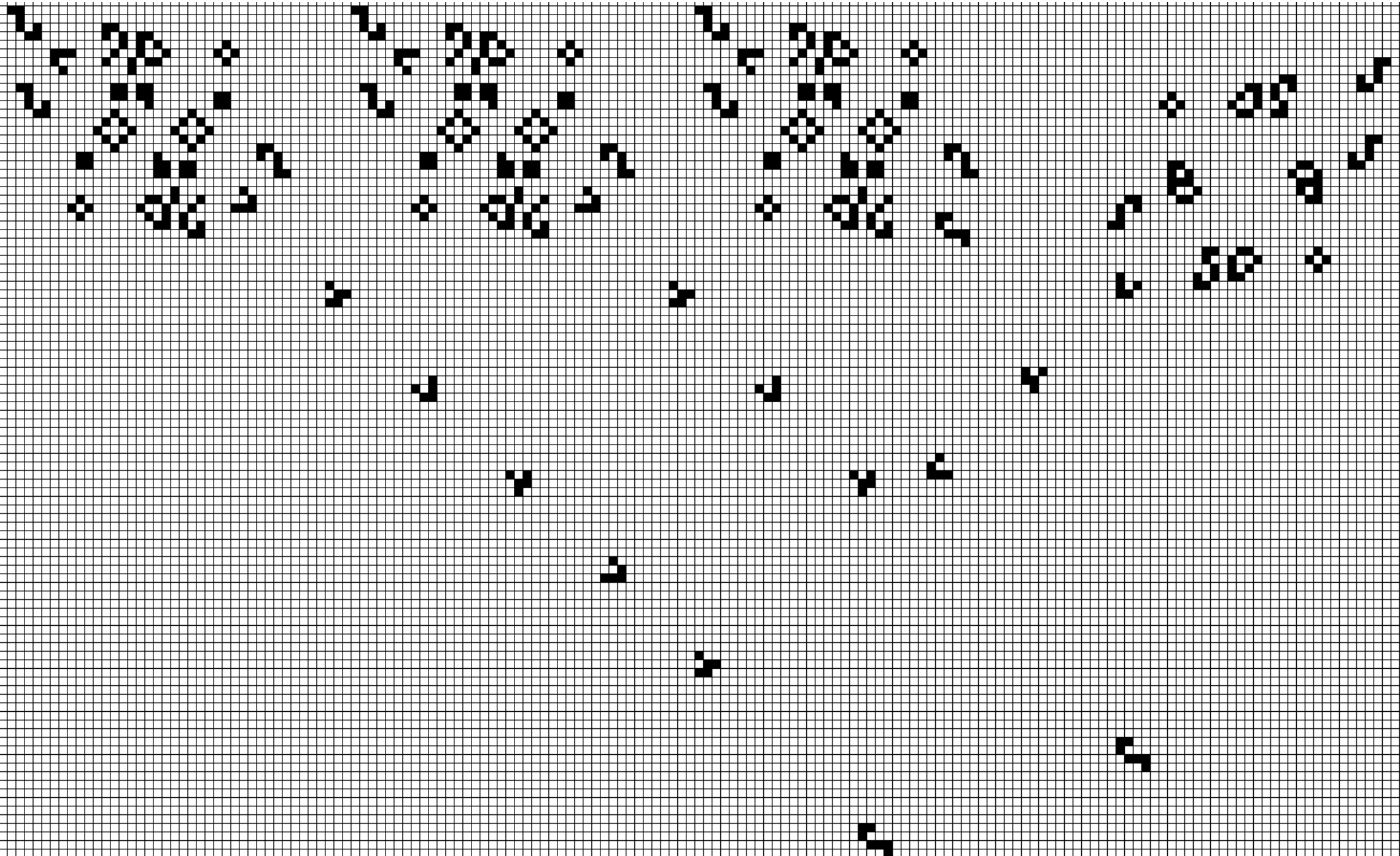
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

303/414

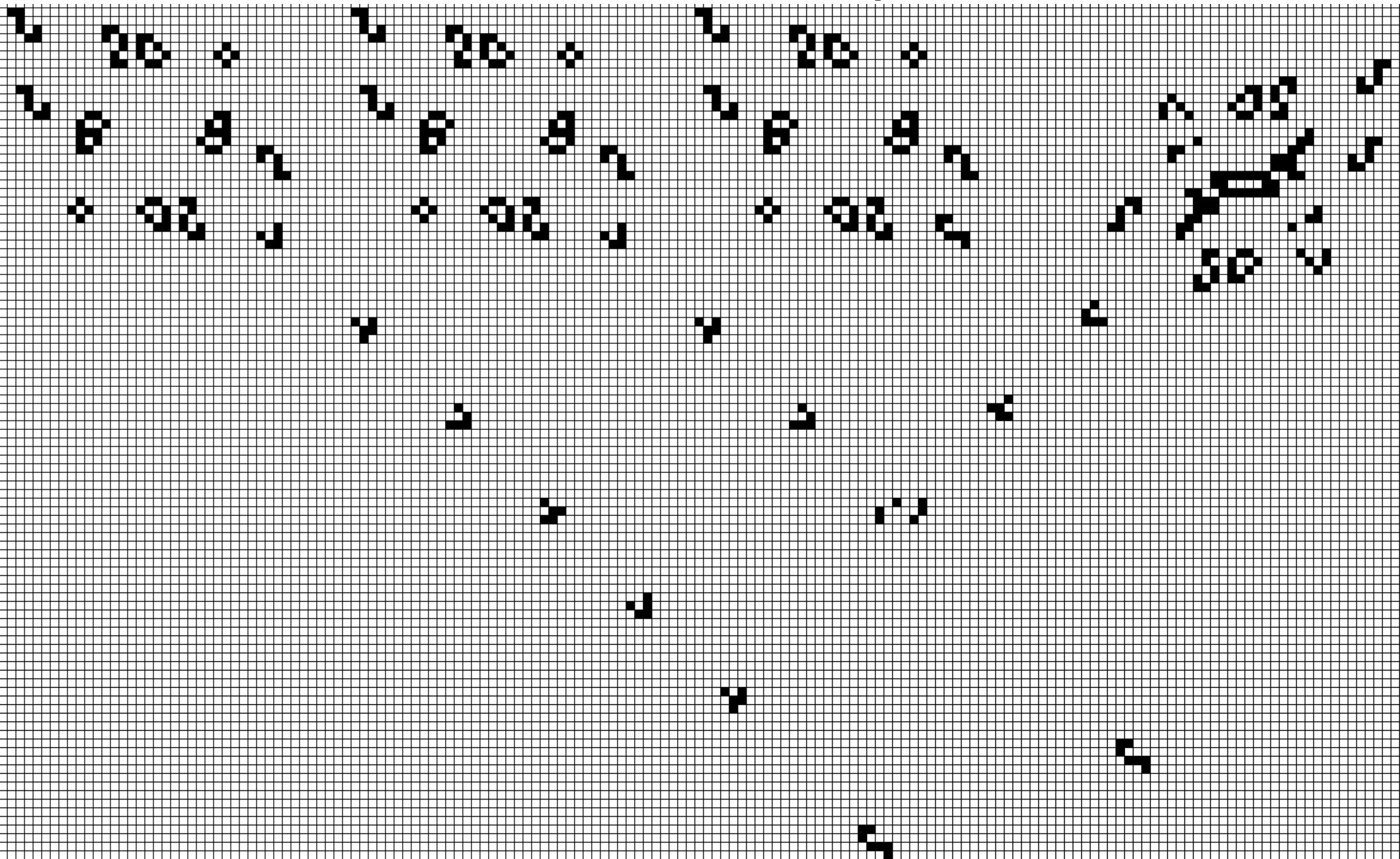
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

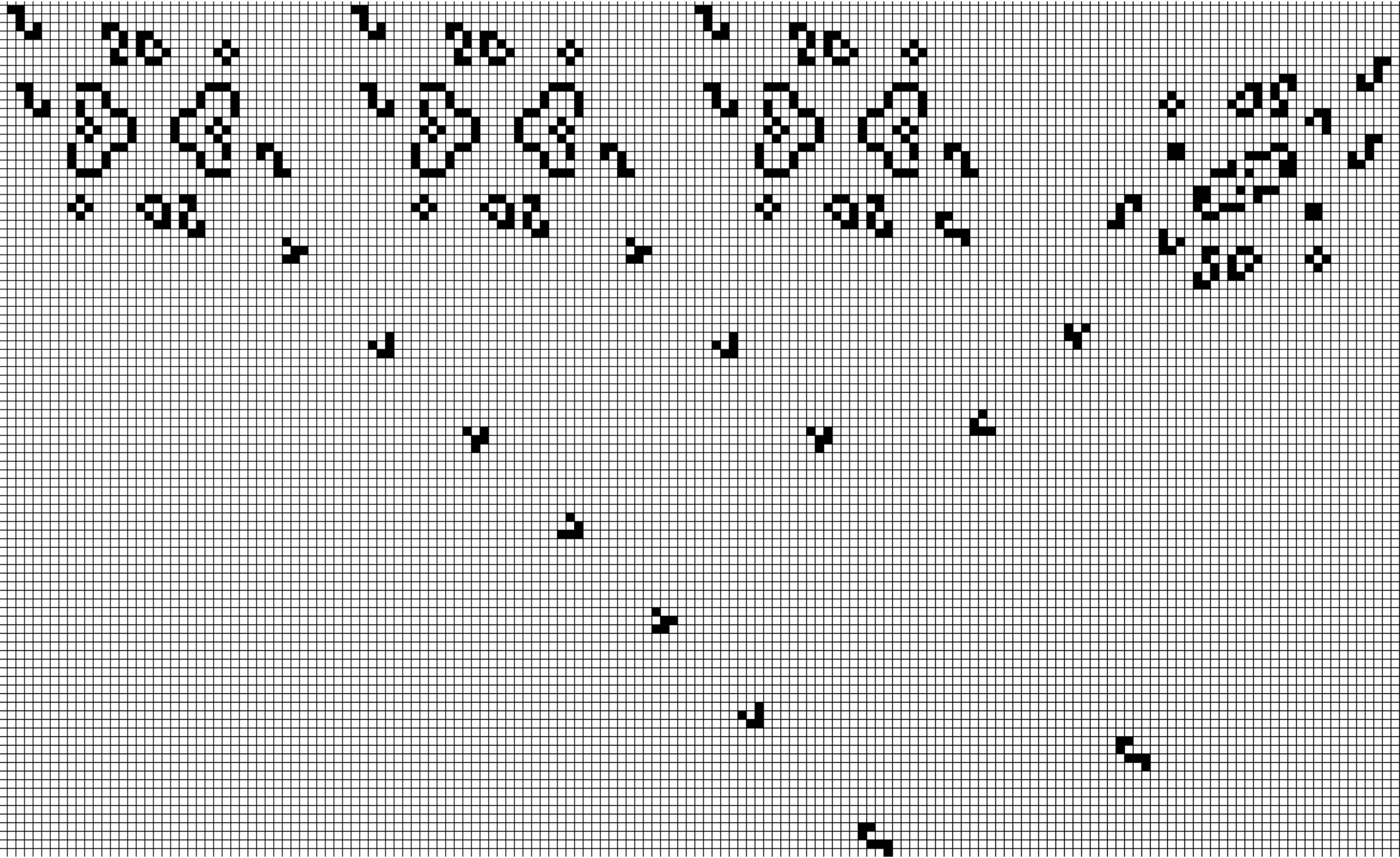
304/414

II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

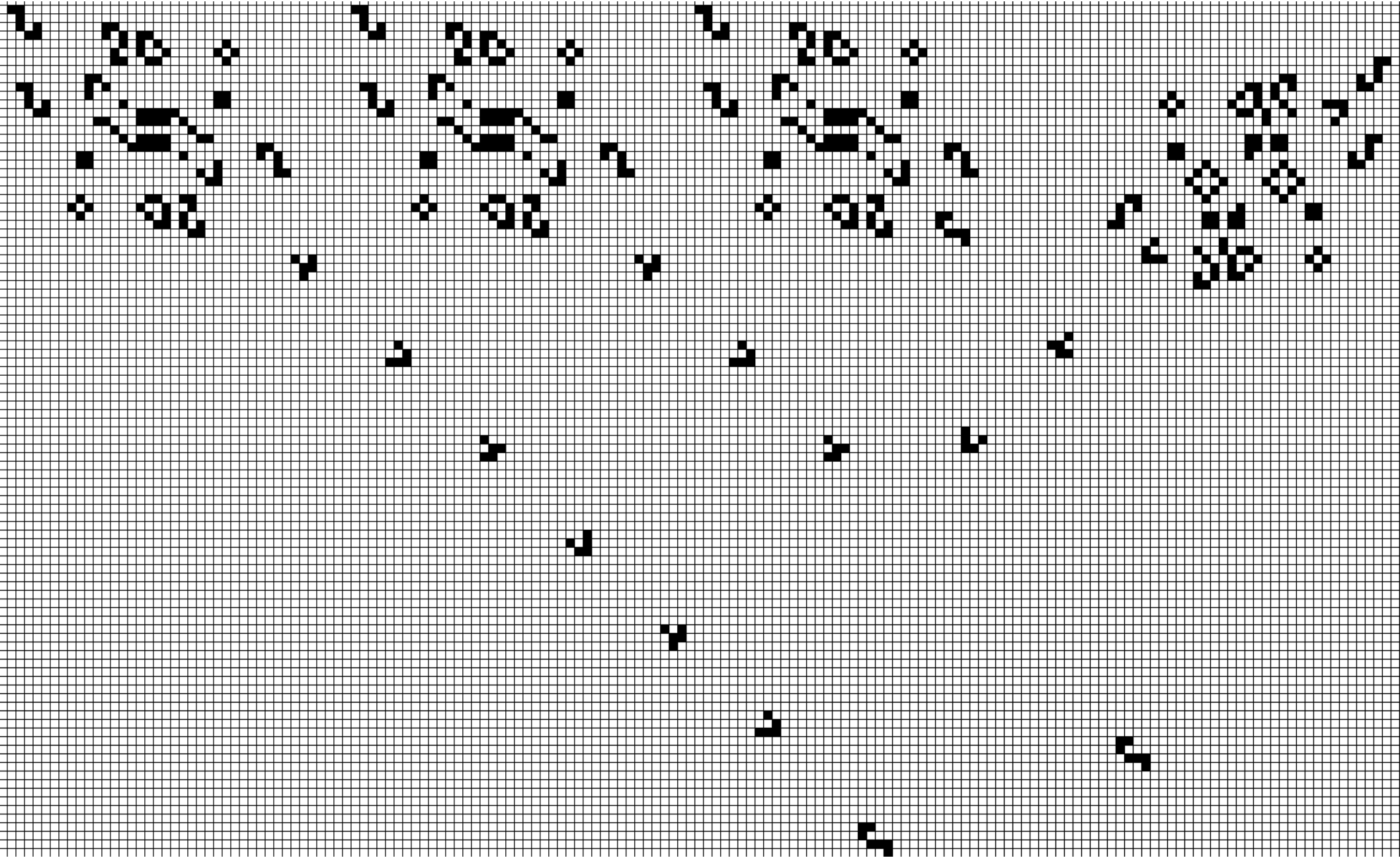
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

306/414

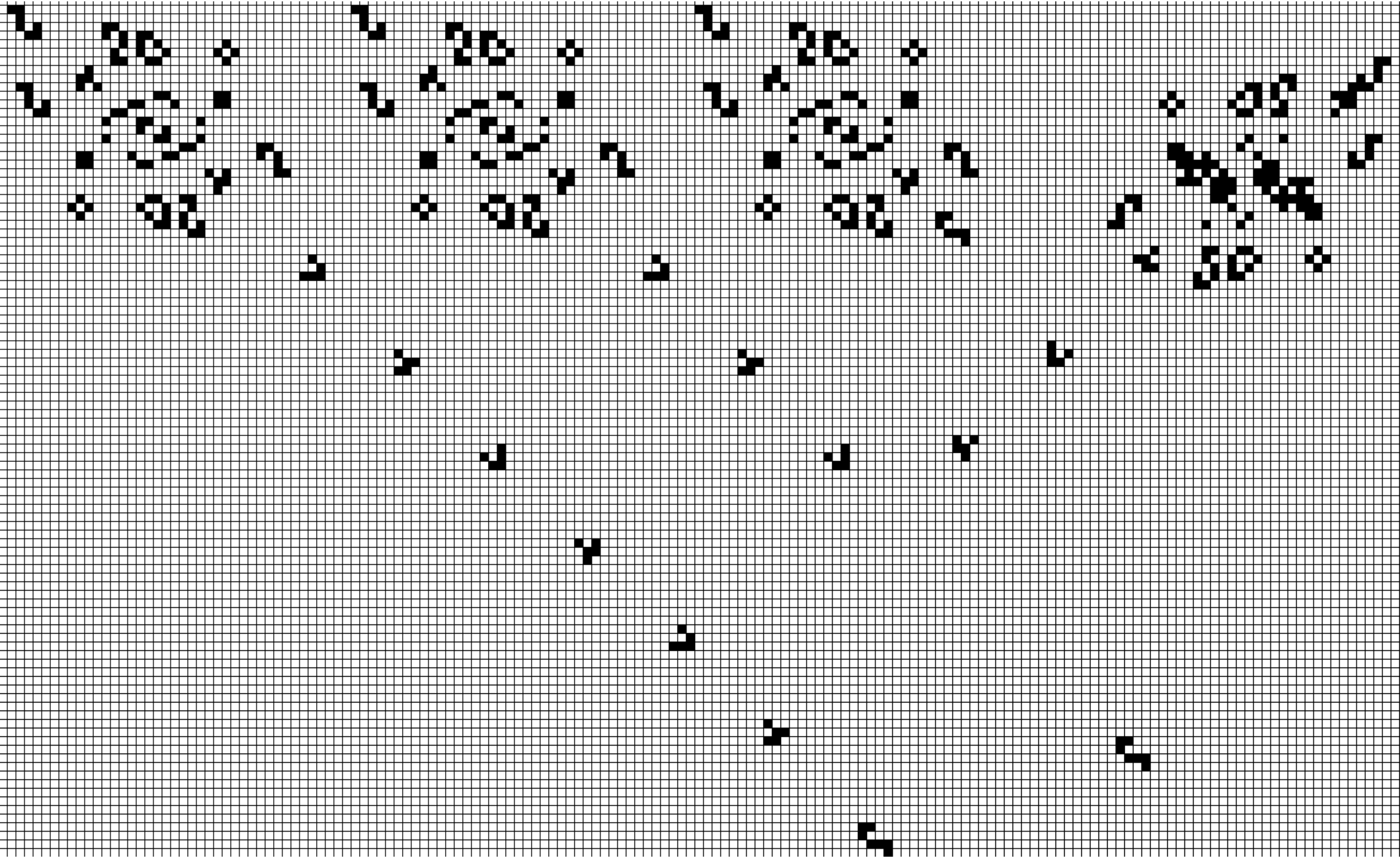
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

307/414

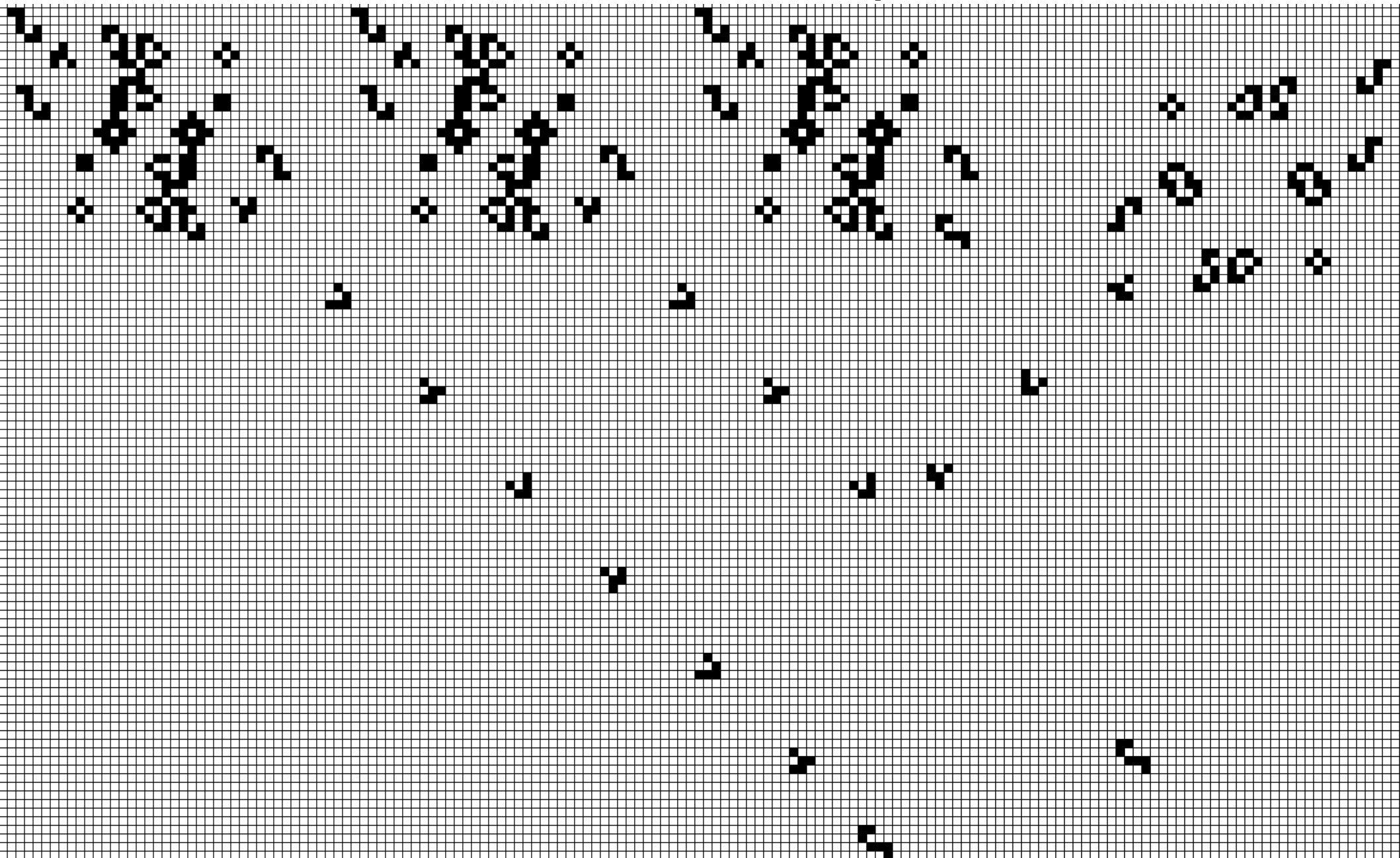
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

308/414

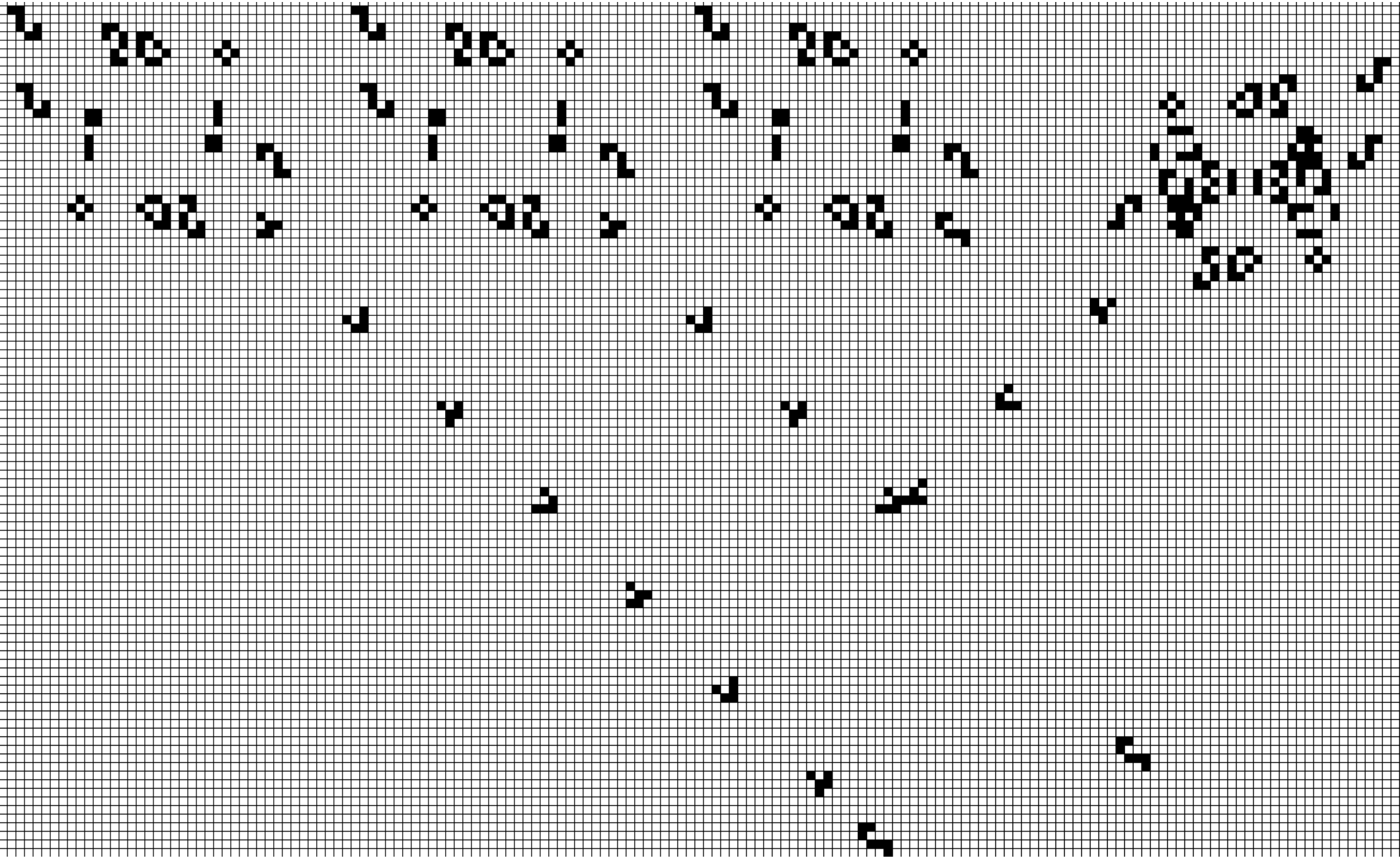
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

309/414

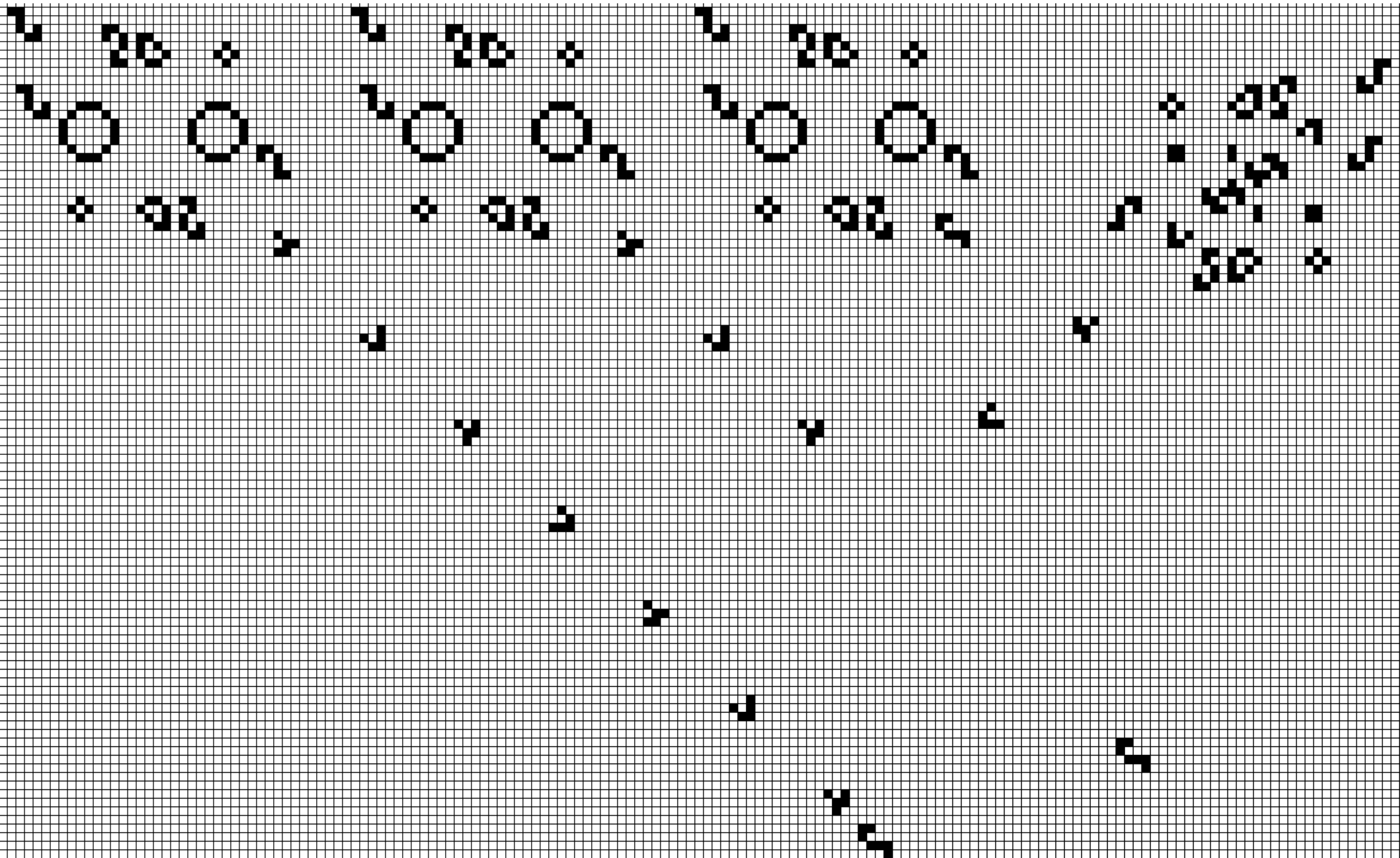
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

310/414

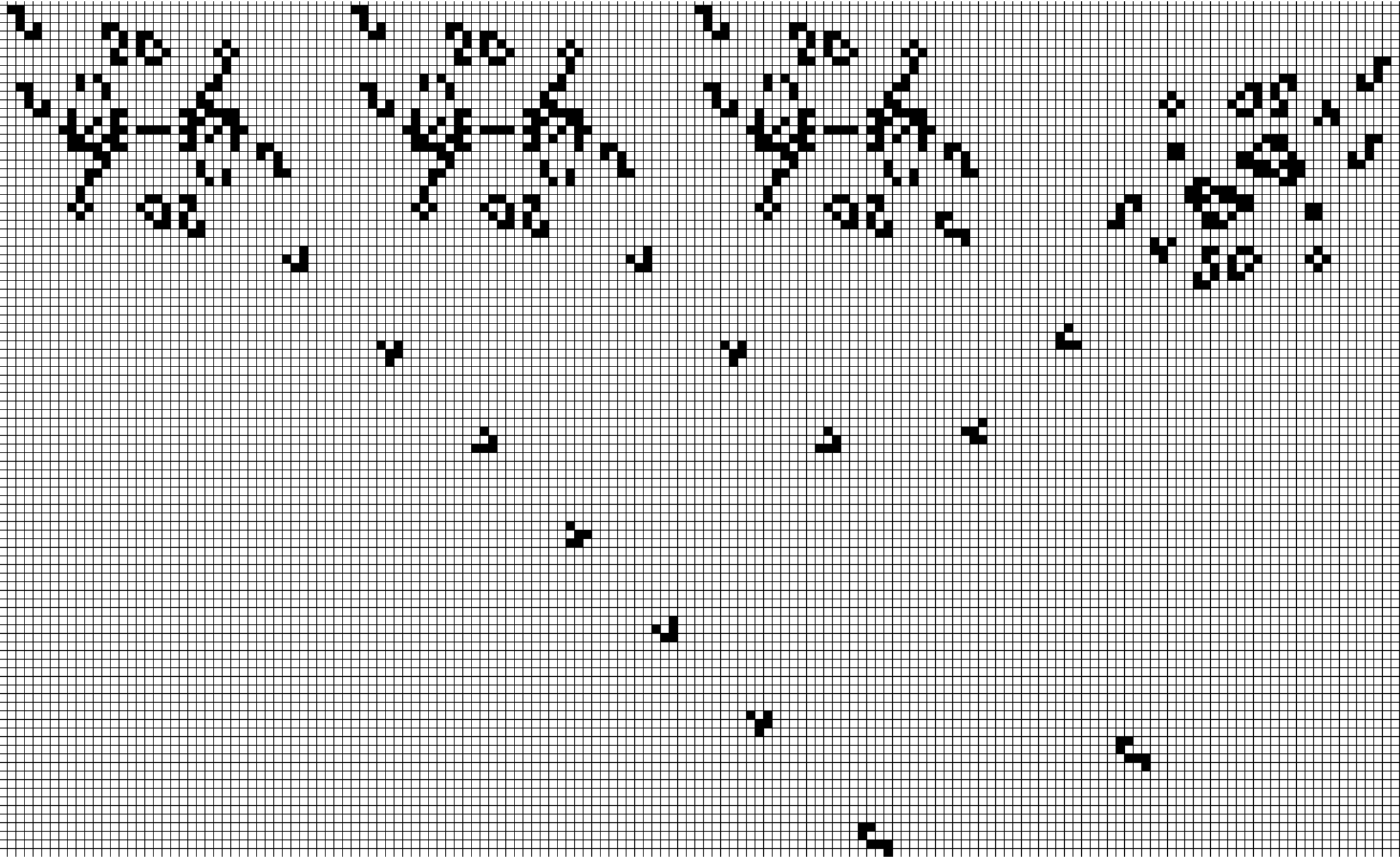
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

311/414

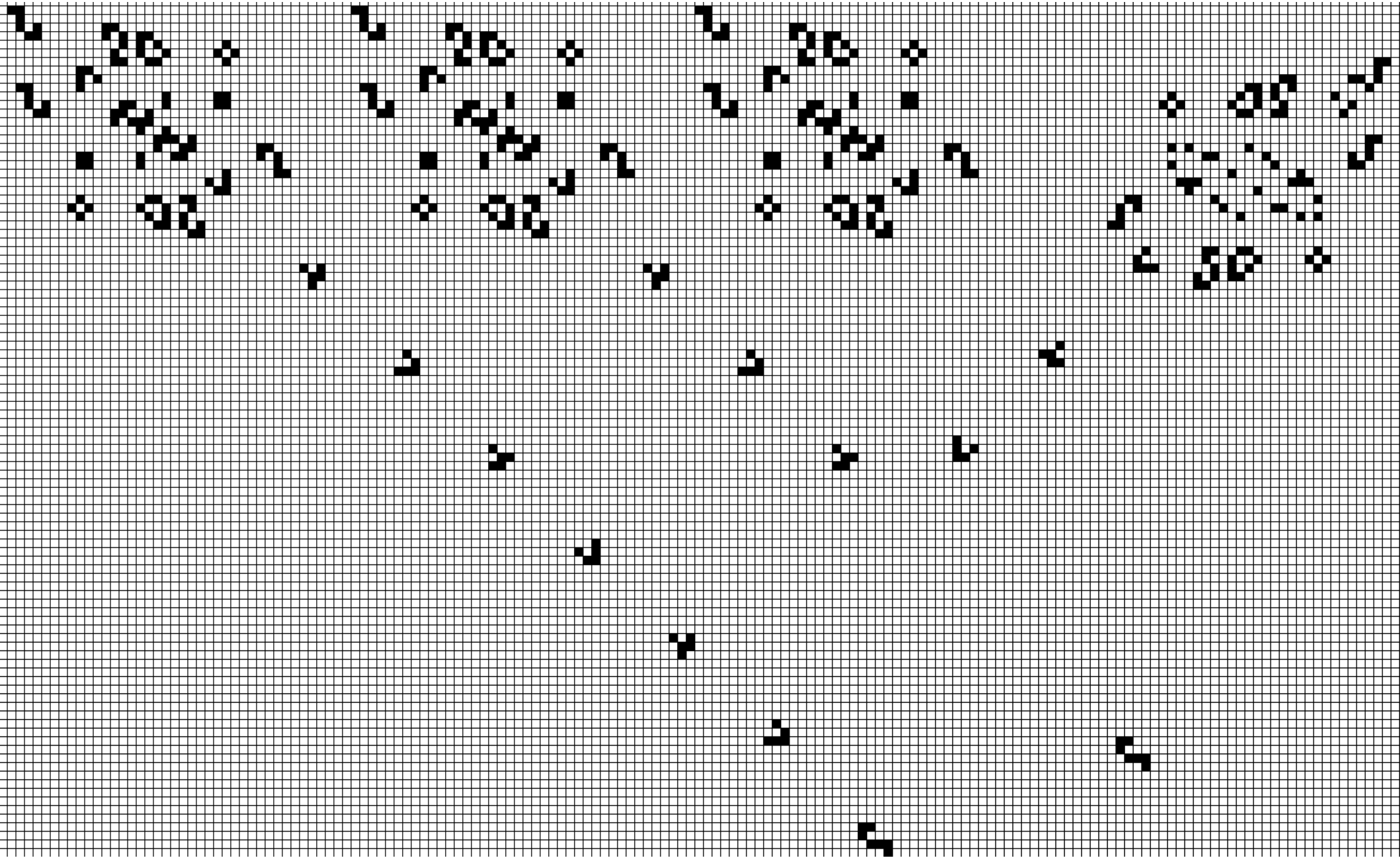
II. Outils nécessaires et composition du filtre



OR : Entrée 1 0

312/414

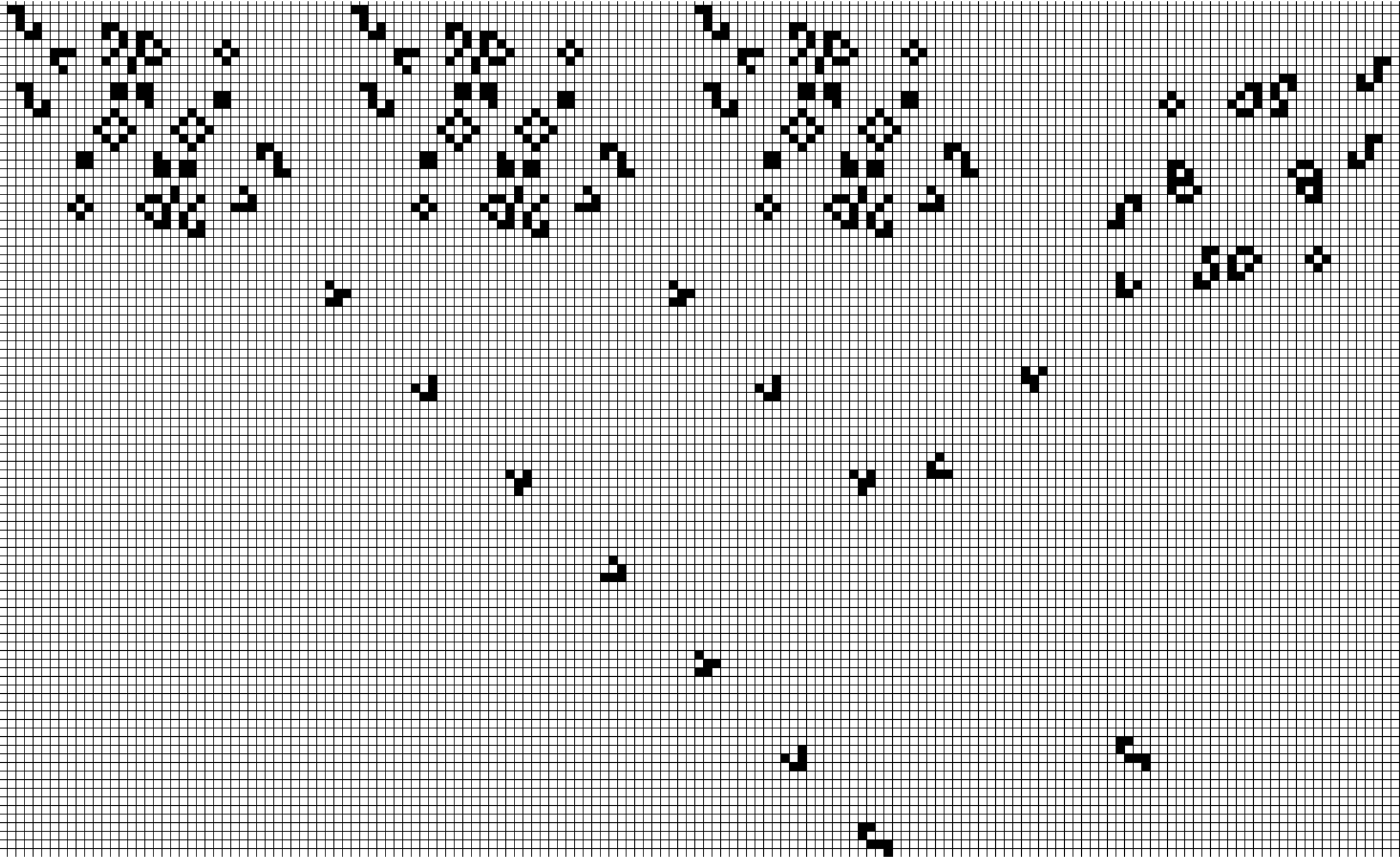
II. Outils nécessaires et composition du filtre



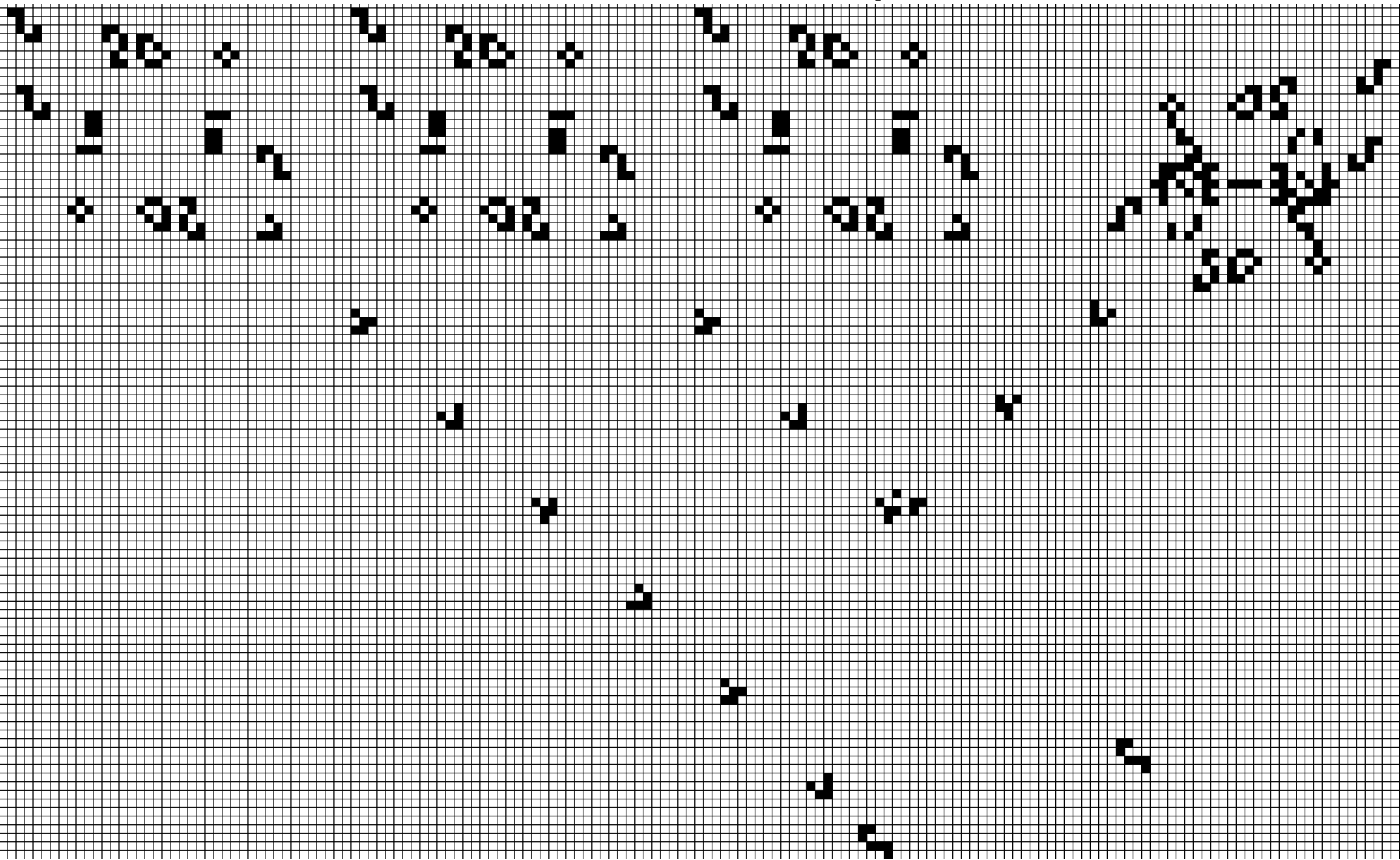
OR : Entrée 1 1

313/414

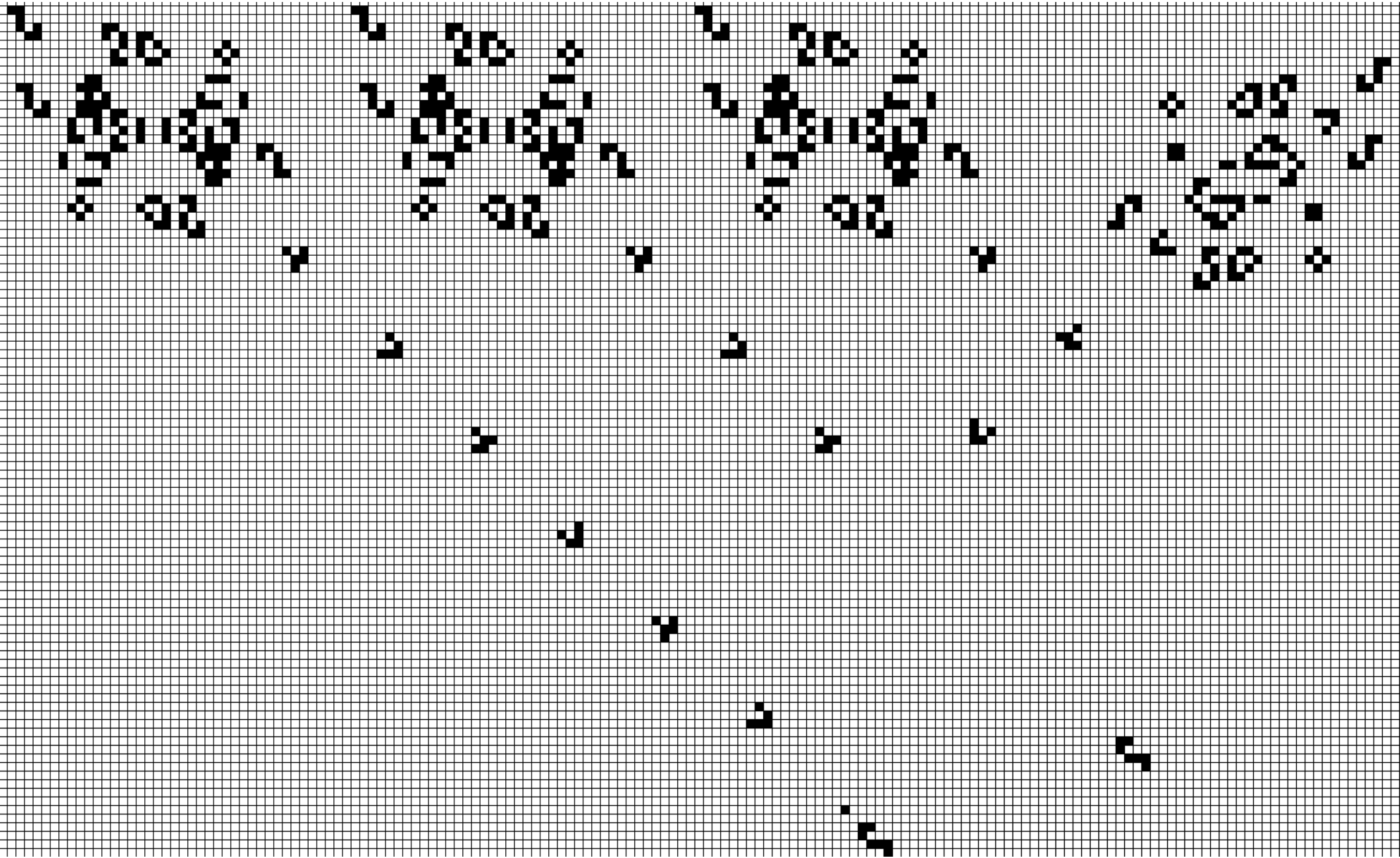
II. Outils nécessaires et composition du filtre



II. Outils nécessaires et composition du filtre



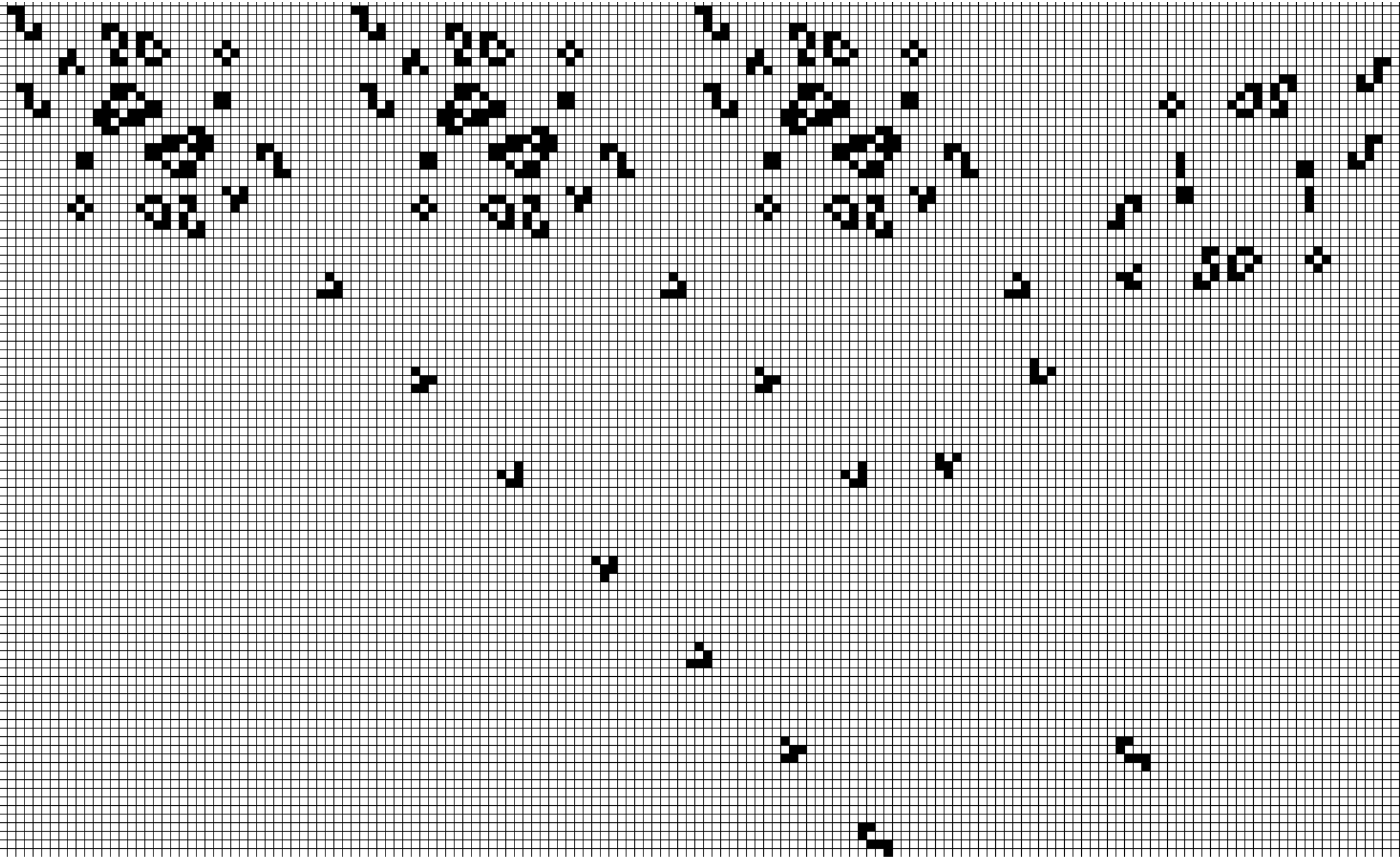
II. Outils nécessaires et composition du filtre



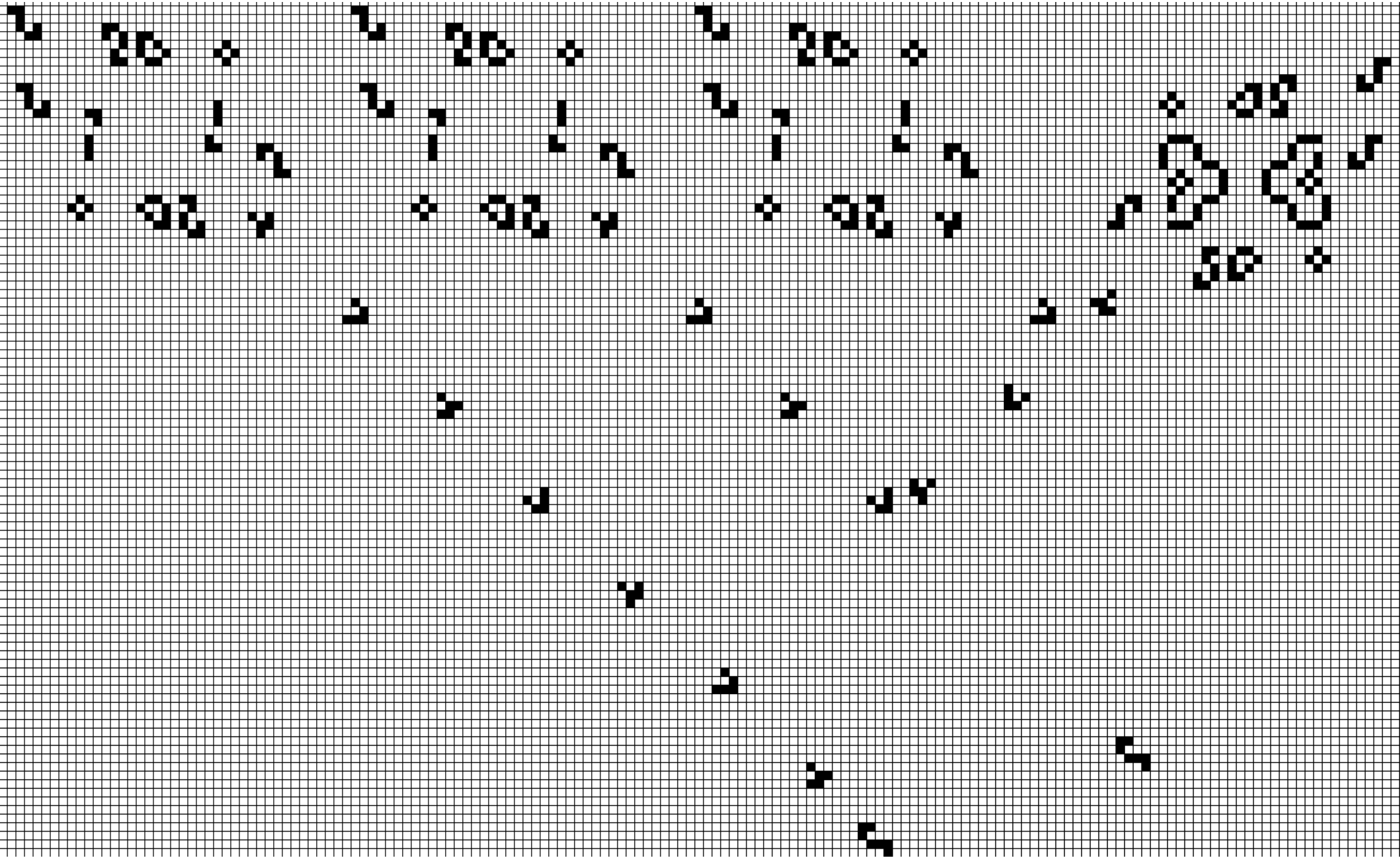
OR : Entrée 1 1

316/414

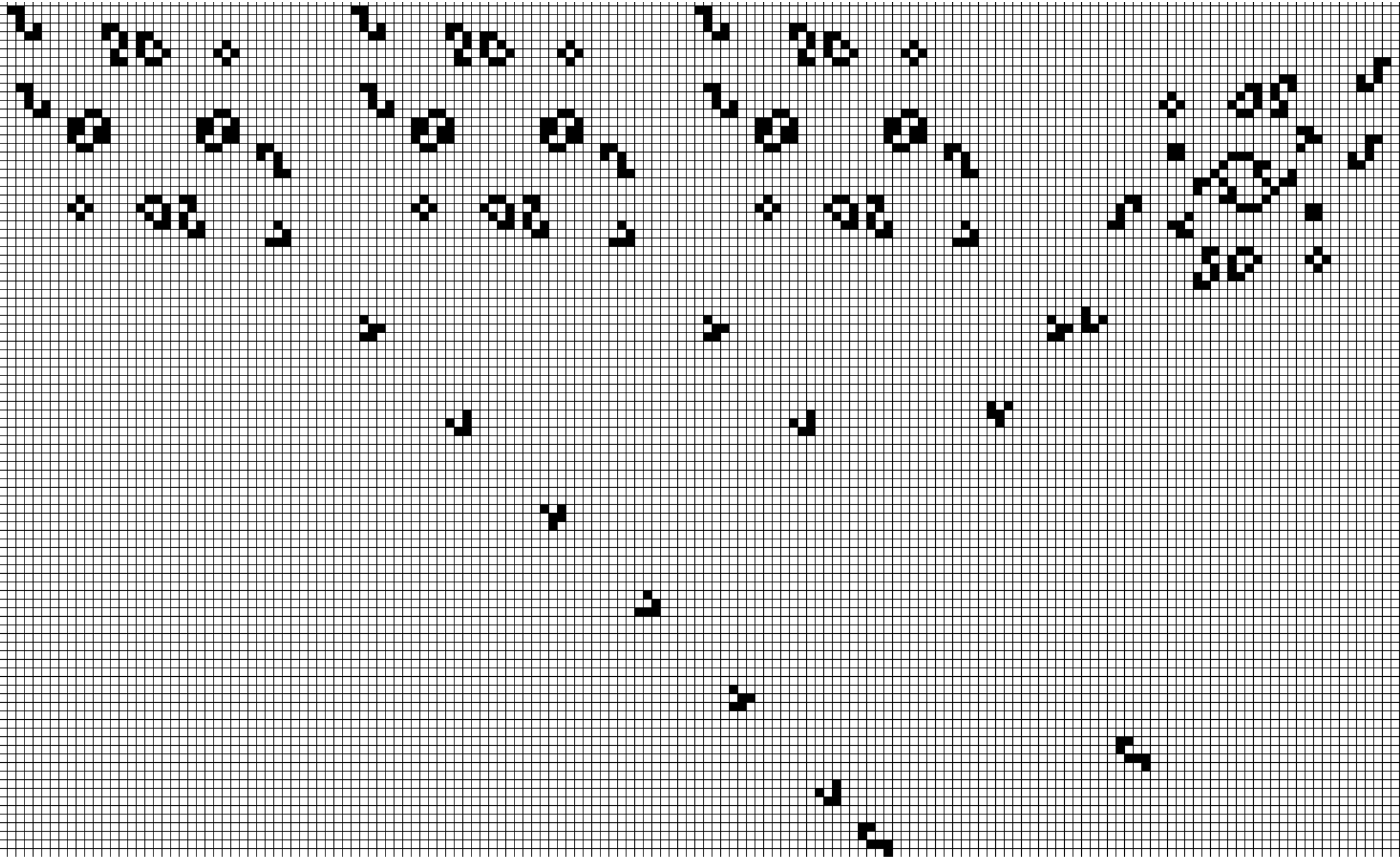
II. Outils nécessaires et composition du filtre



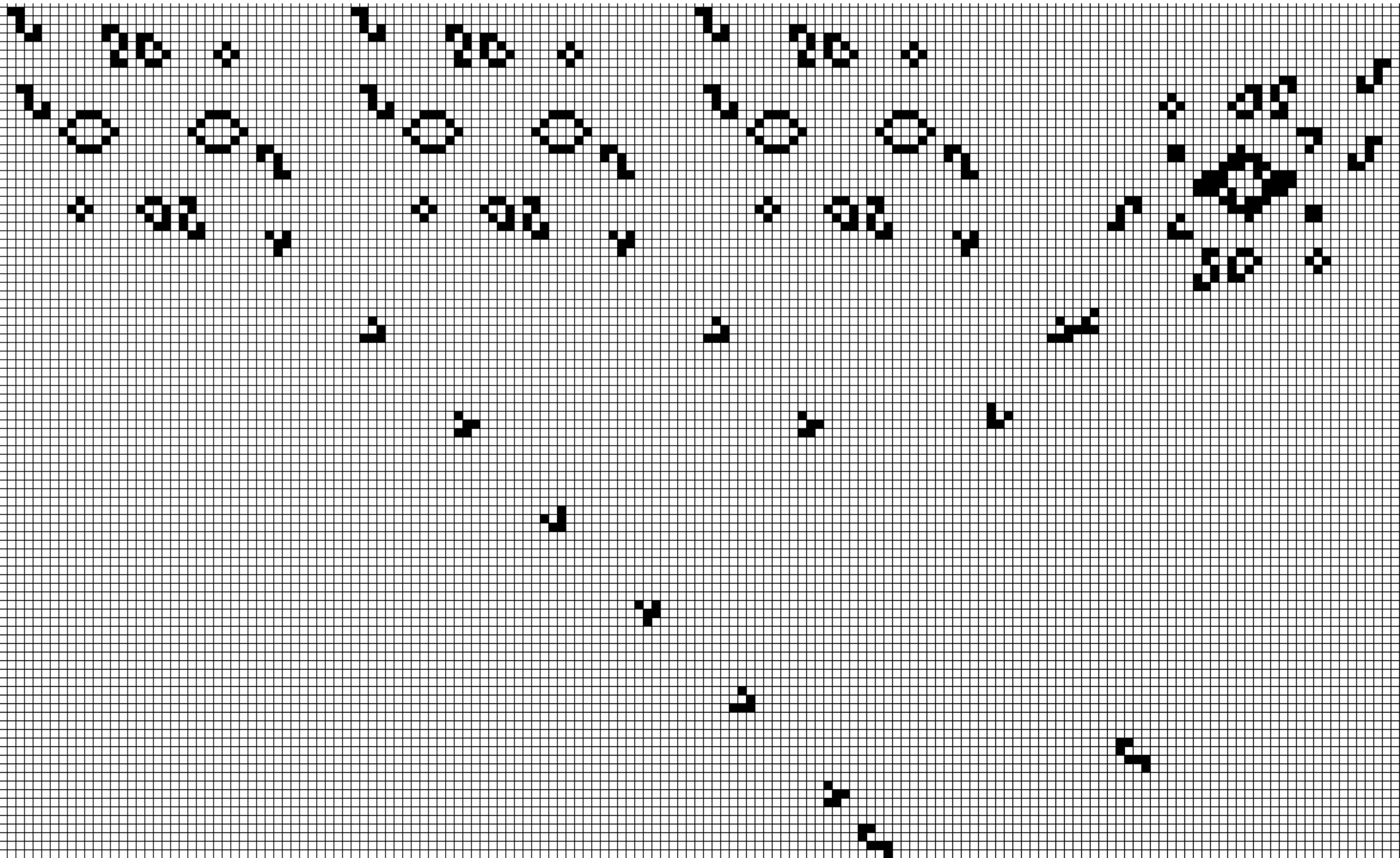
II. Outils nécessaires et composition du filtre



II. Outils nécessaires et composition du filtre



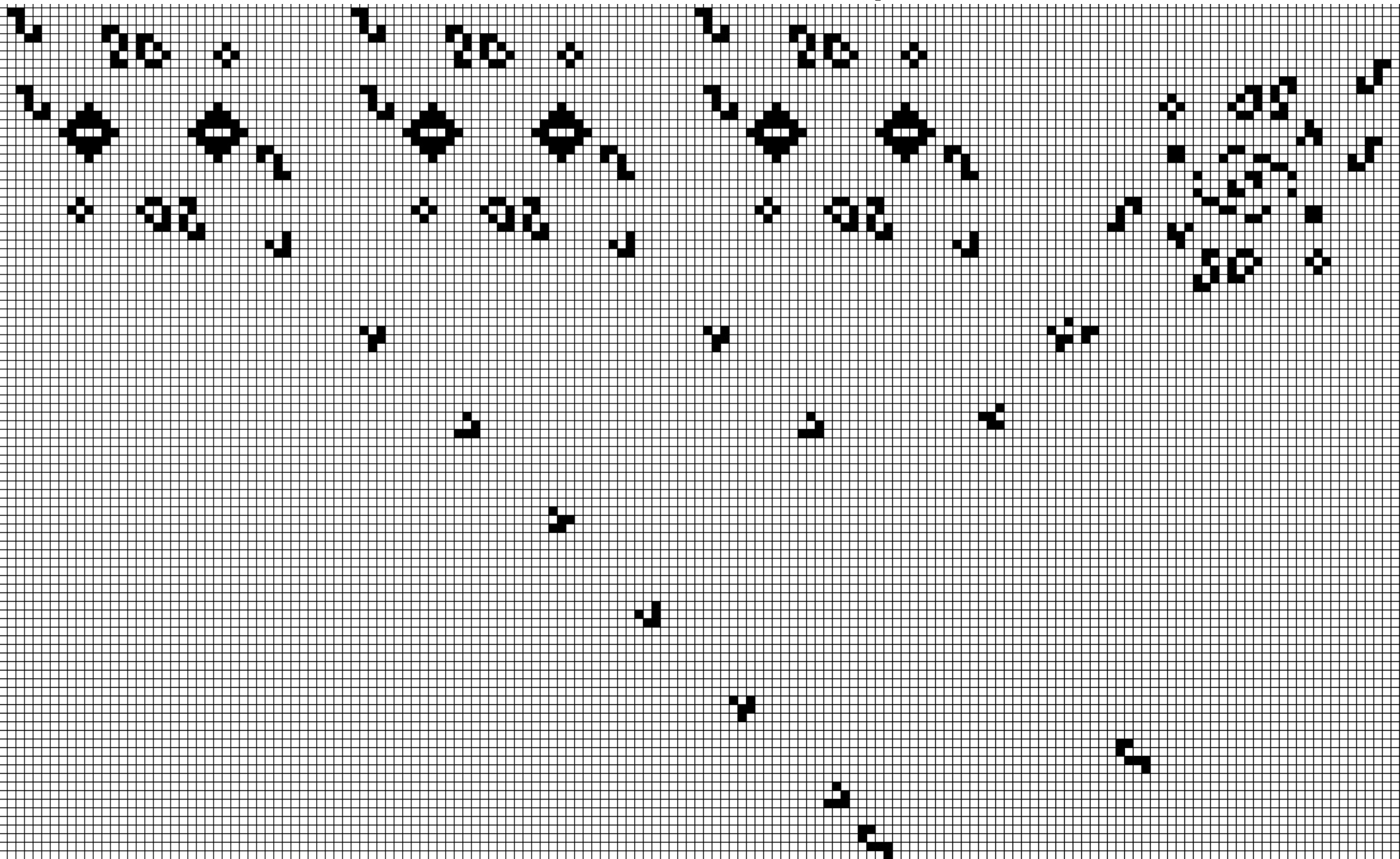
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

320/414

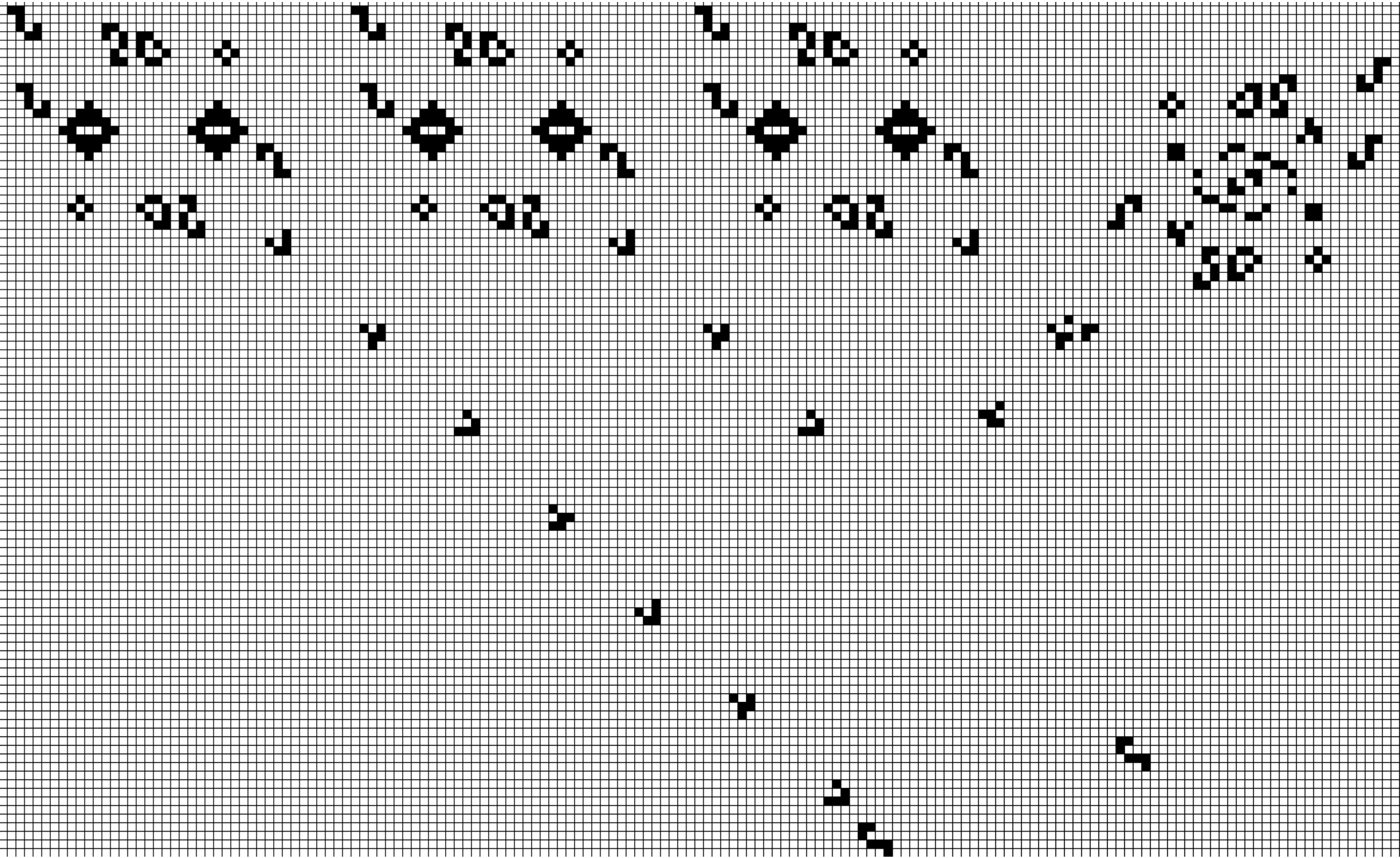
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

321/414

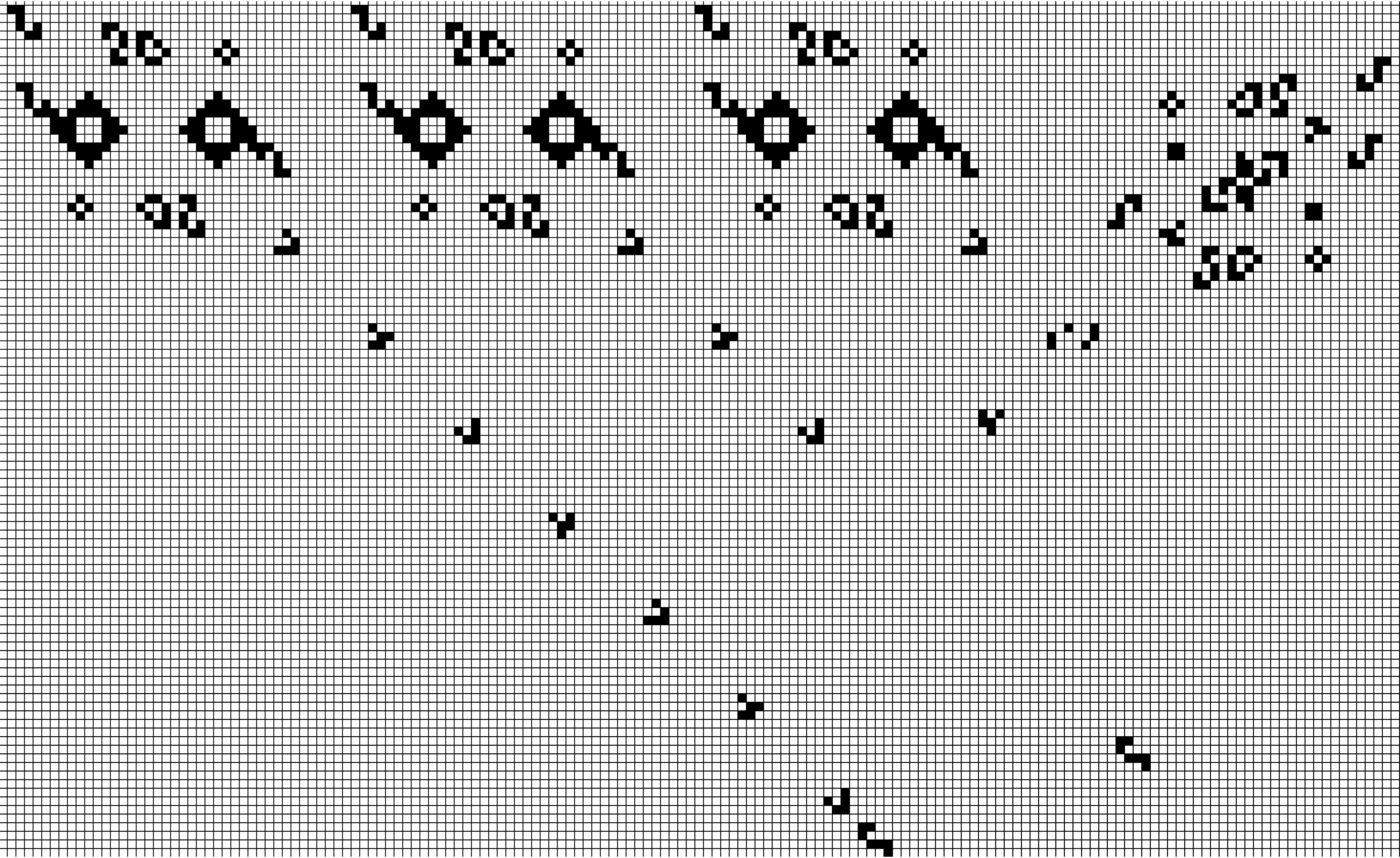
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

322/414

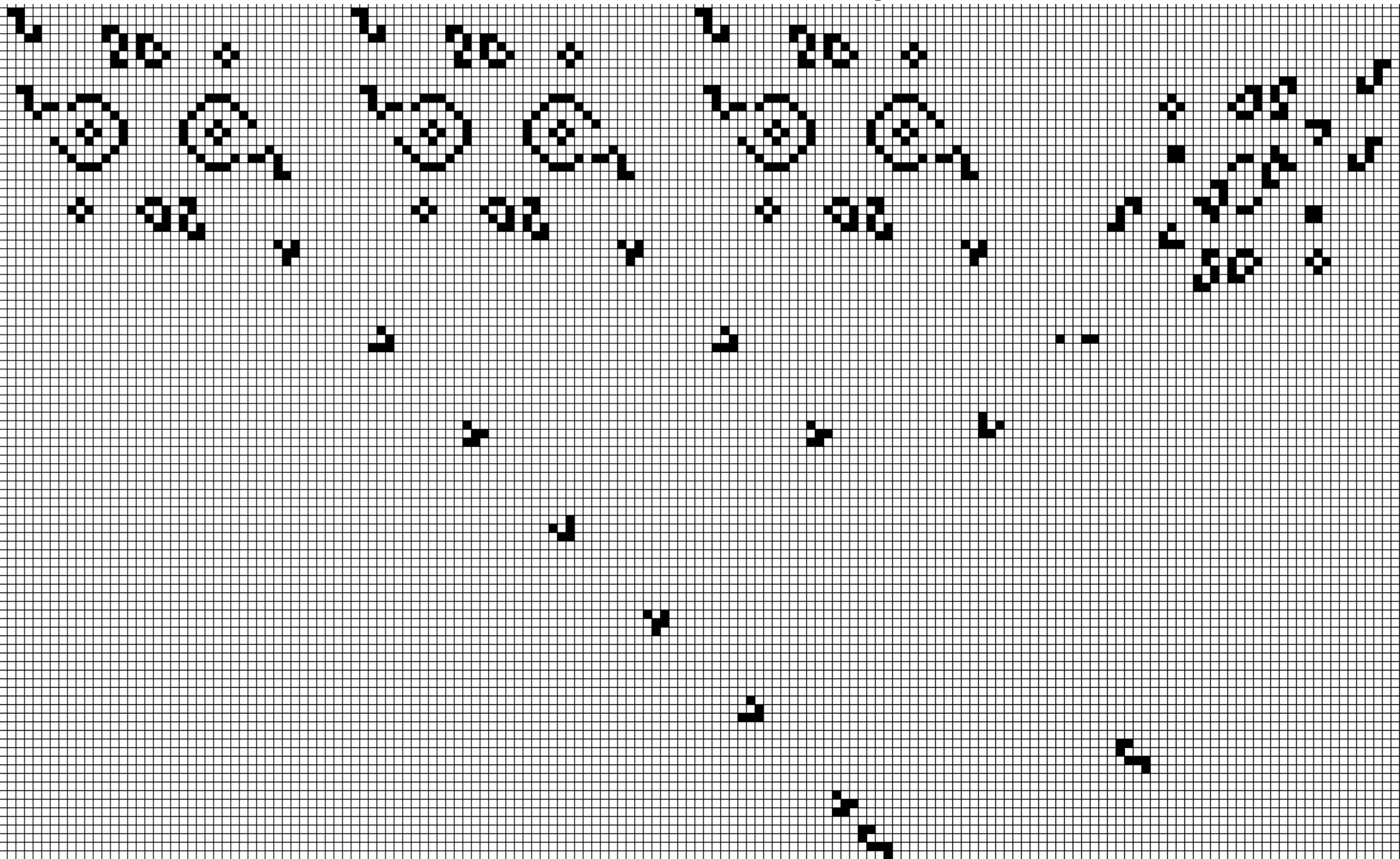
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

323/414

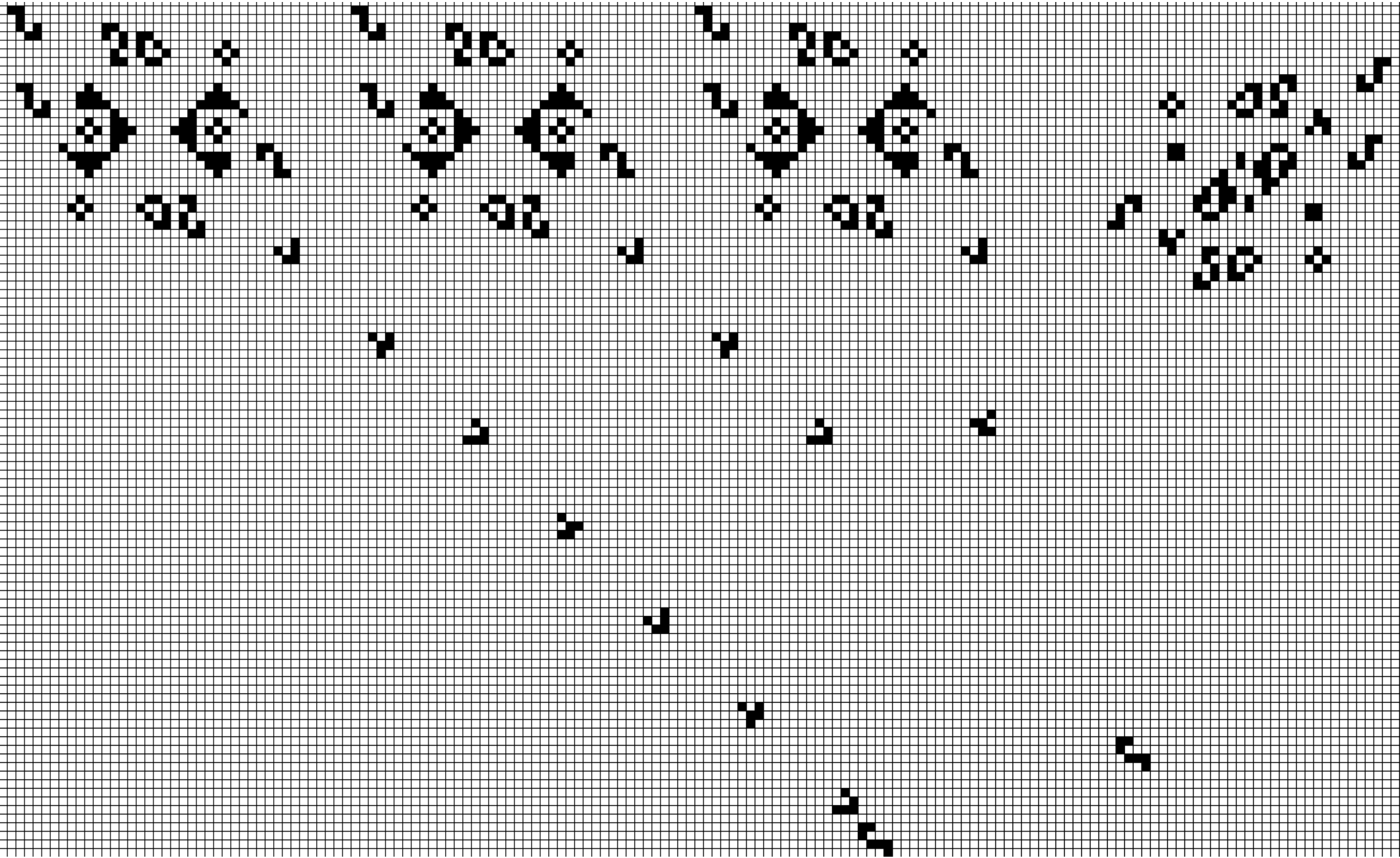
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

324/414

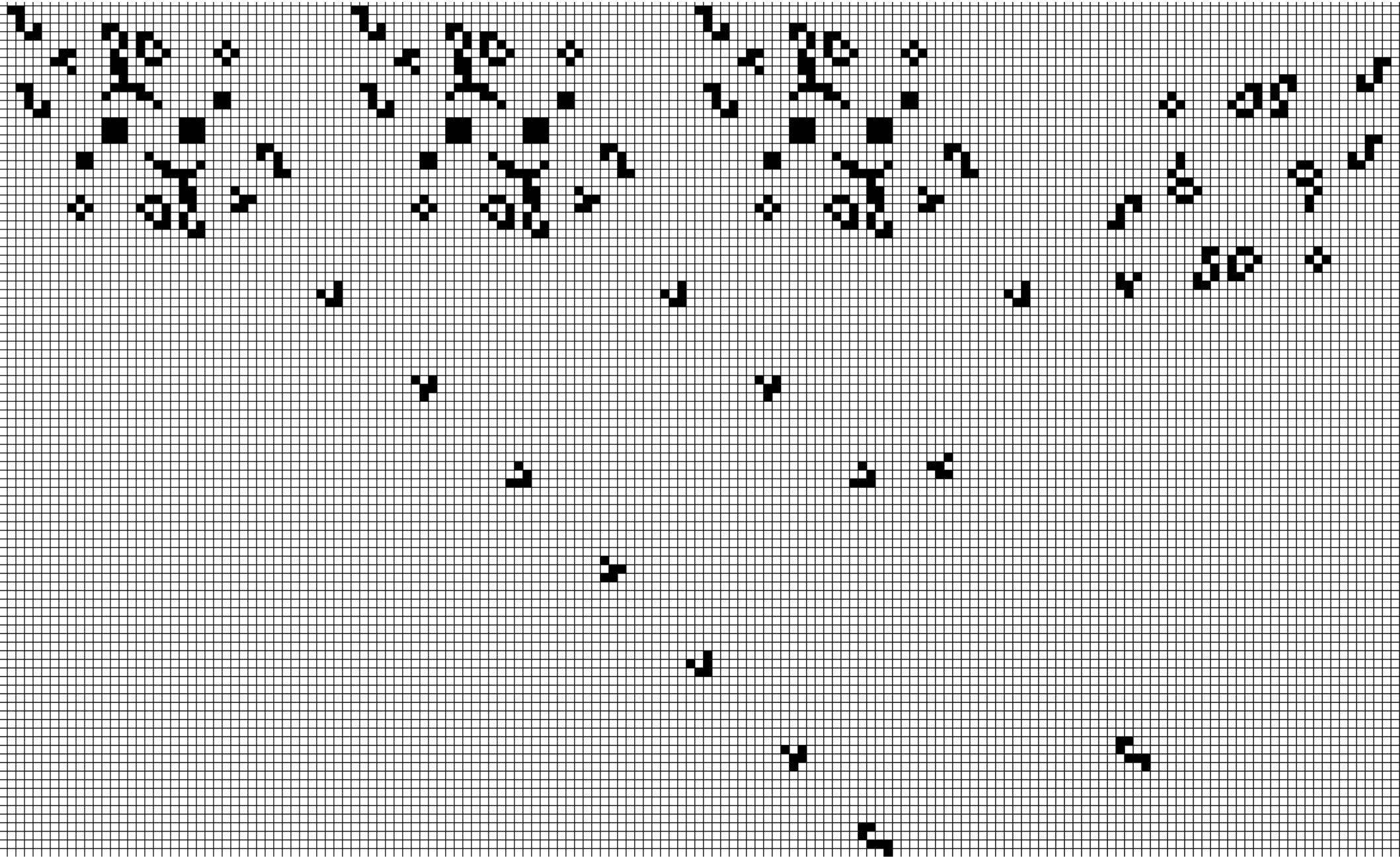
II. Outils nécessaires et composition du filtre



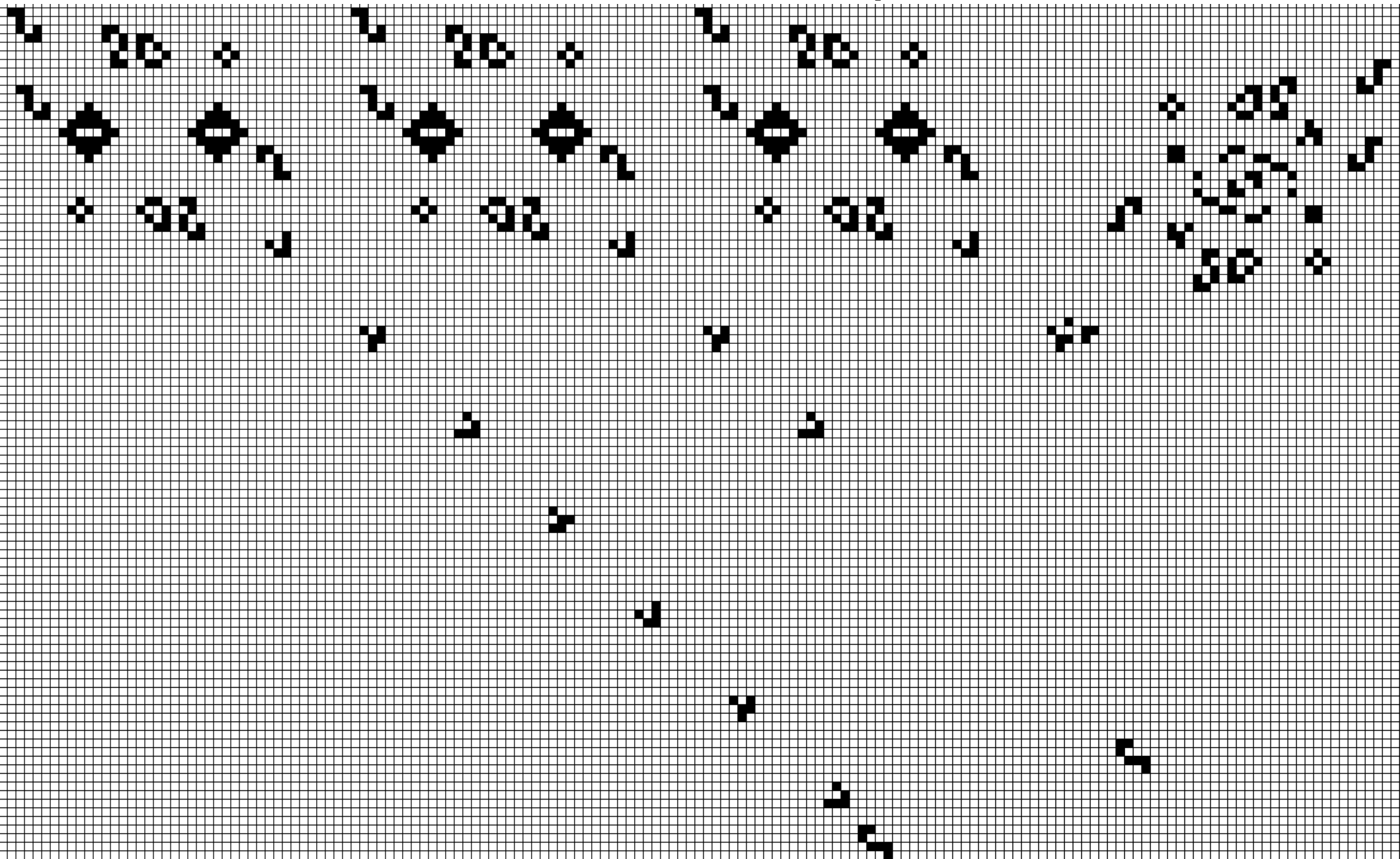
OR : Entrée 1 1

325/414

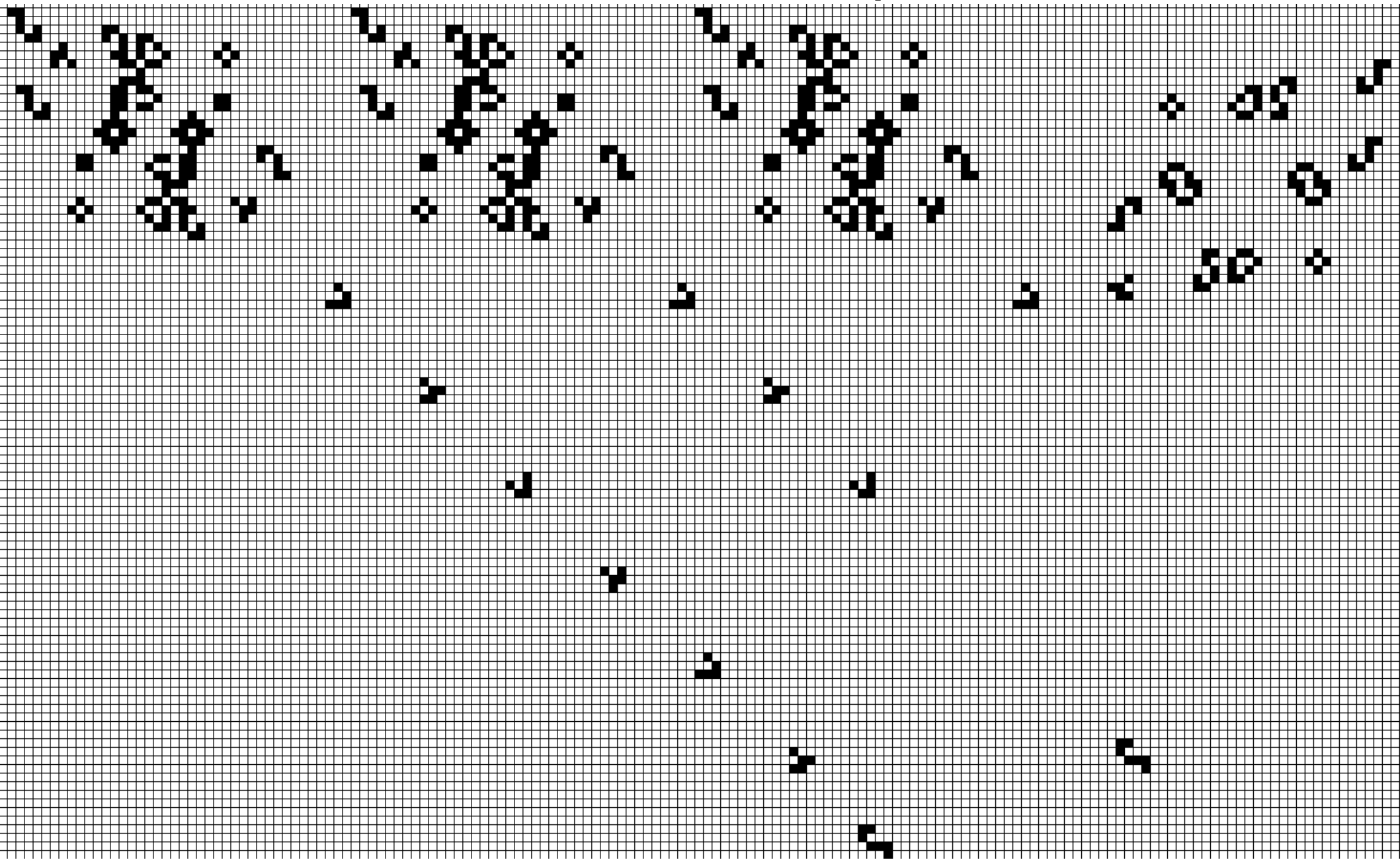
II. Outils nécessaires et composition du filtre



II. Outils nécessaires et composition du filtre



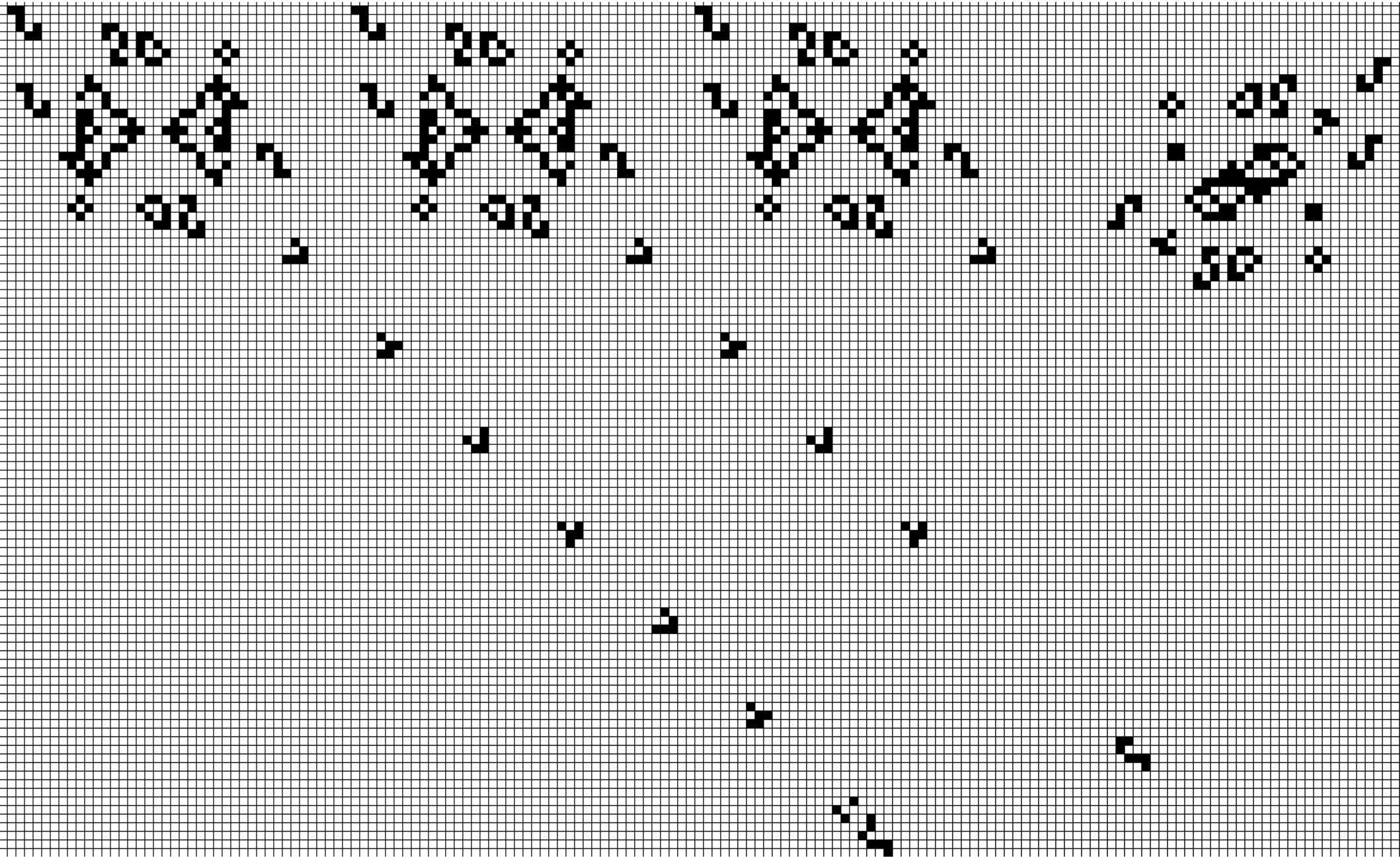
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

328/414

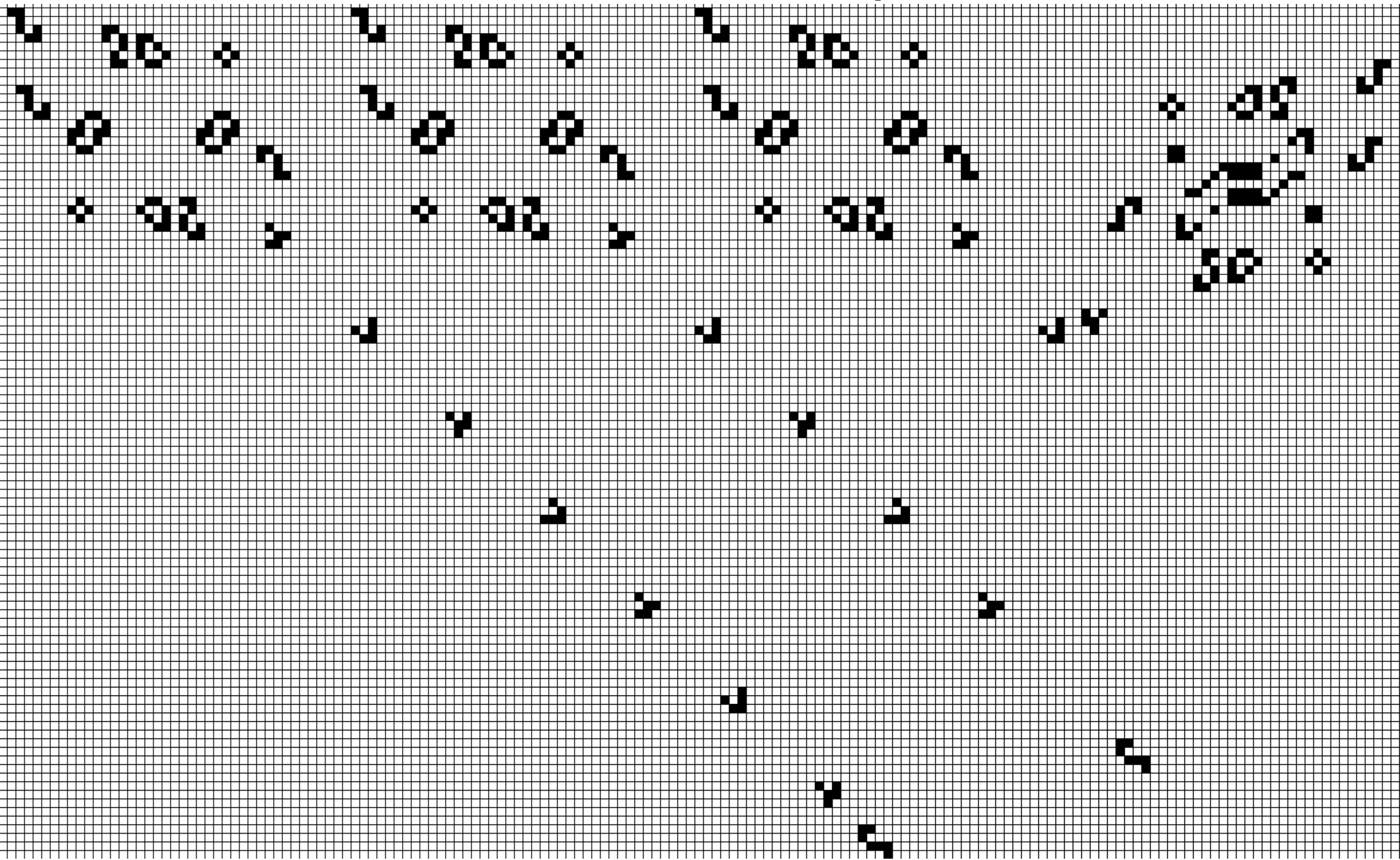
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

329/414

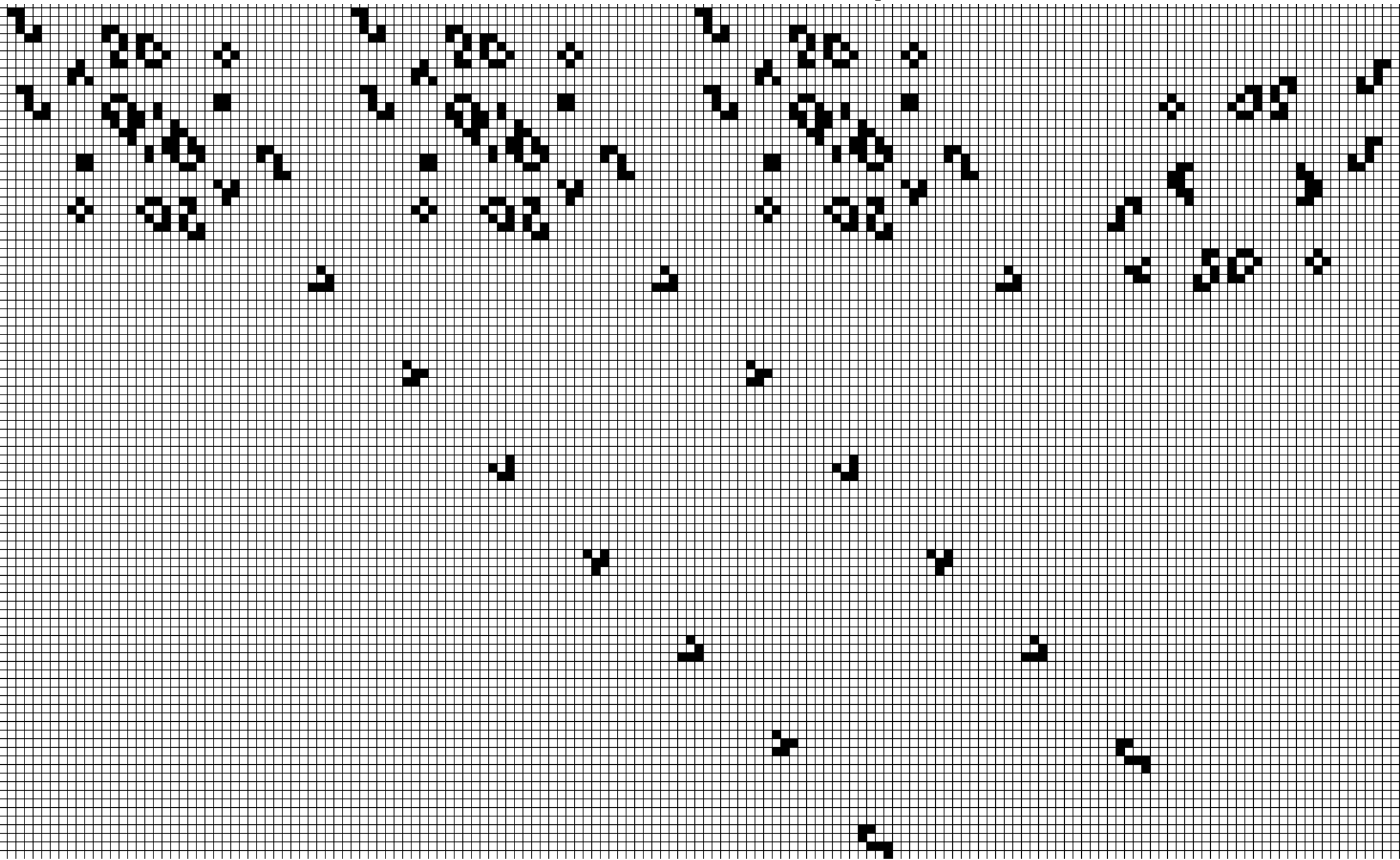
II. Outils nécessaires et composition du filtre



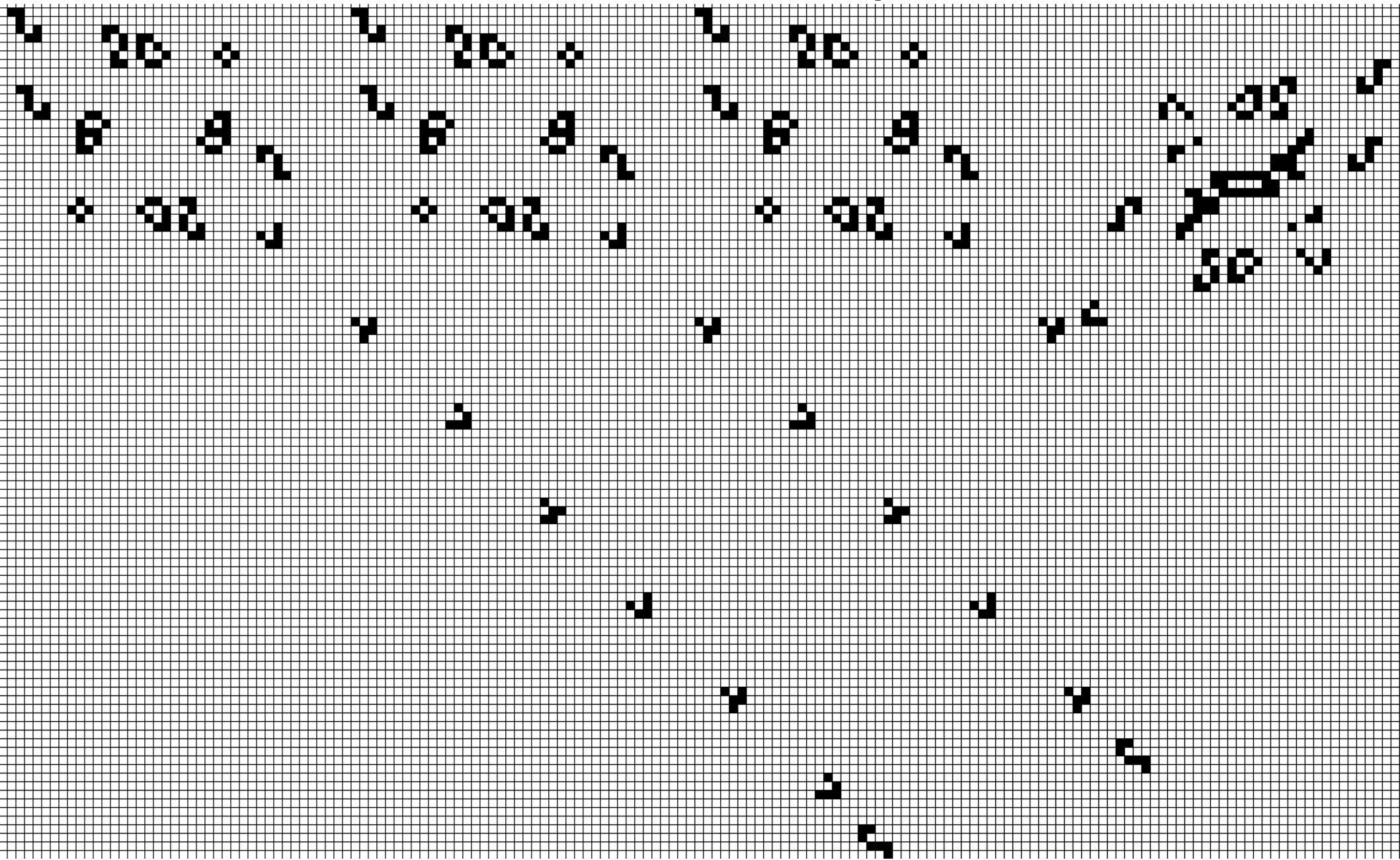
OR : Entrée 1 1

330/414

II. Outils nécessaires et composition du filtre



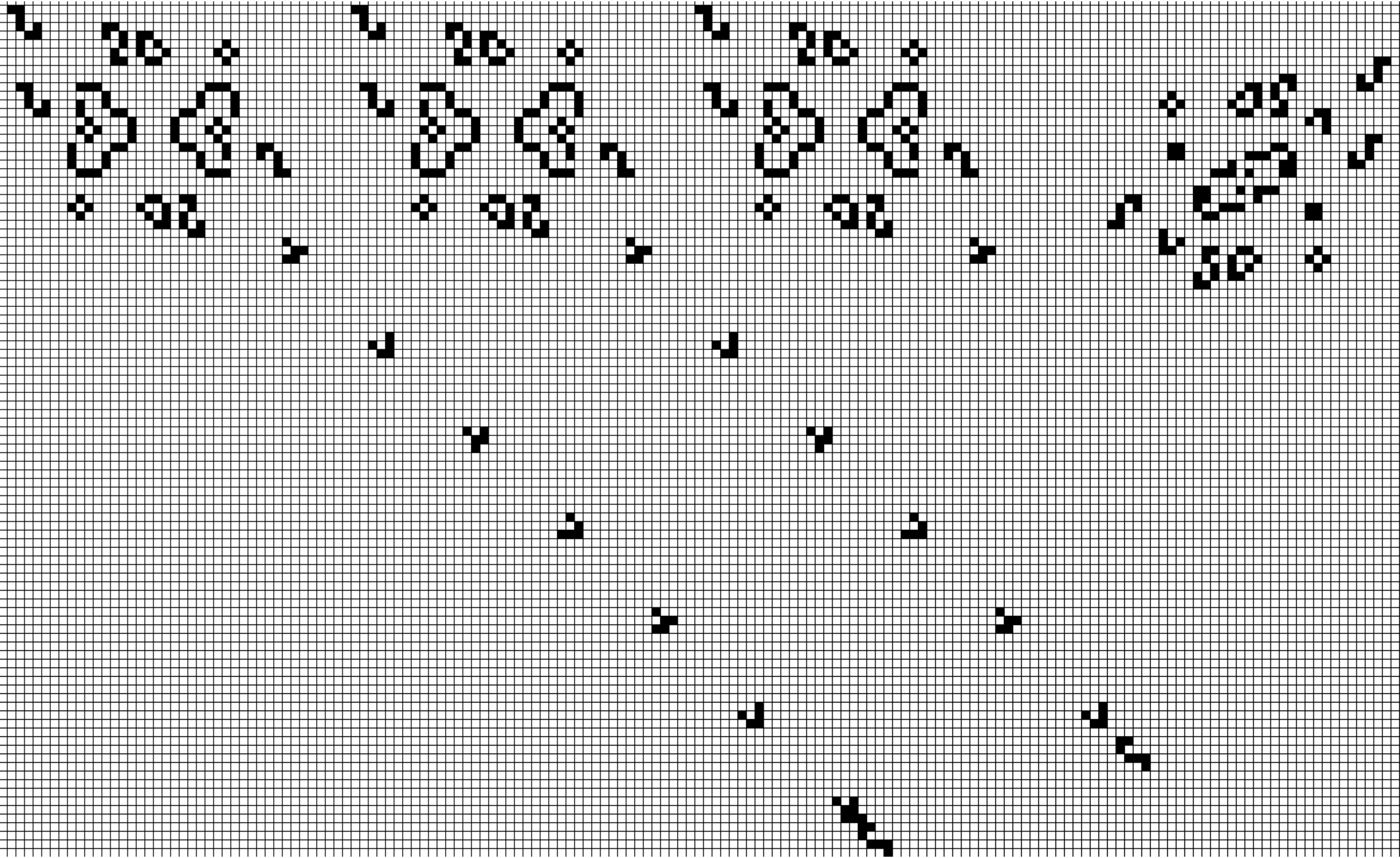
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

332/414

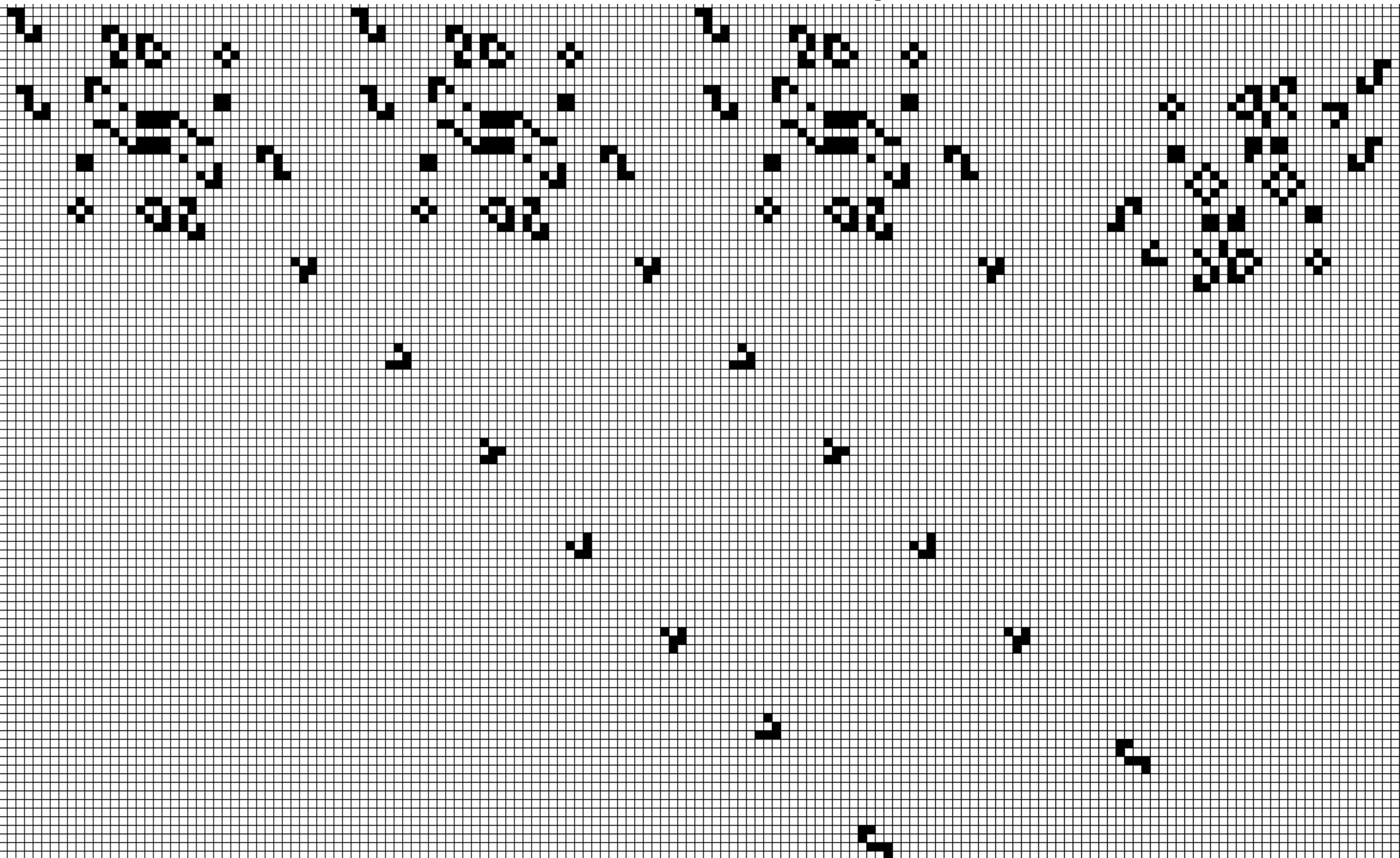
II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

333/414

II. Outils nécessaires et composition du filtre



OR : Entrée 1 1

334/414

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Semi-Additionneur :

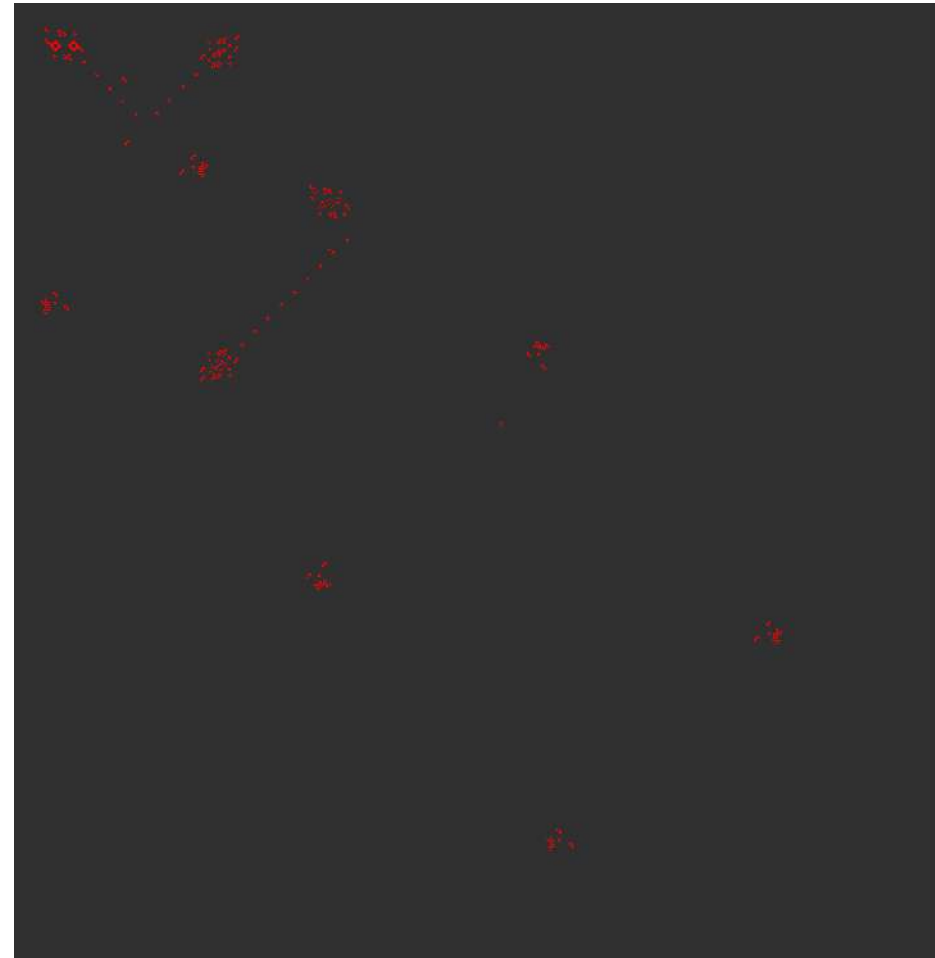
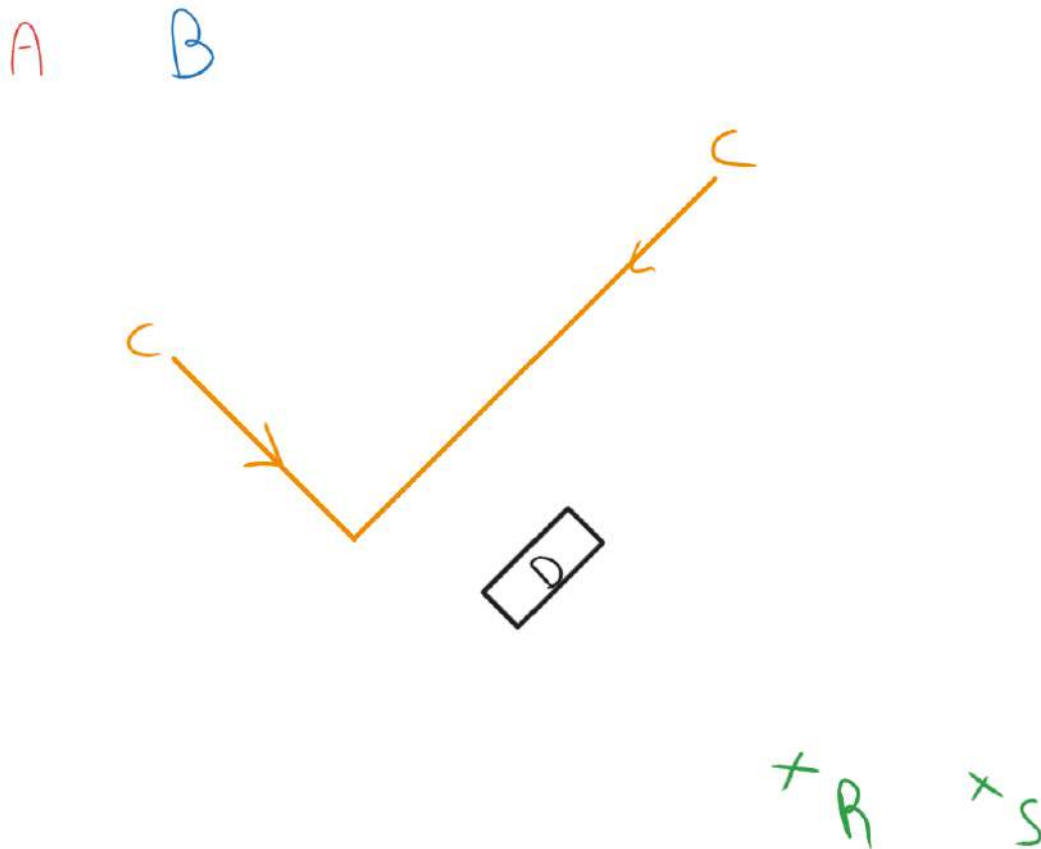
S : Bit de poids faible
R : Bit de poids fort

<i>A</i>	<i>B</i>	<i>S</i>	<i>R</i>
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

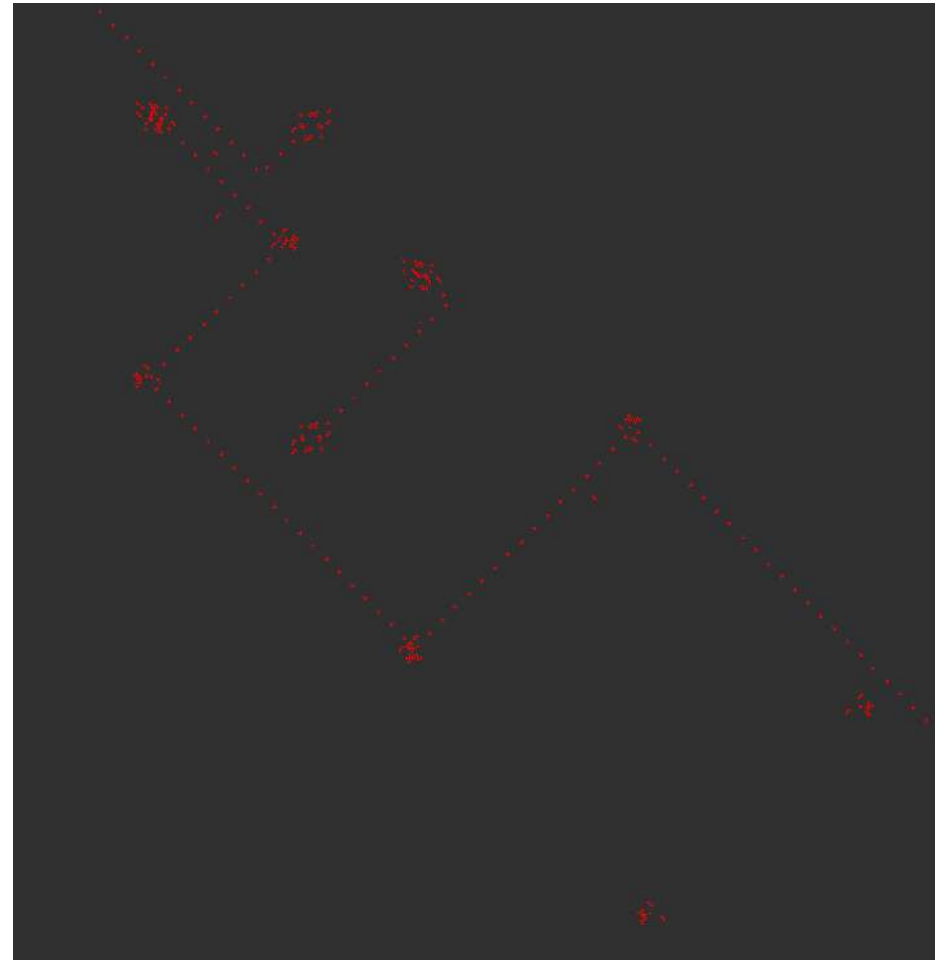
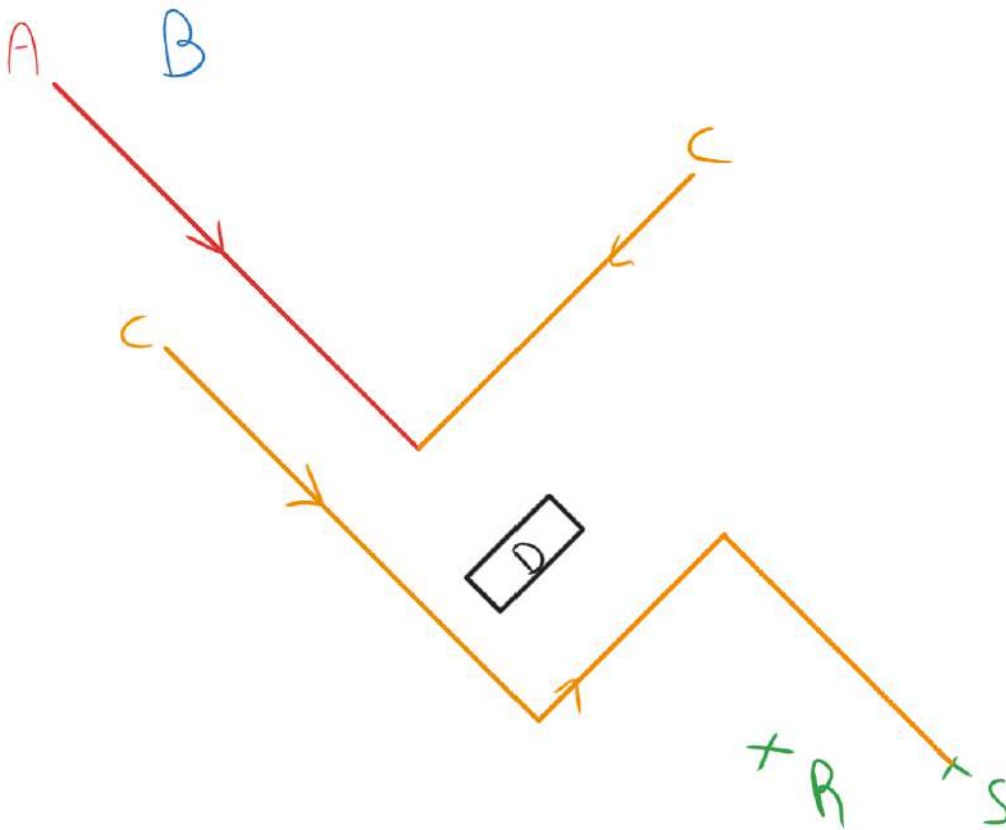
Semi-Additionneur : $A = 0 \mid B = 0 \mid S = 0 \mid R = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

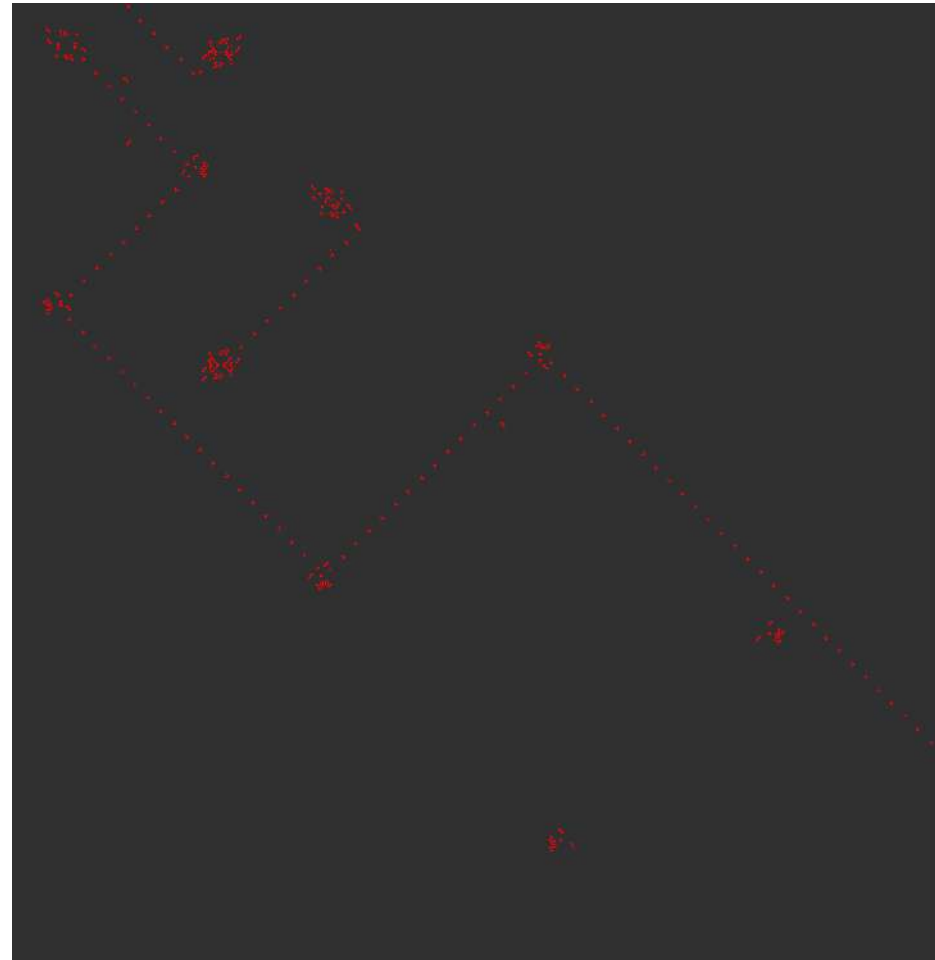
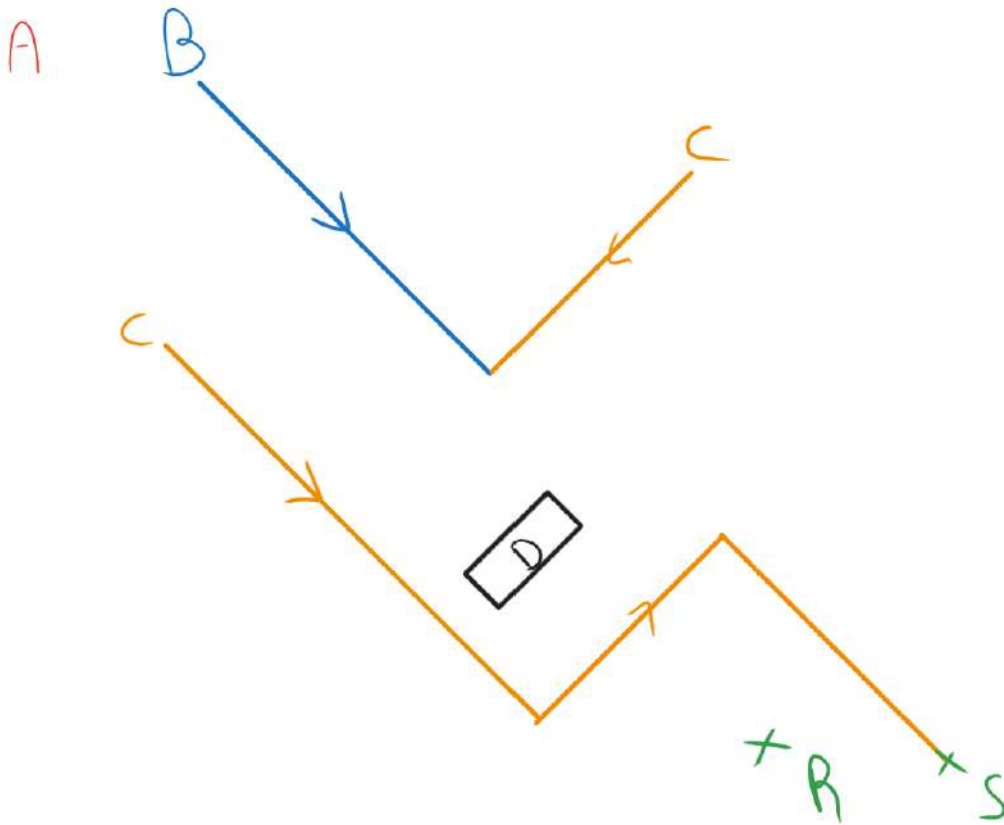
Semi-Additionneur : $A = 1 \mid B = 0 \mid S = 1 \mid R = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

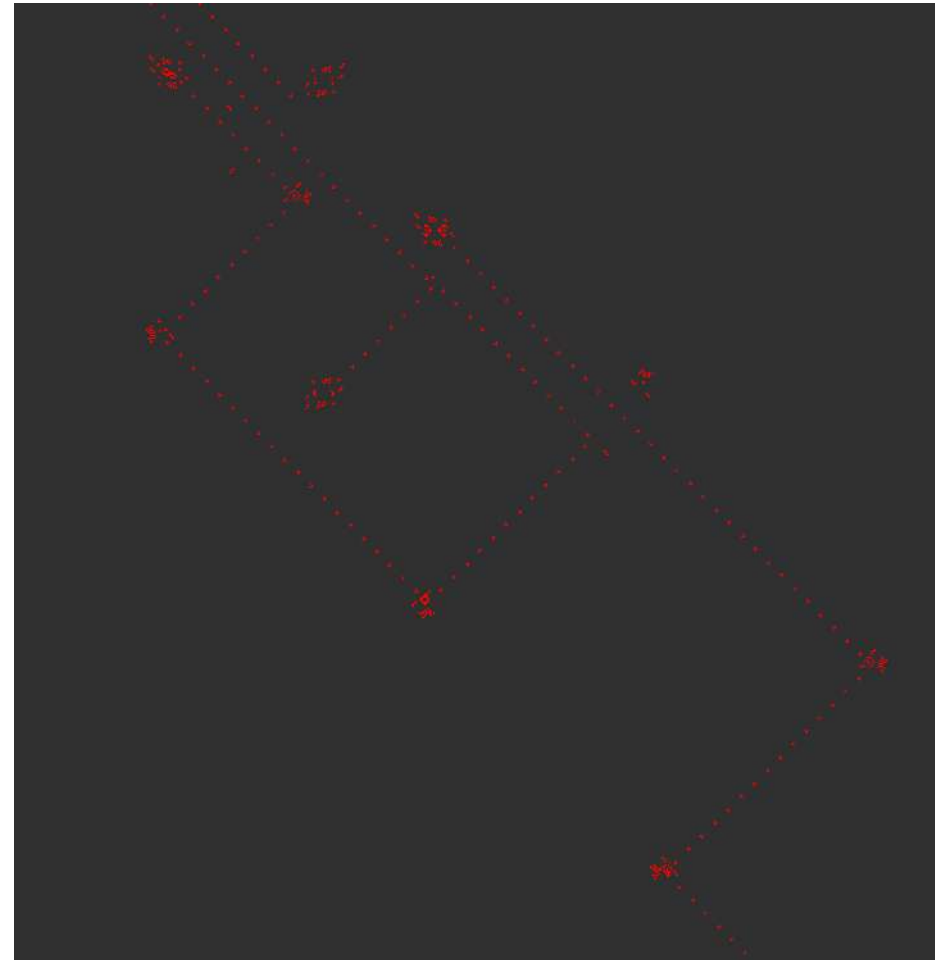
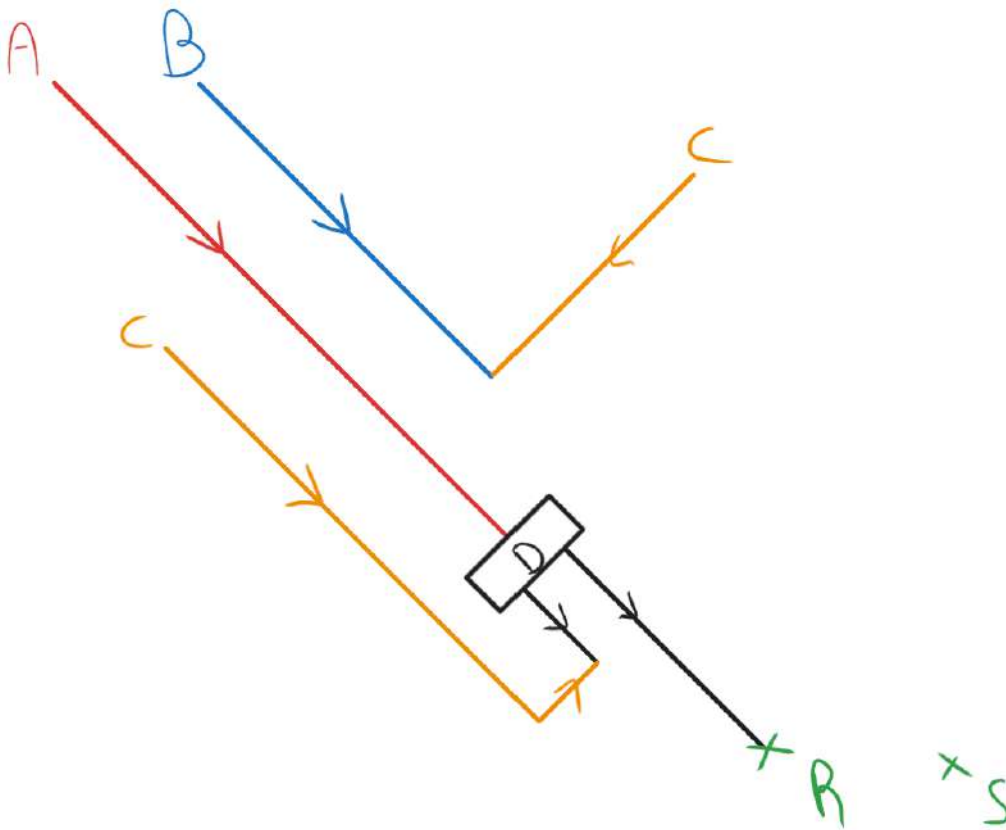
Semi-Additionneur : $A = 0 \mid B = 1 \mid S = 1 \mid R = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

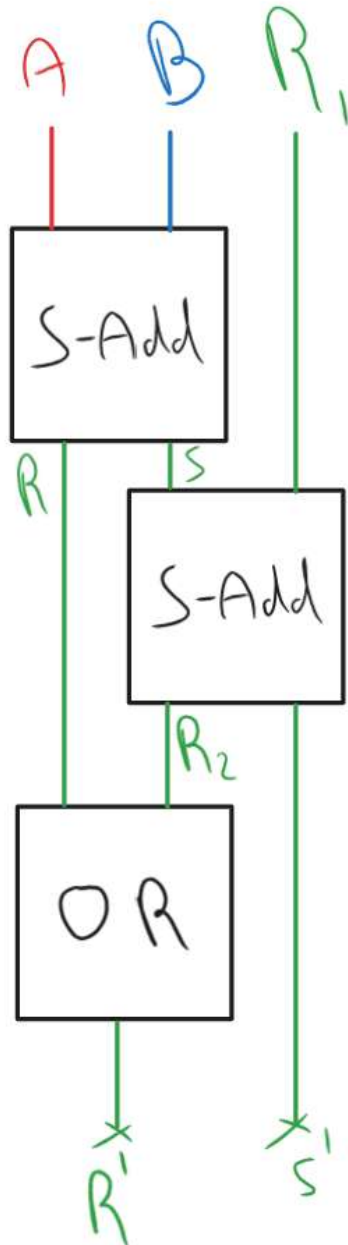
Semi-Additionneur : $A = 1 \mid B = 1 \mid S = 0 \mid R = 1$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur :

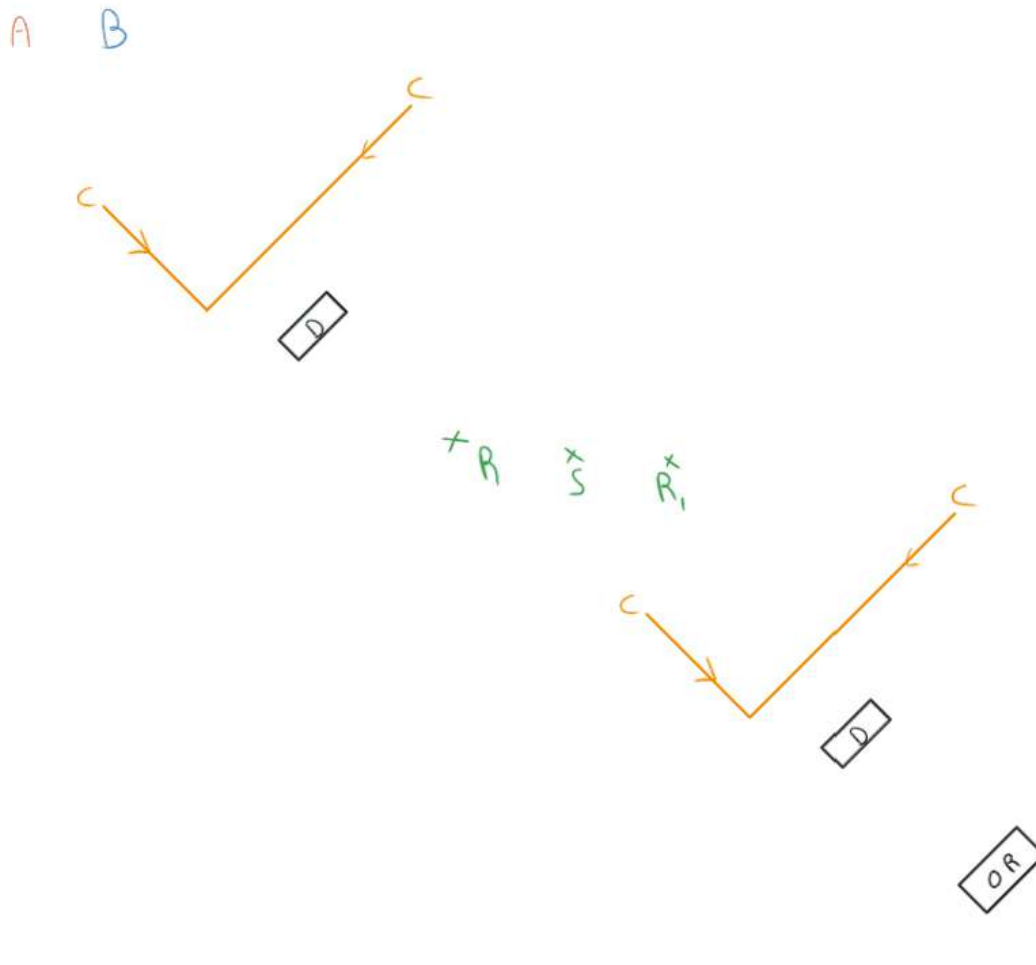


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 0 \mid B = 0 \mid R_1 = 0$

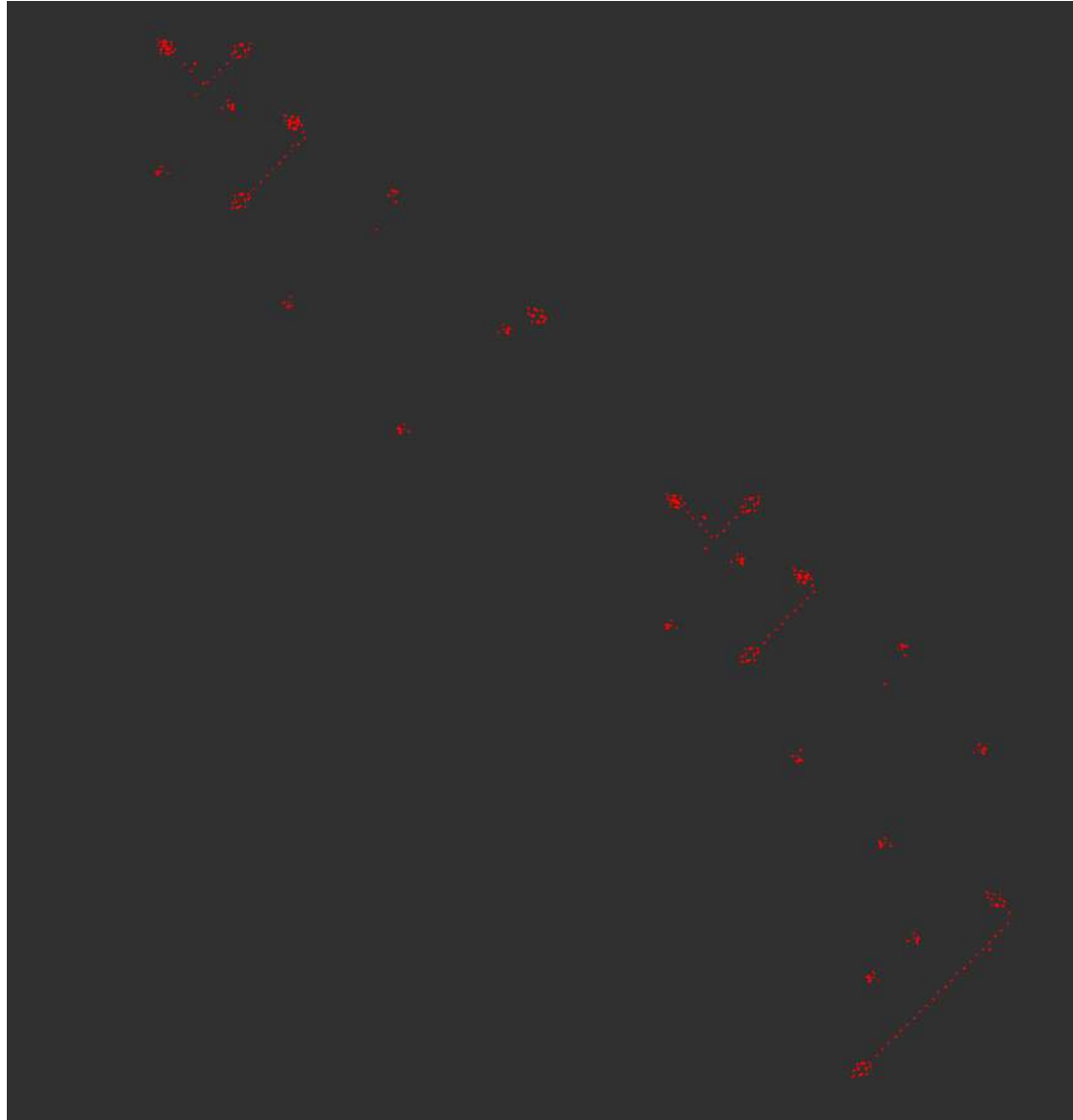


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

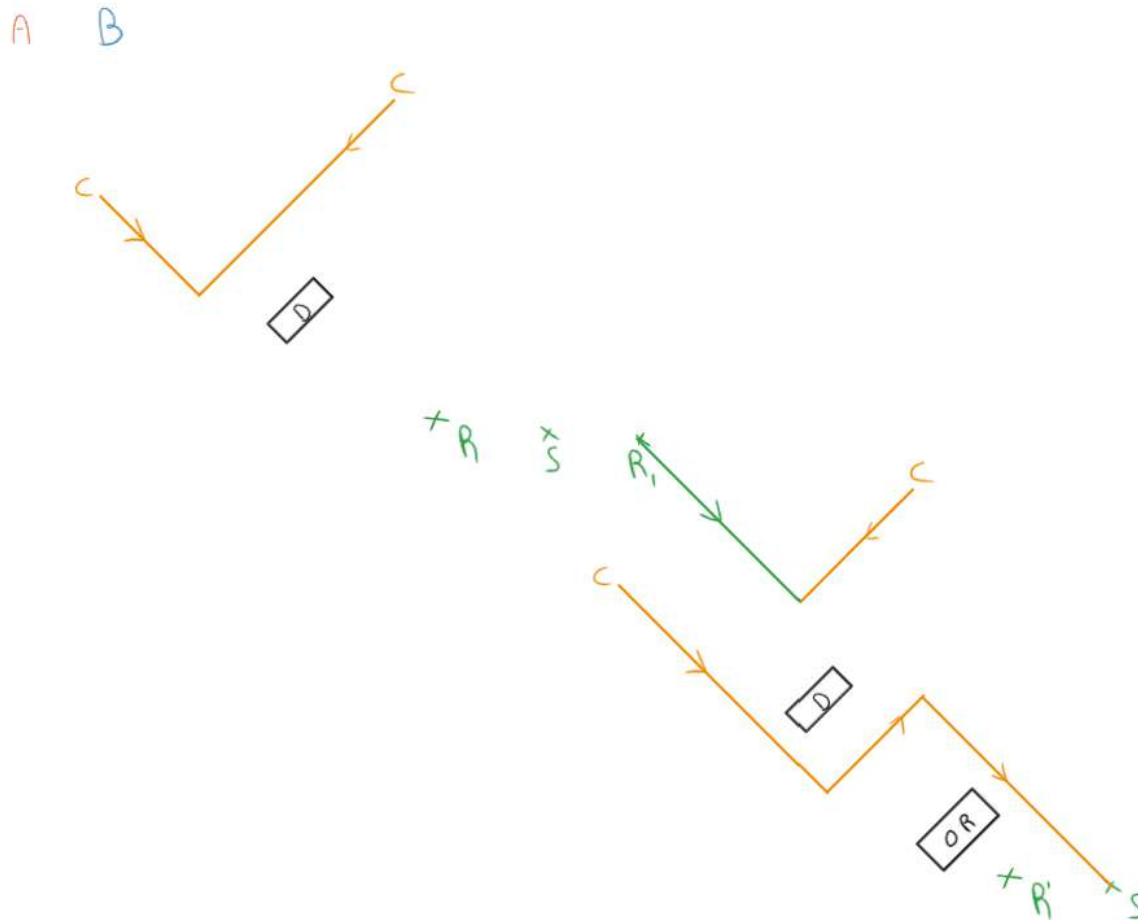
Additionneur : $A = 0 \mid B = 0 \mid R_1 = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 0 \mid B = 0 \mid R_1 = 1$

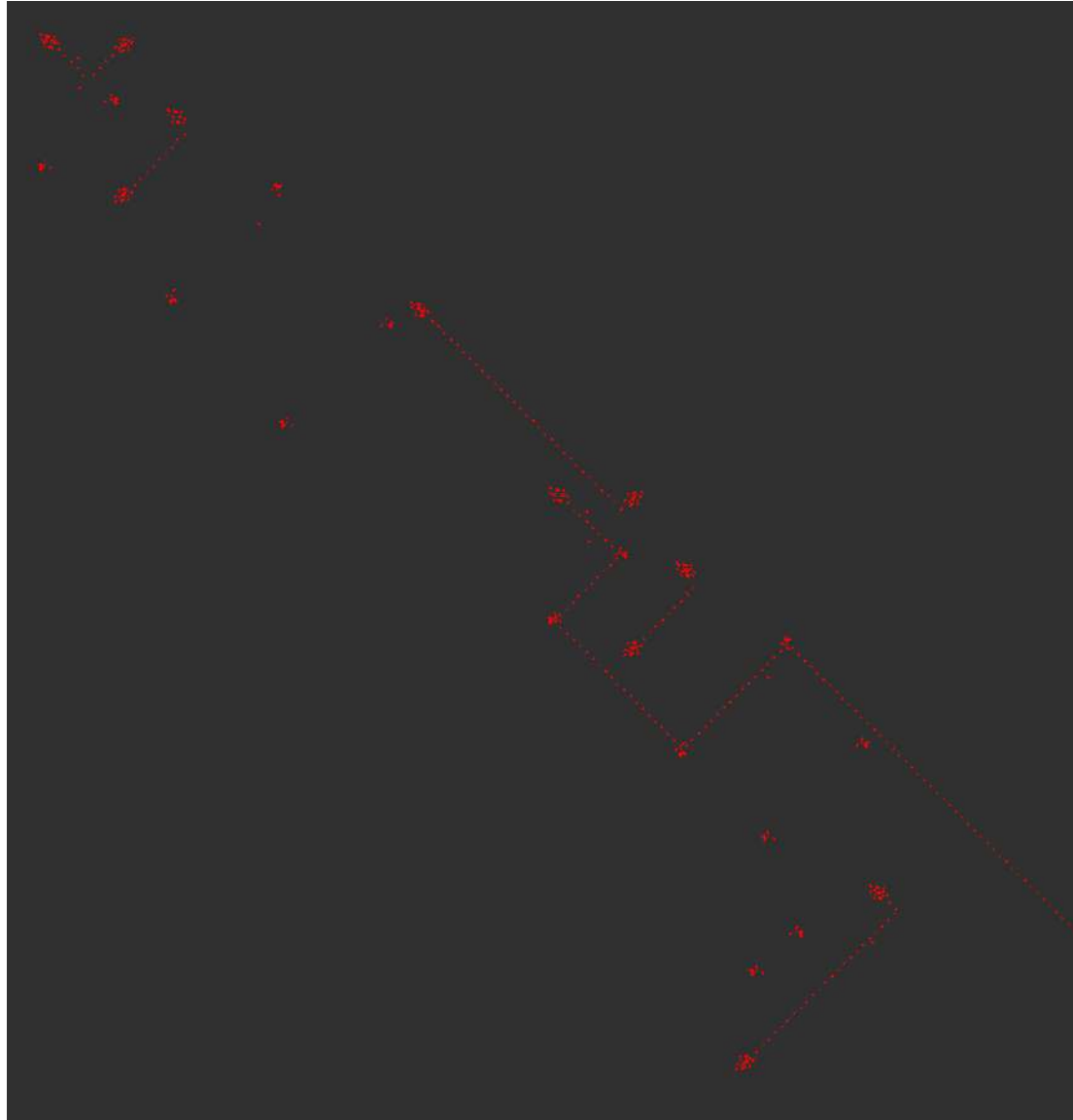


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

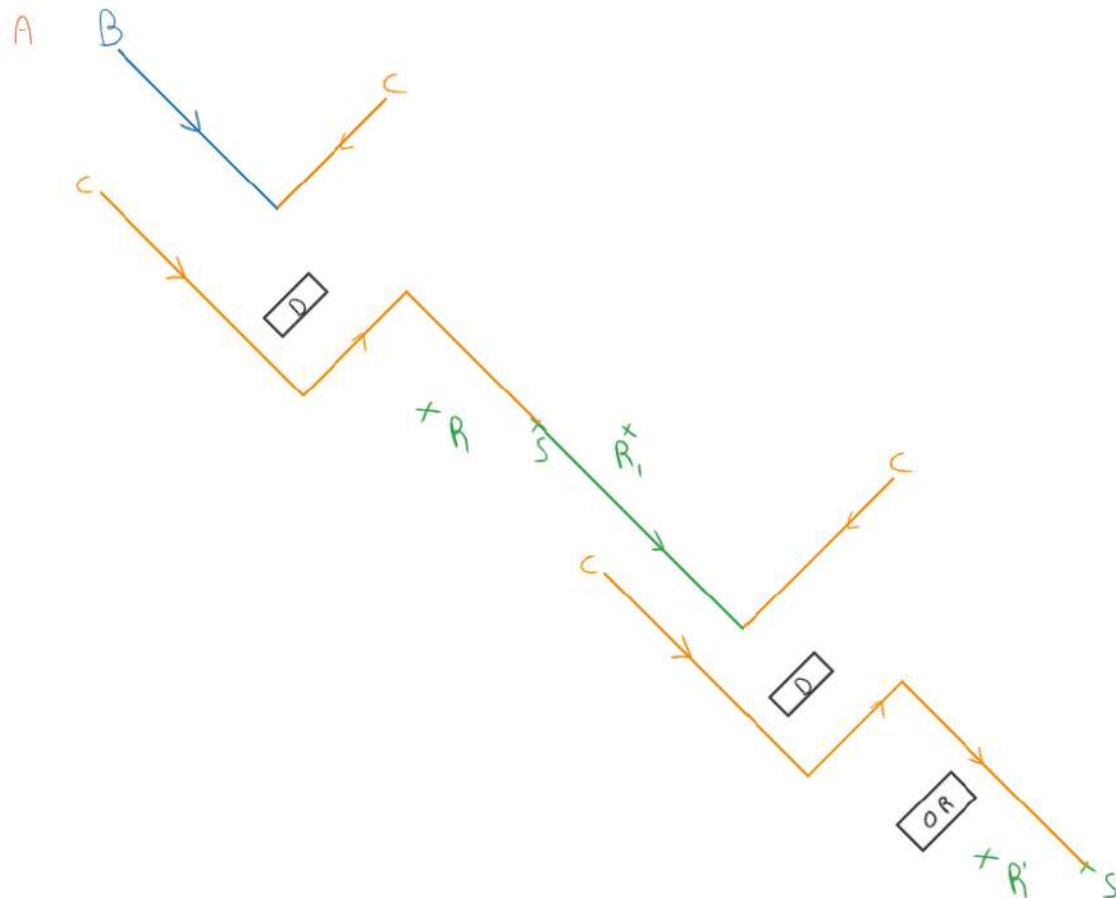
Additionneur : $A = 0 \mid B = 0 \mid R_1 = 1$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 0 \mid B = 1 \mid R_1 = 0$

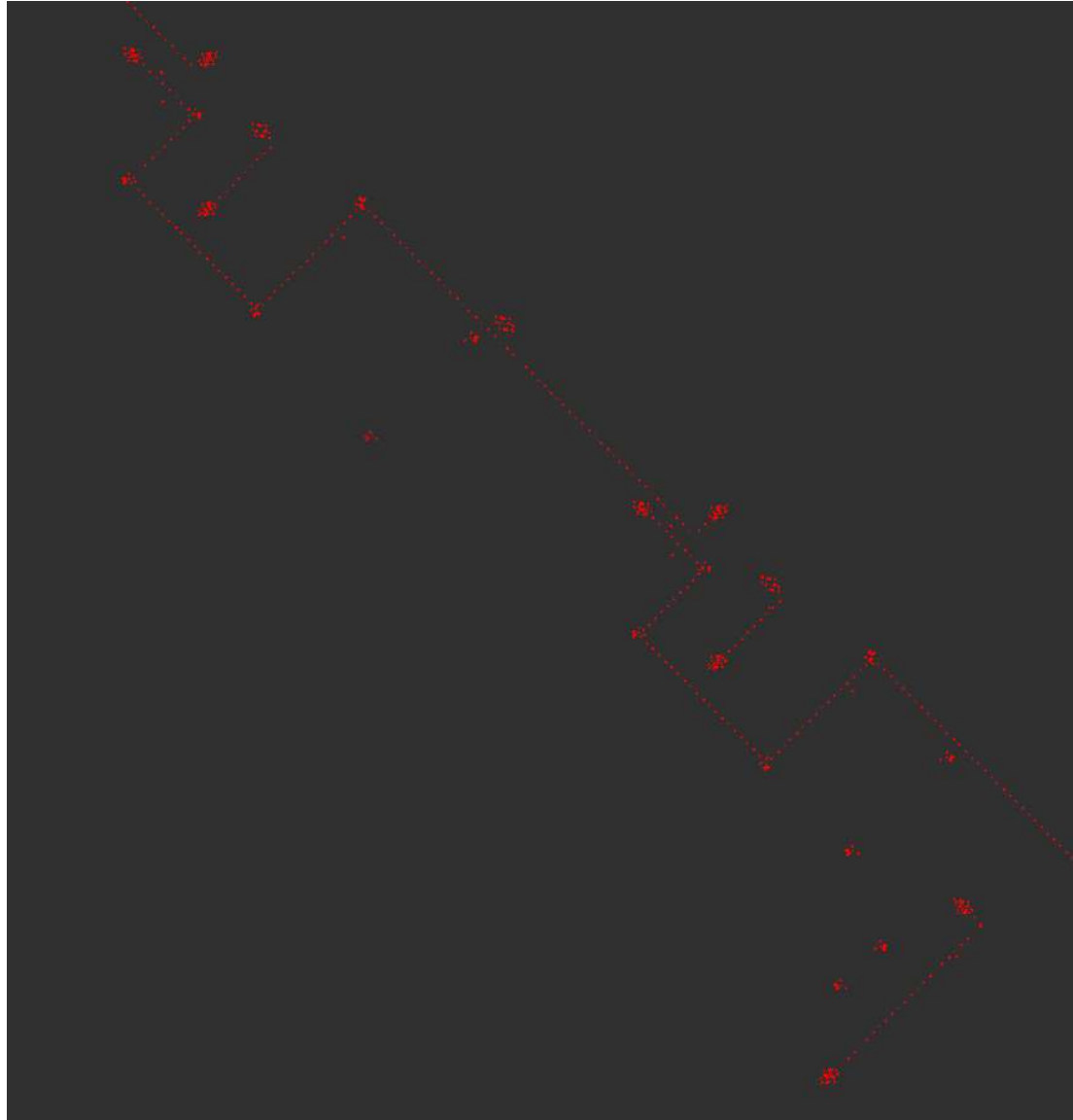


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

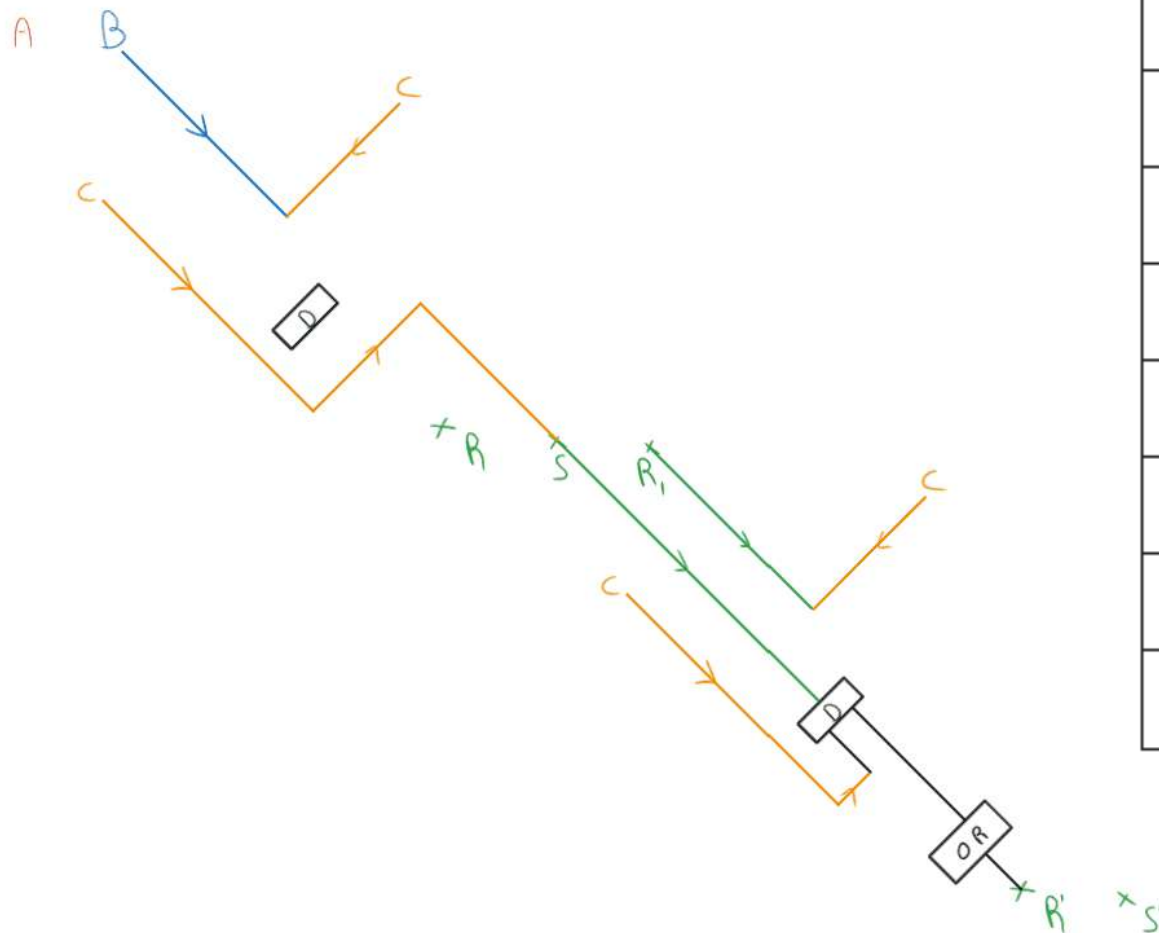
Additionneur : $A=0 \mid B=1 \mid R_1=0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 0 \mid B = 1 \mid R_1 = 1$

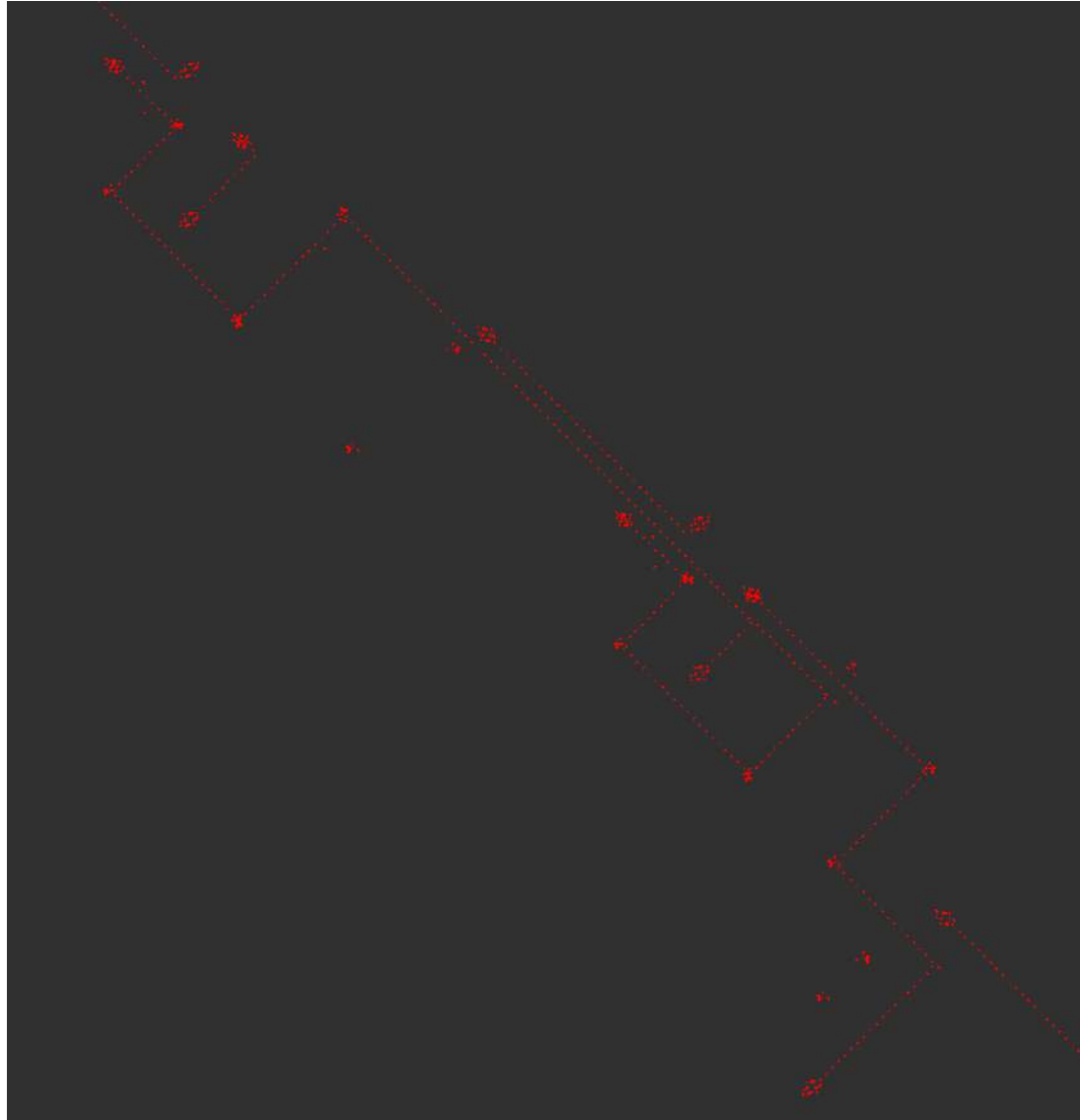


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

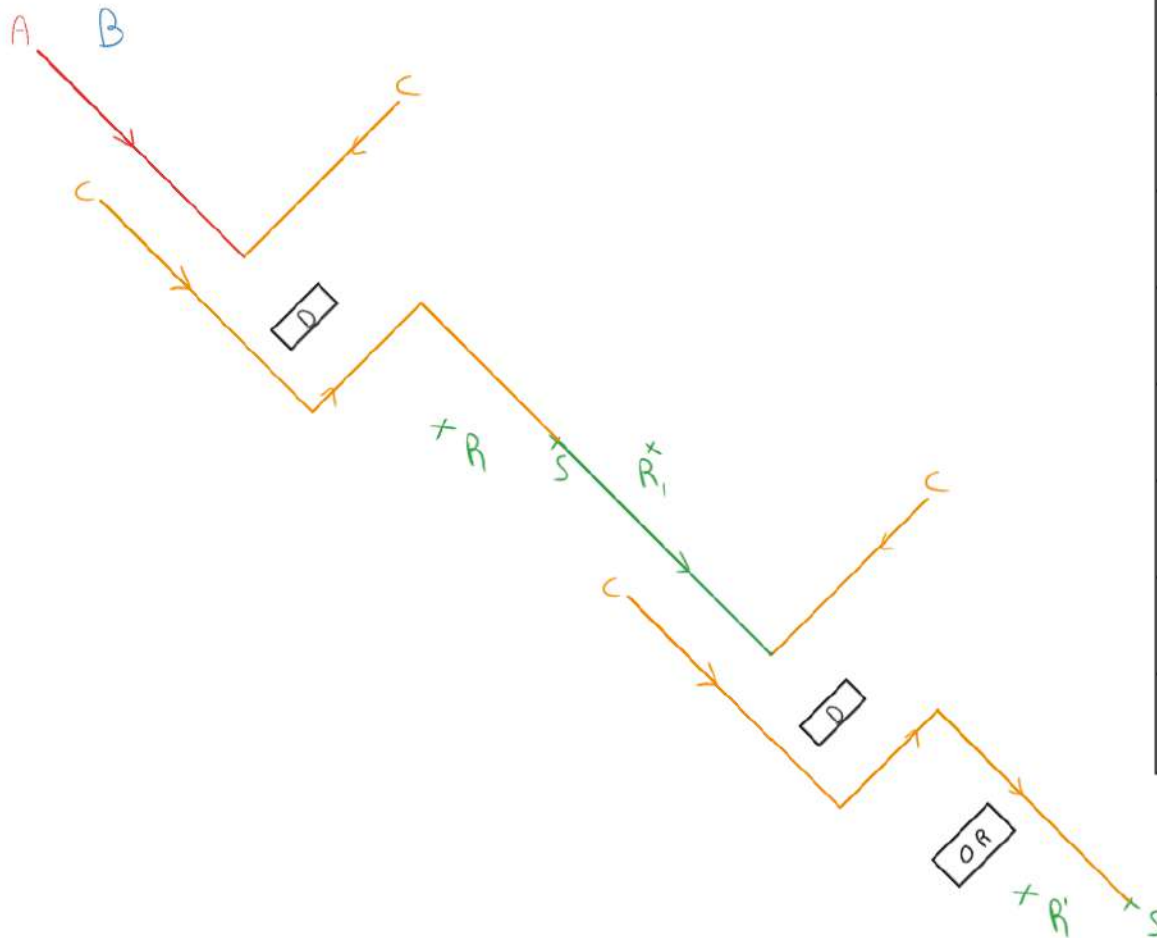
Additionneur : $A=0 \mid B=1 \mid R_1=1$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 1 \mid B = 0 \mid R_1 = 0$

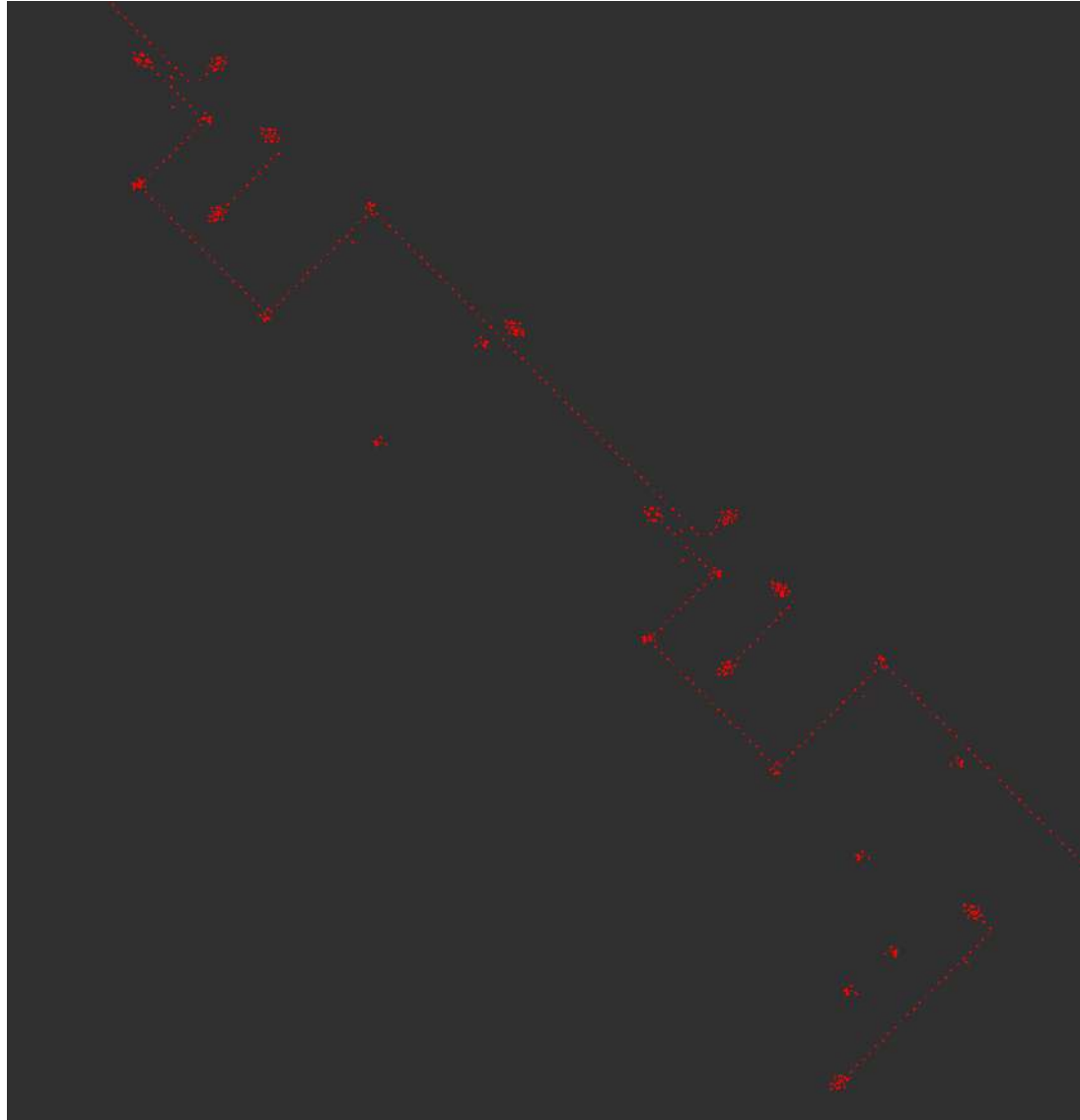


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

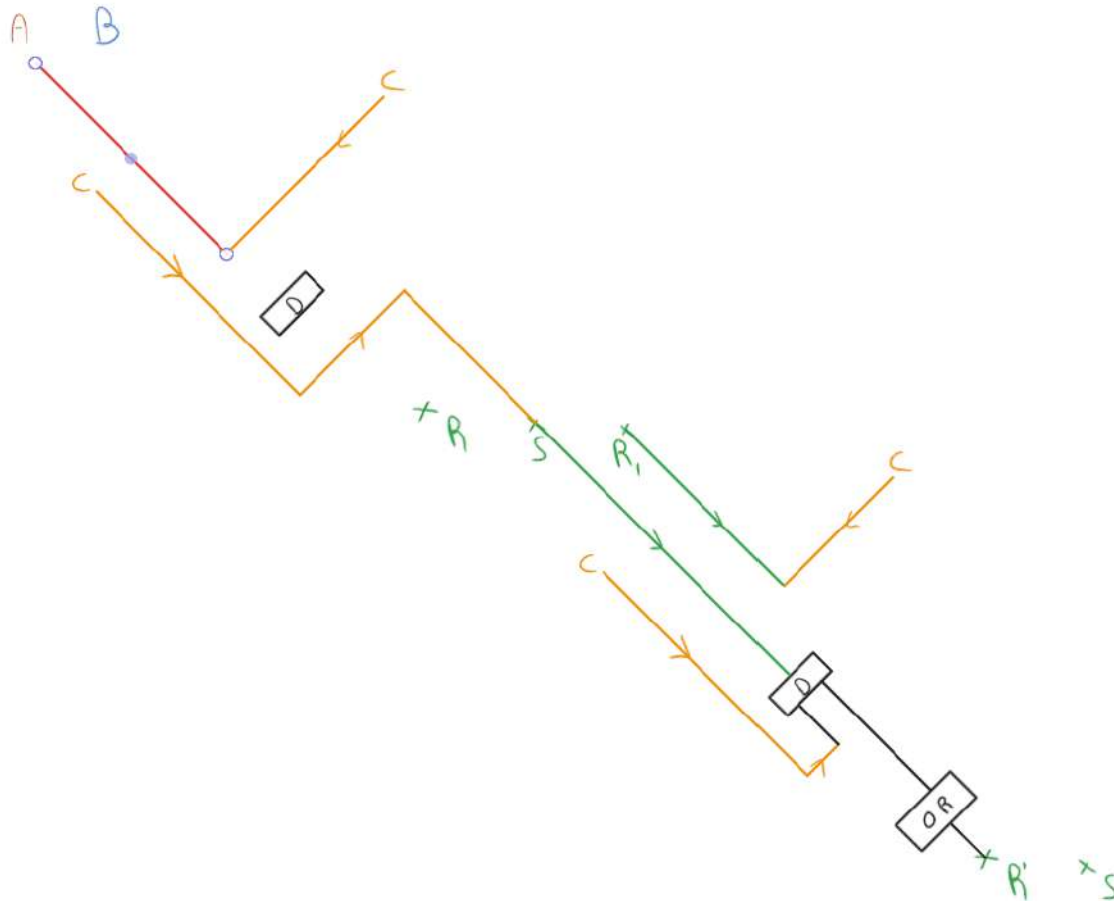
Additionneur : $A = 1 \mid B = 0 \mid R_1 = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 1 \mid B = 0 \mid R_1 = 1$

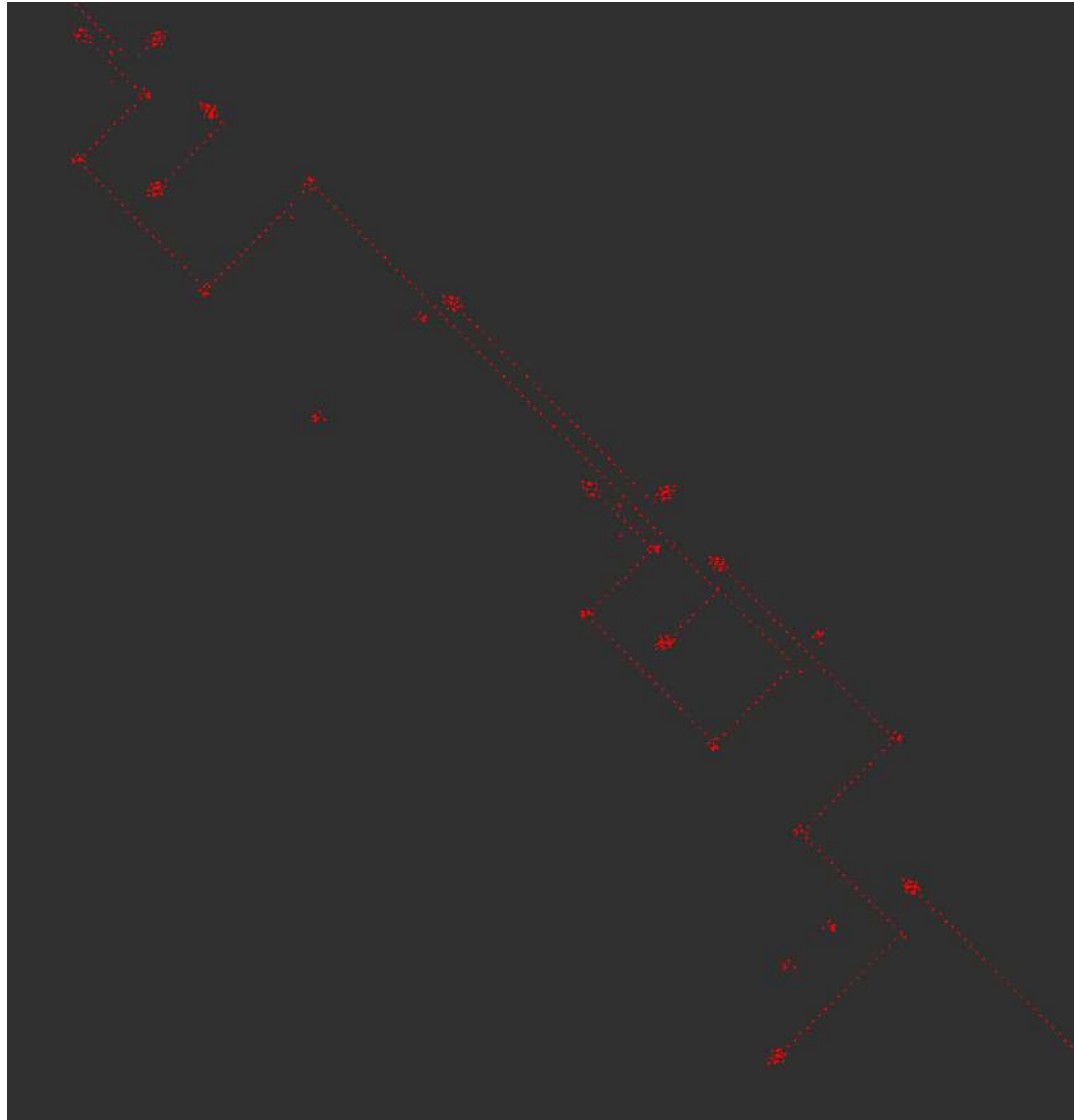


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

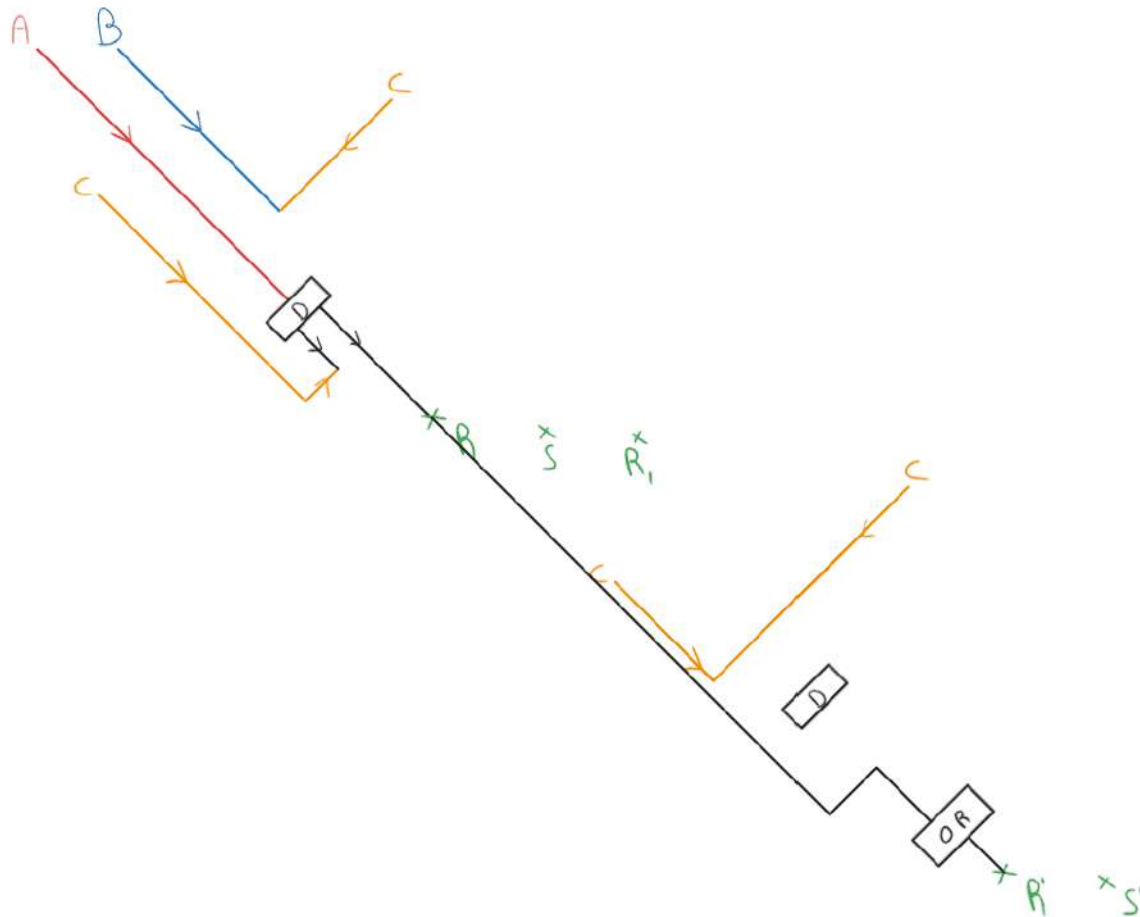
Additionneur : $A = 1 \mid B = 0 \mid R_1 = 1$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 1 \mid B = 1 \mid R_1 = 0$

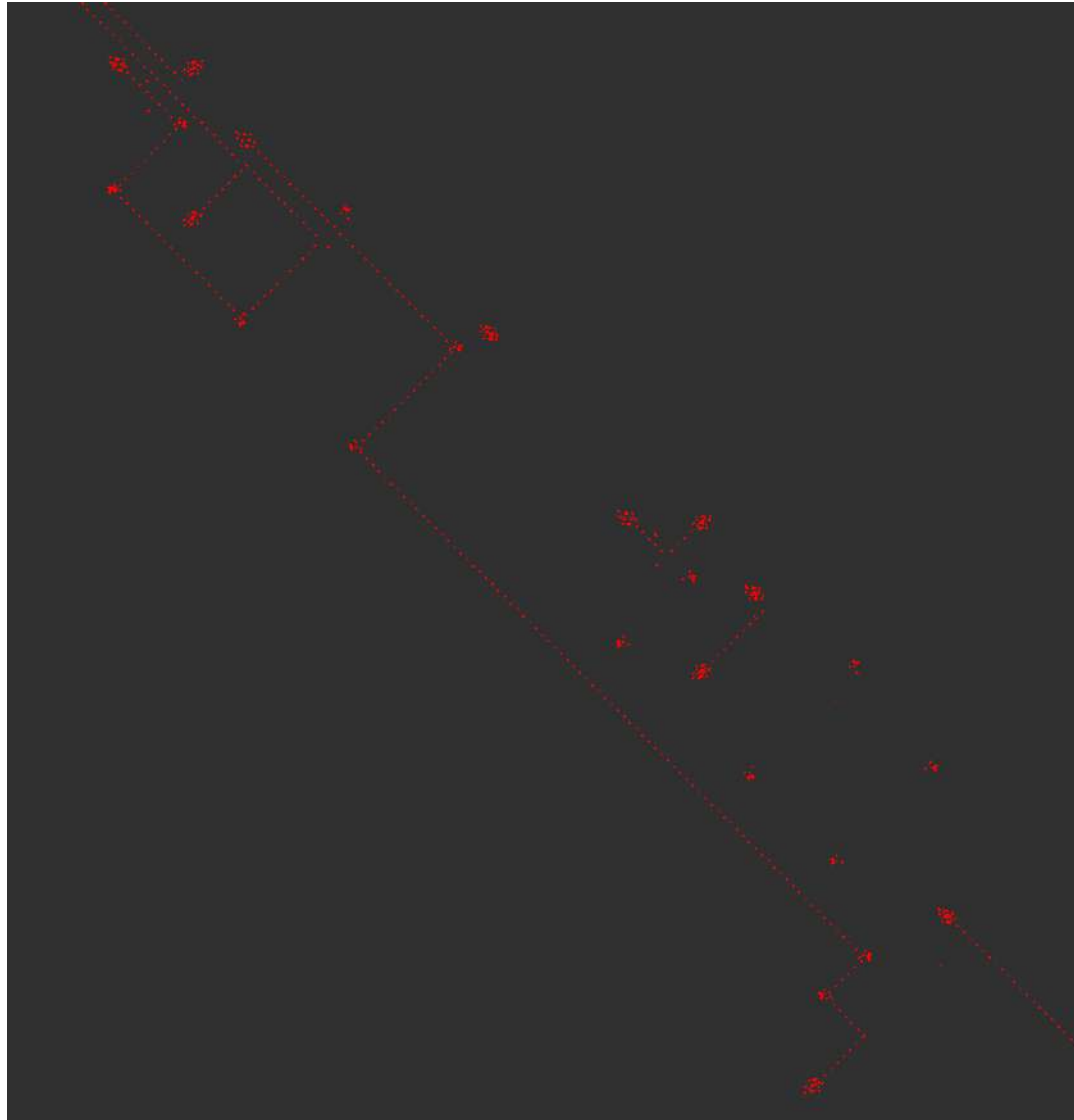


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

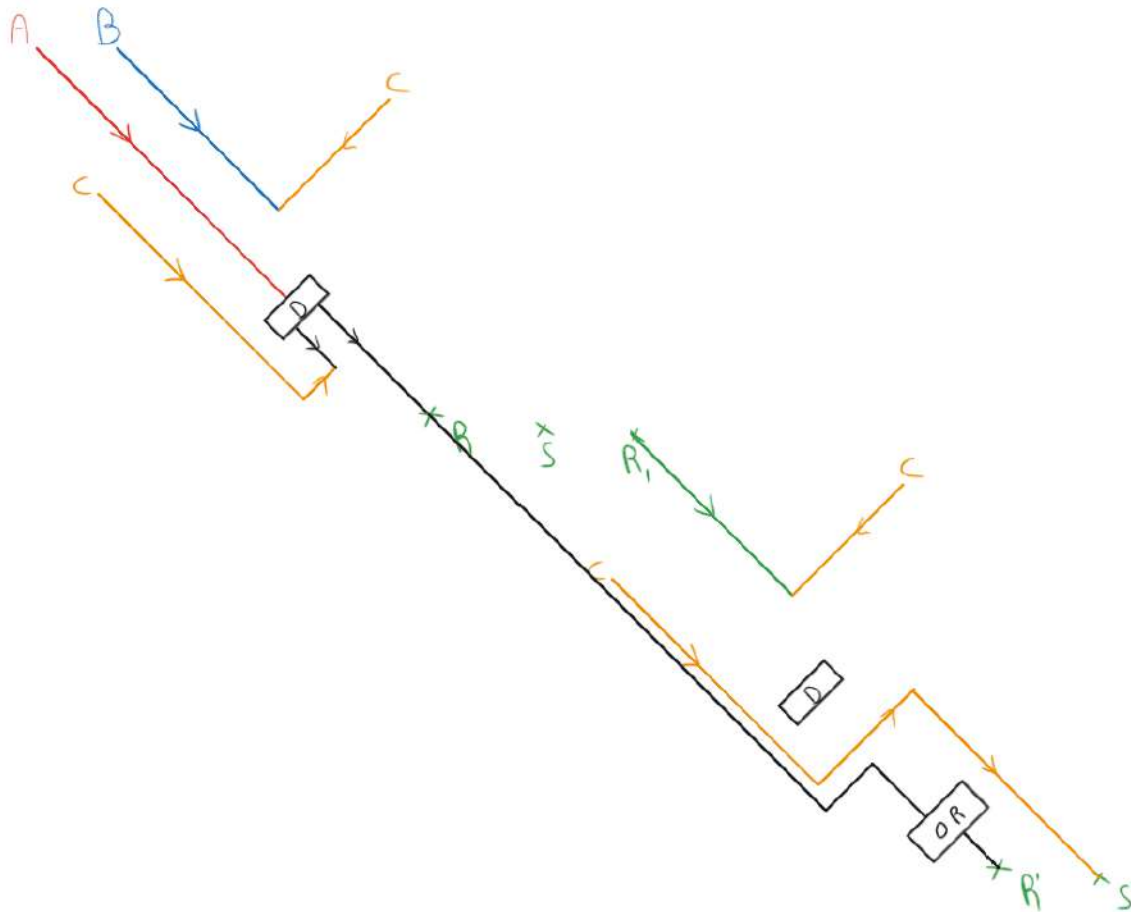
Additionneur : $A = 1 \mid B = 1 \mid R_1 = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur : $A = 1 \mid B = 1 \mid R_1 = 1$

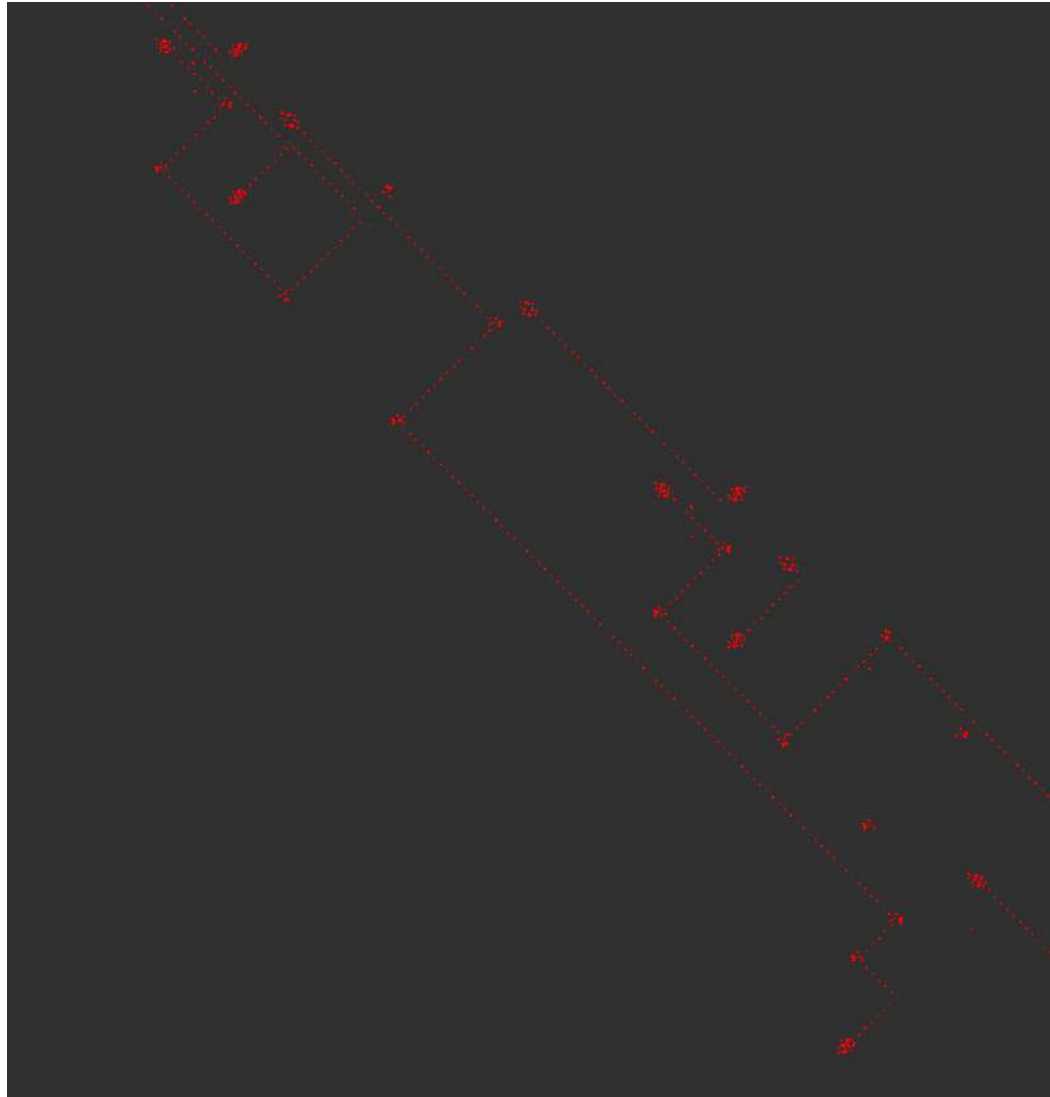


A	B	R_1	S'	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

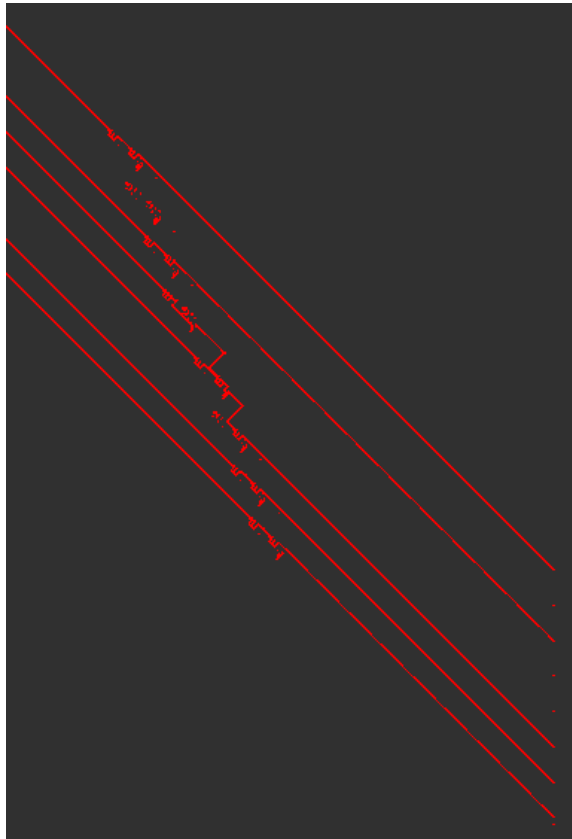
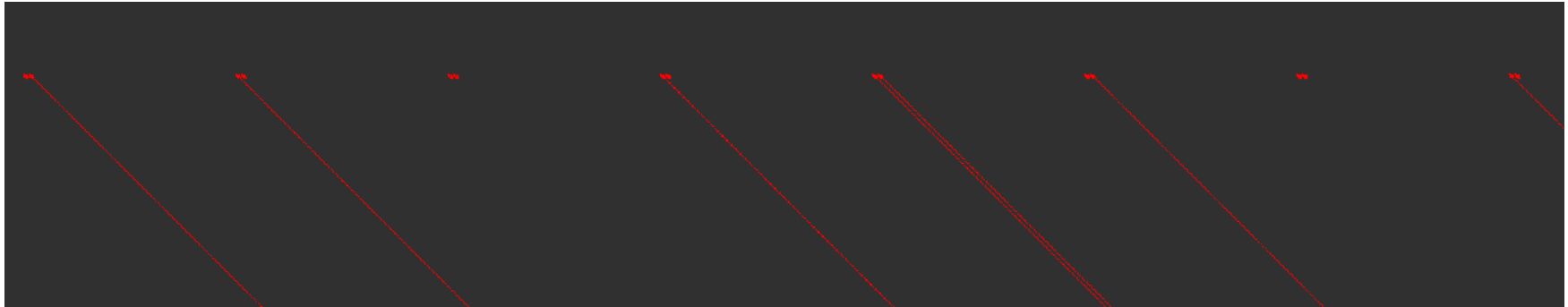
Additionneur : $A = 1 \mid B = 1 \mid R_1 = 0$



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Additionneur 8 bits :

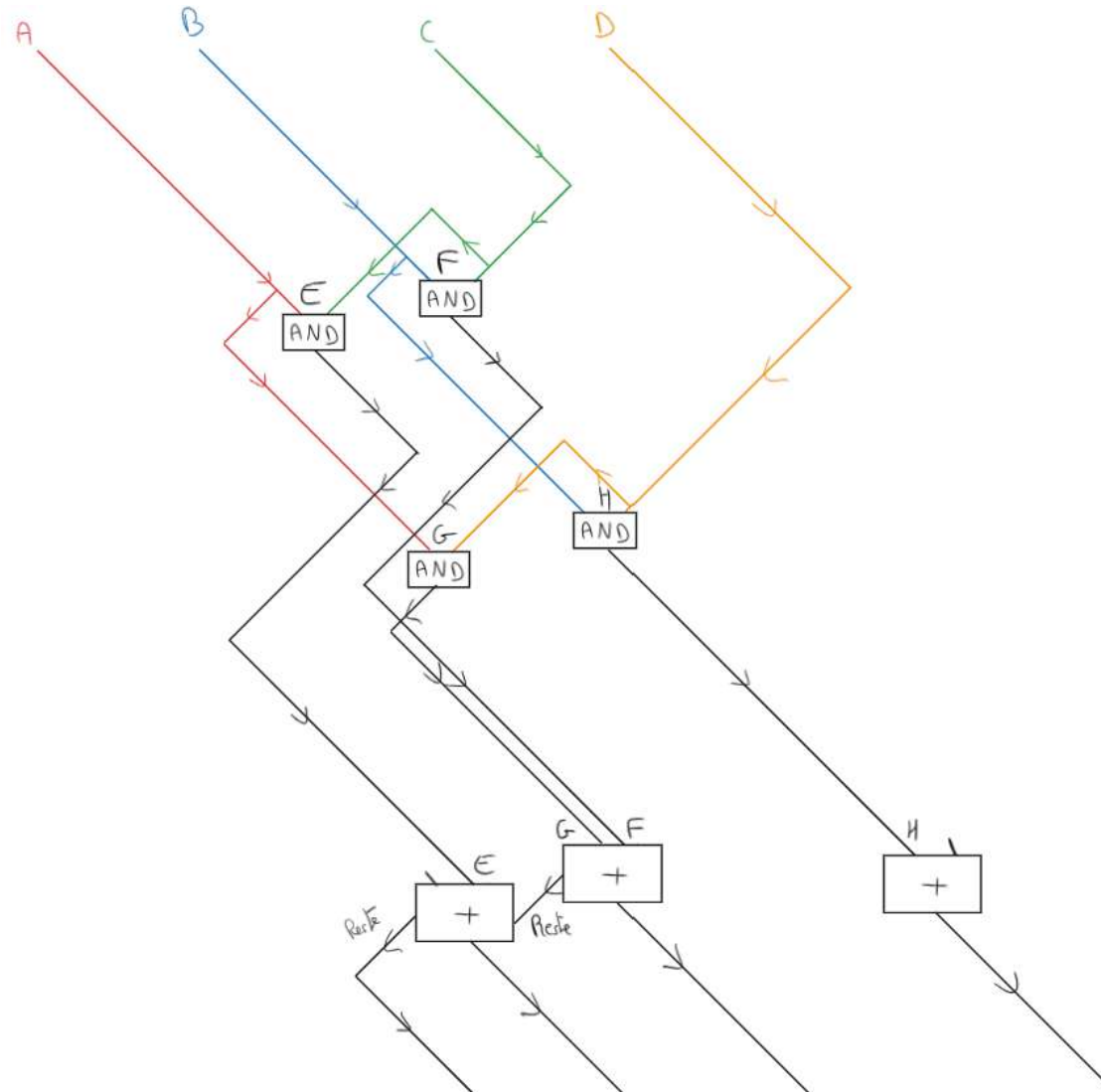


$$\begin{aligned} 10001100 + 01011001 \\ &= \\ 140 + 89 \\ &= \\ 229 \\ &= \\ 11100101 \end{aligned}$$

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

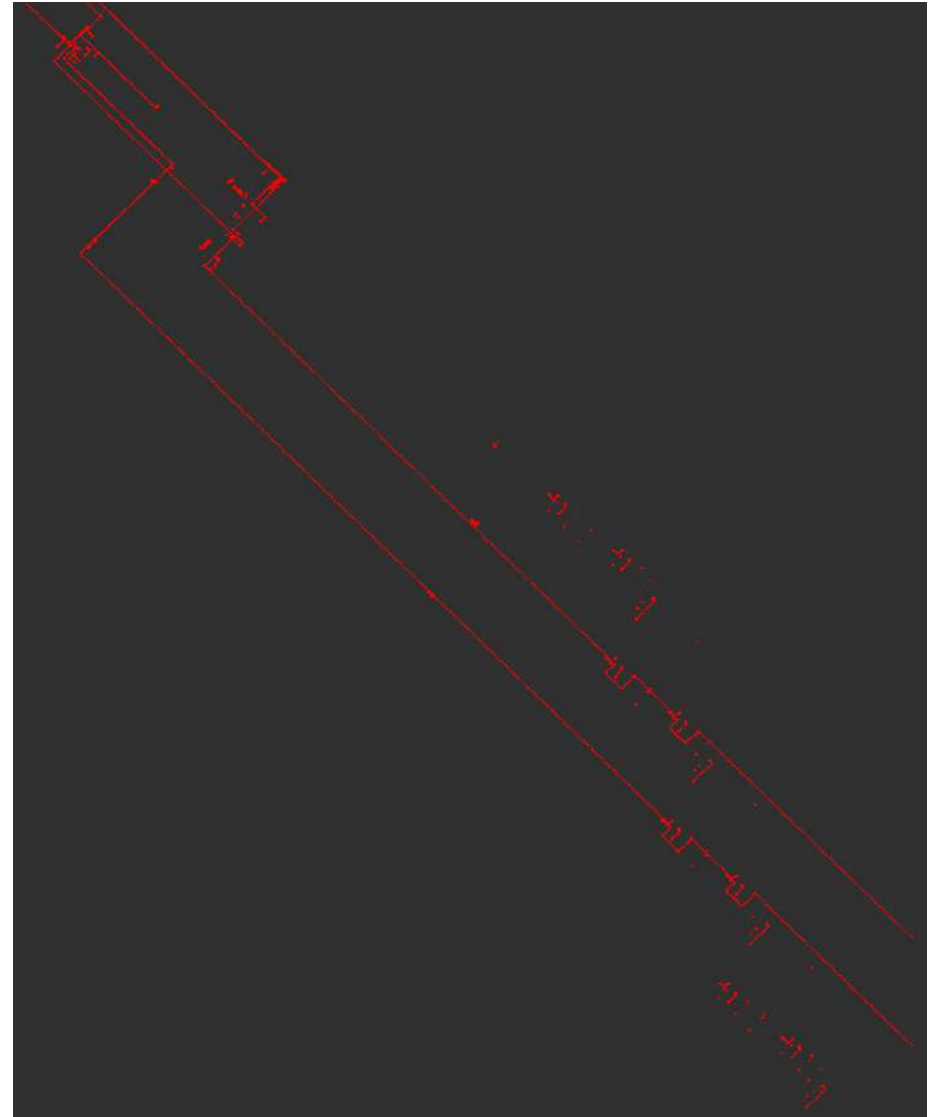
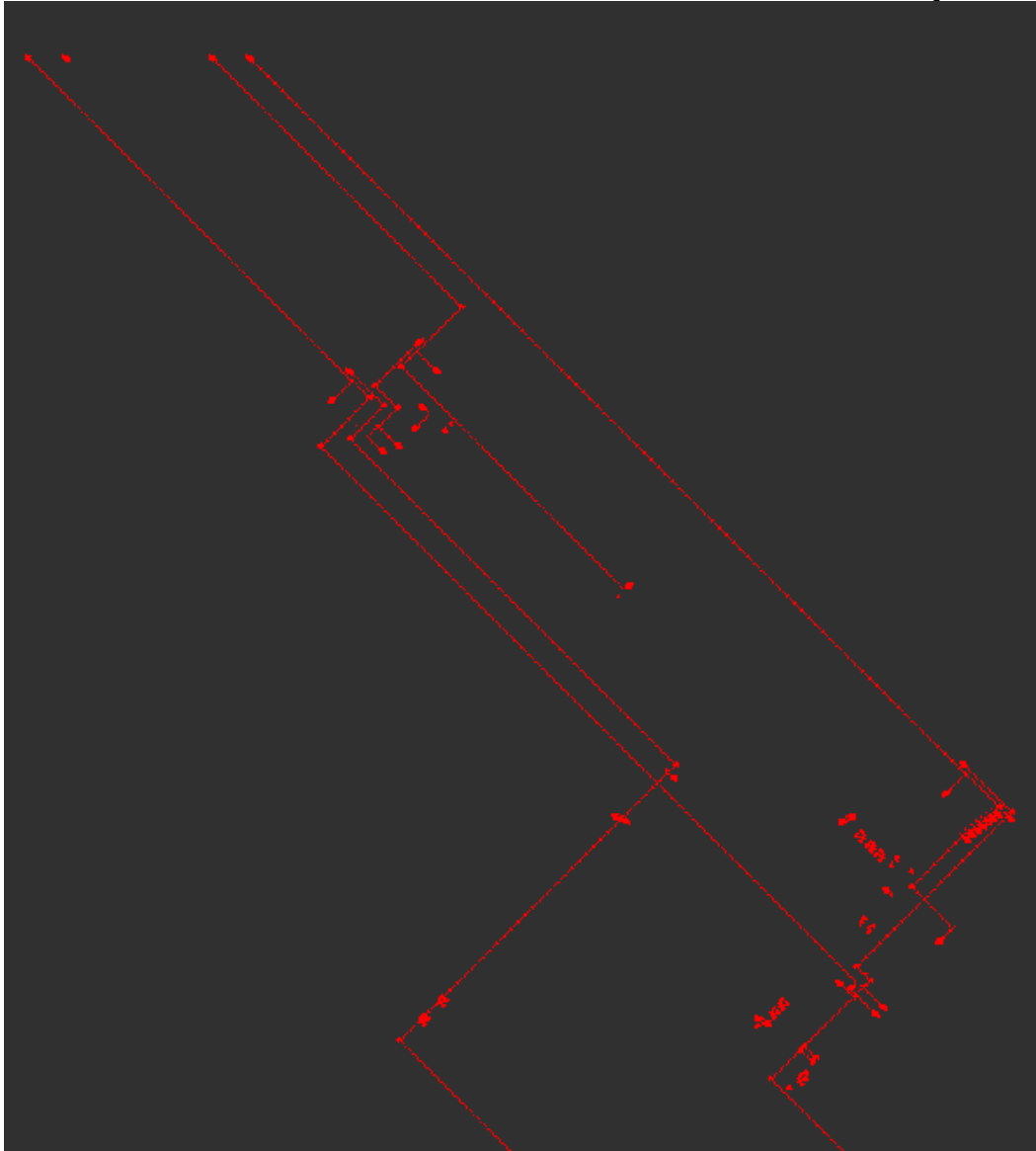
Multiplicateur :



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Multiplicateur :



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Multiplicateur 4x3 bits :

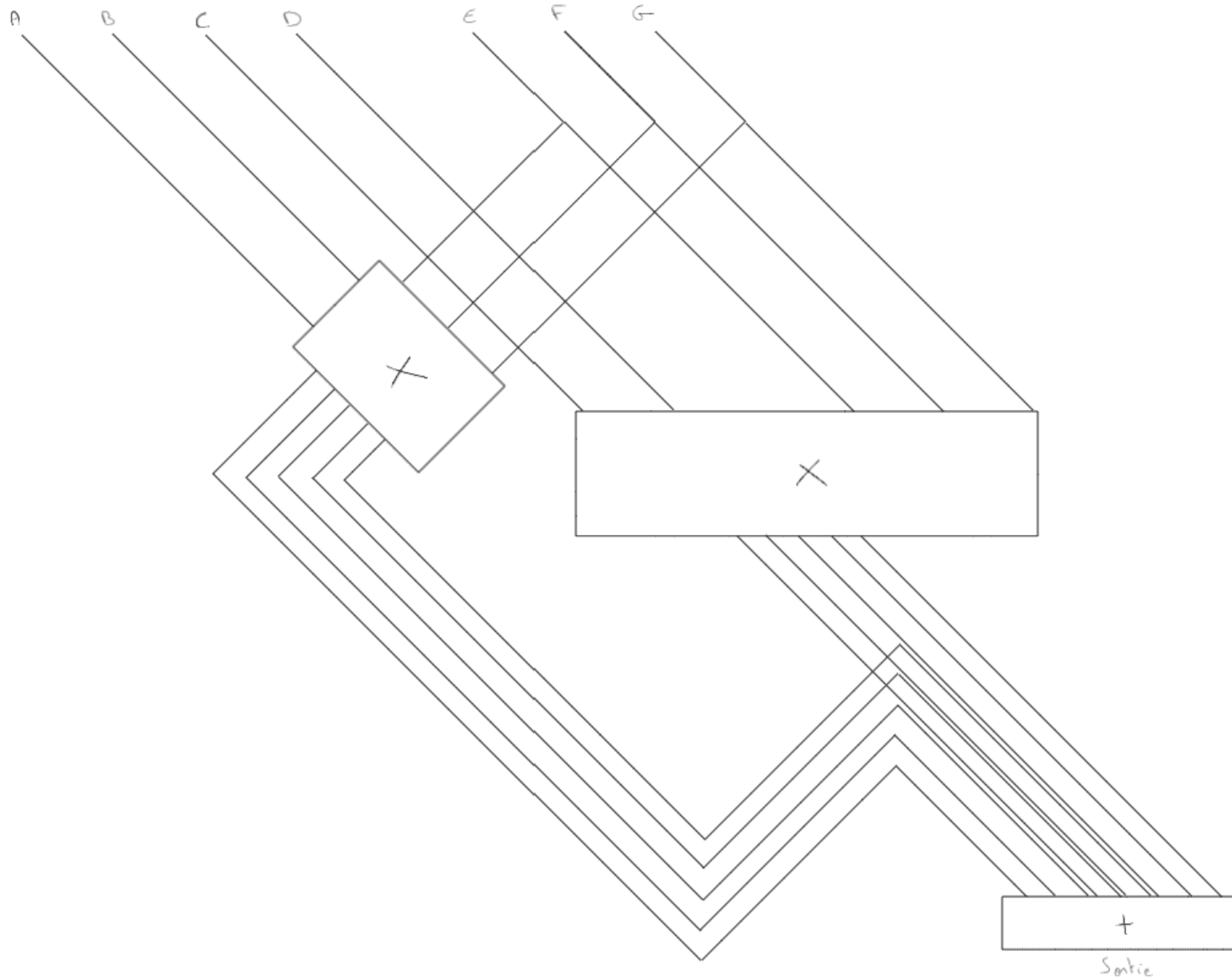
Soient $A, B, C, D, E, F, G \in \{0, 1\}$,

$$\begin{aligned} & (A2^3 + B2^2 + C2^1 + D2^0)(E2^2 + F2^1 + G2^0) \\ & \quad = \\ & \quad 2^2(A2^1 + B2^0)(E2^2 + F2^1 + G2^0) \\ & \quad \quad + \\ & \quad (F2^1 + G2^0)(E2^2 + F2^1 + G2^0) \end{aligned}$$

II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

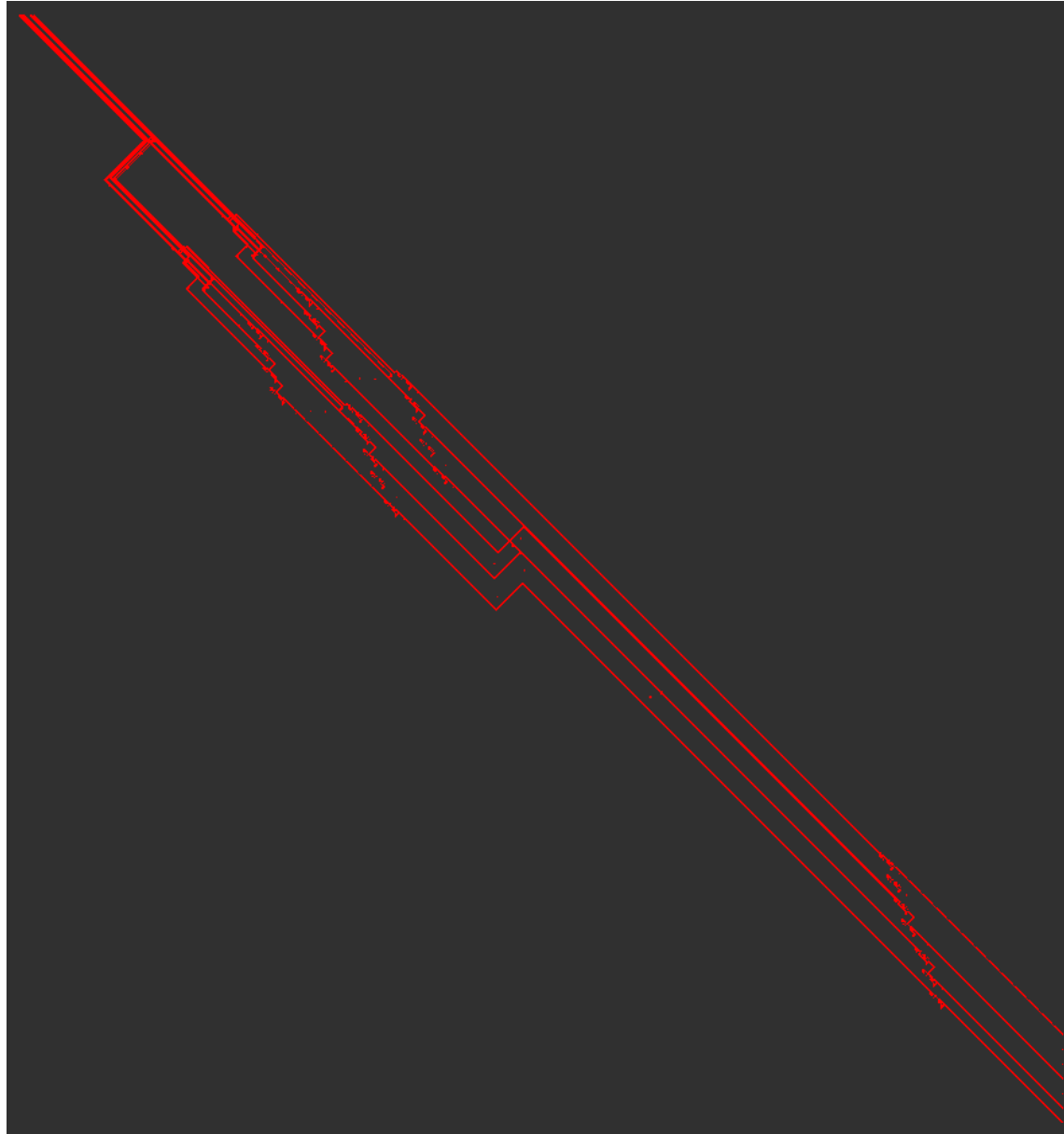
Multiplicateur 4x3 bits :



II. Outils nécessaires et composition du filtre

3. Additionneur et multiplicateur

Multiplicateur 4x3 bits :



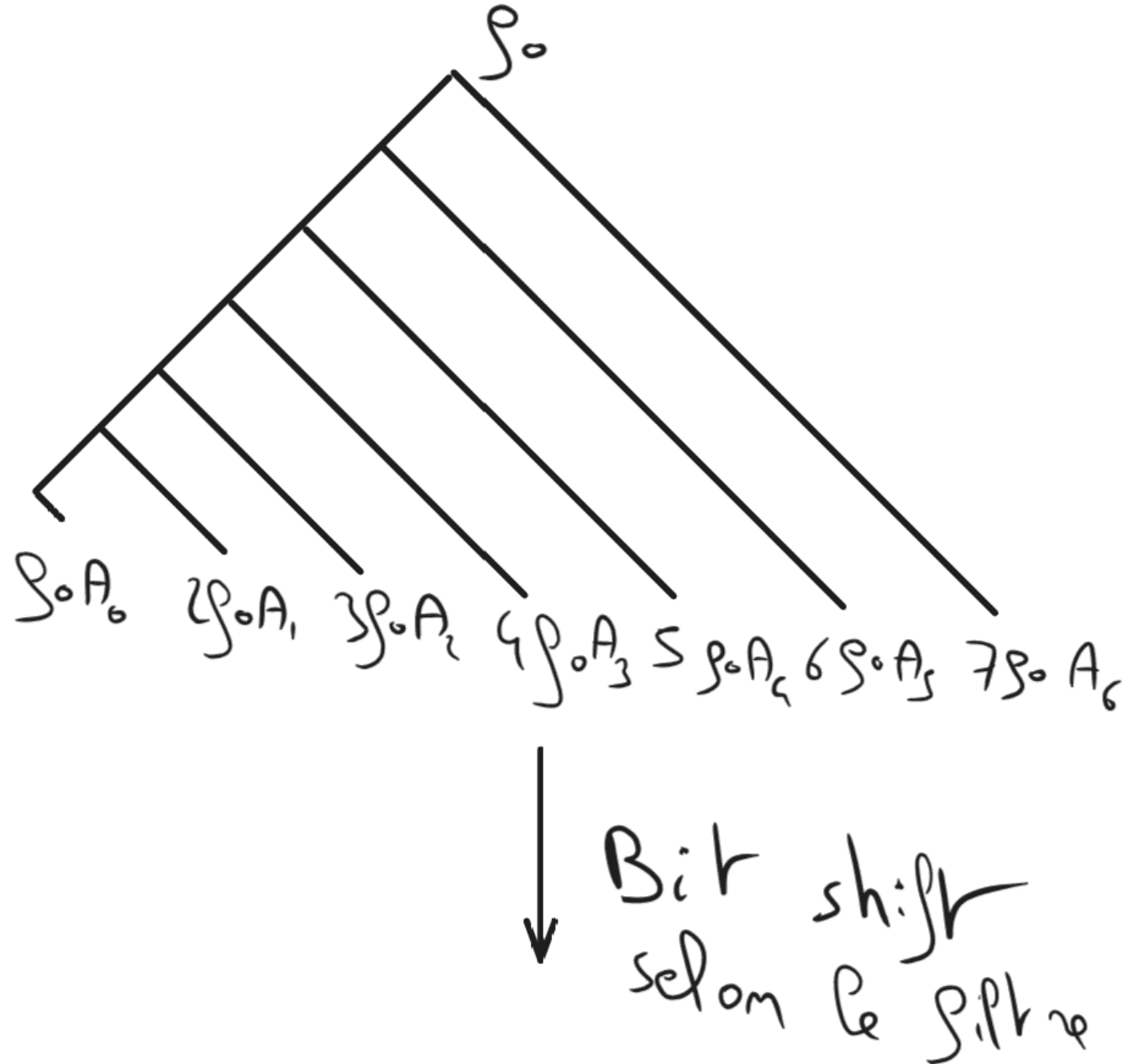
$$\begin{aligned} 1111 &\times 11 \\ &= \\ 15 &\times 7 \\ &= \\ 105 \\ &= \\ 1101001 \end{aligned}$$

II. Outils nécessaires et composition du filtre

4. Un filtre modulable

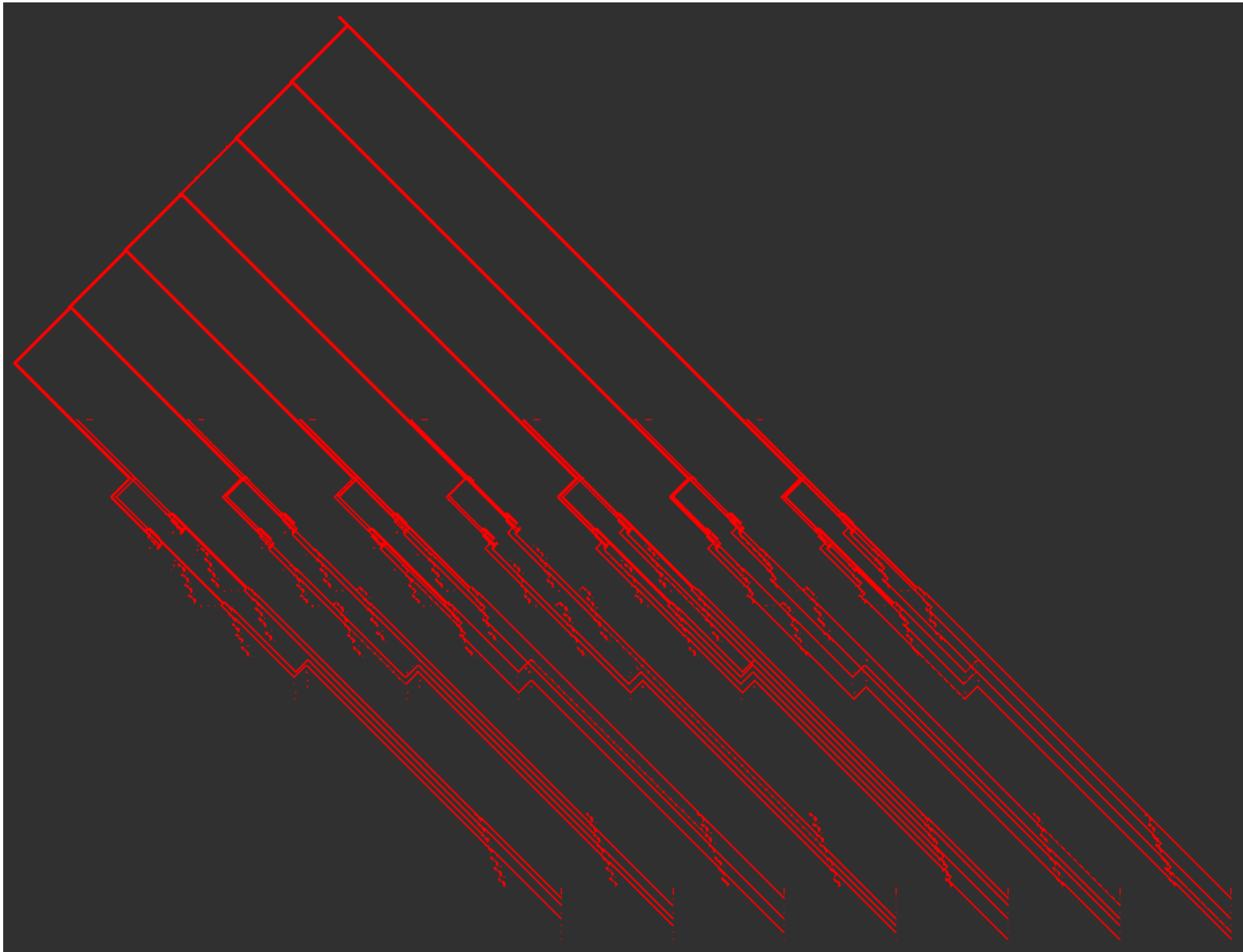
II. Outils nécessaires et composition du filtre

4. Un filtre modulable



II. Outils nécessaires et composition du filtre

4. Un filtre modulable



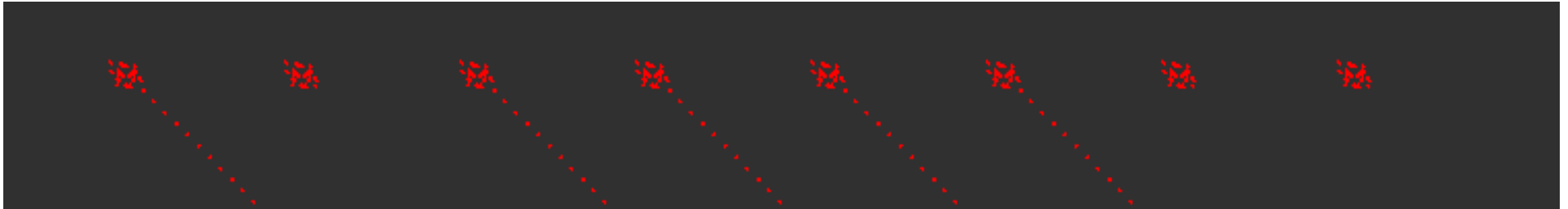
III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée) appliqués à ces différents filtres

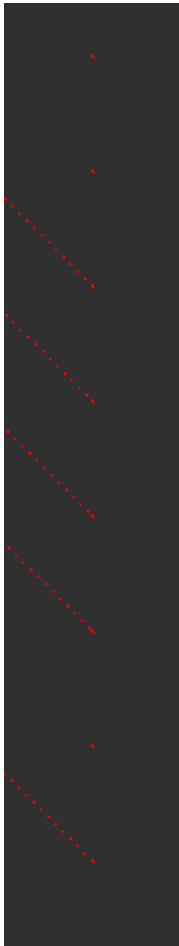
III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

188



Passe bas :



188

Passe Haut :



2

Passe Bande :

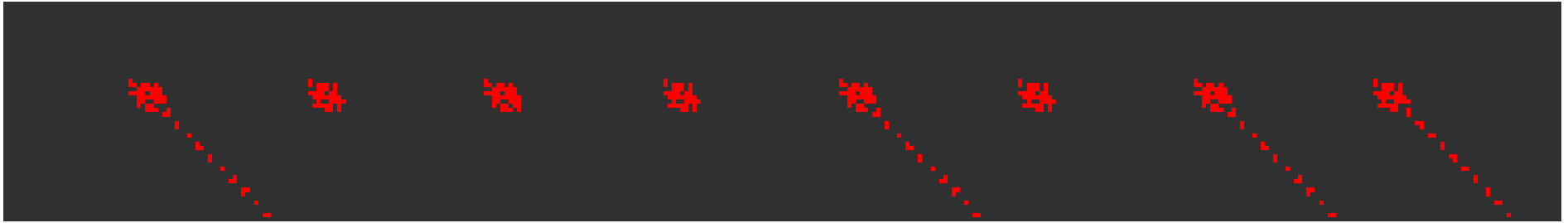


47

III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

139



Passe bas :

69

Passe Haut :

4

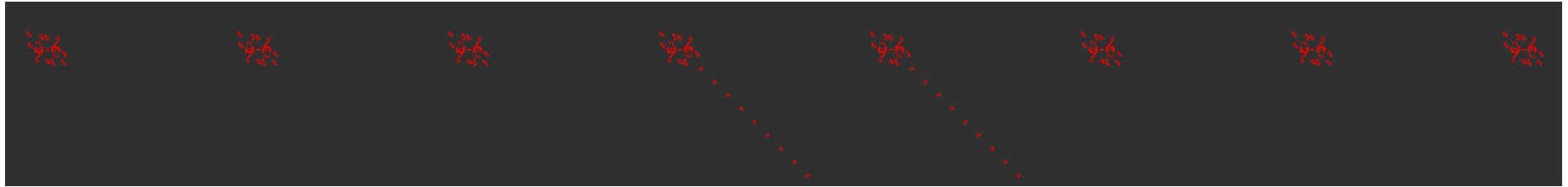
Passe Bande :

69

III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

24



Passe bas :

6



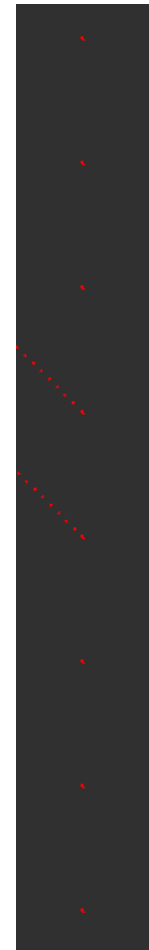
Passe Haut :

1



Passe Bande :

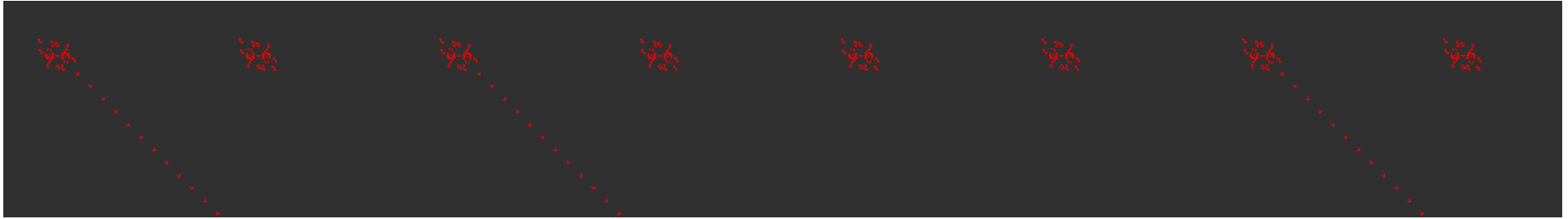
24



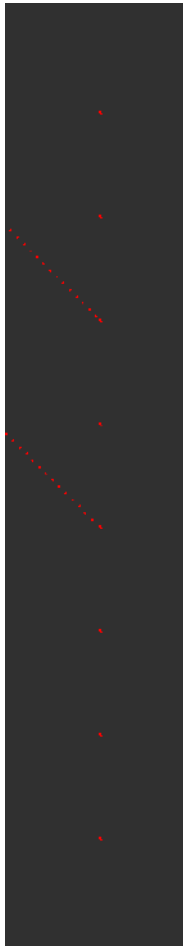
III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

162

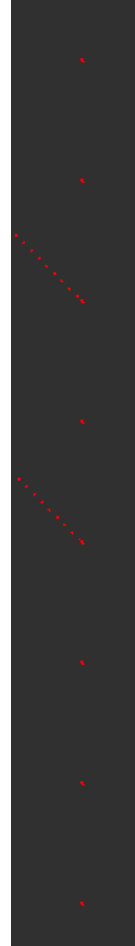


Passe bas :



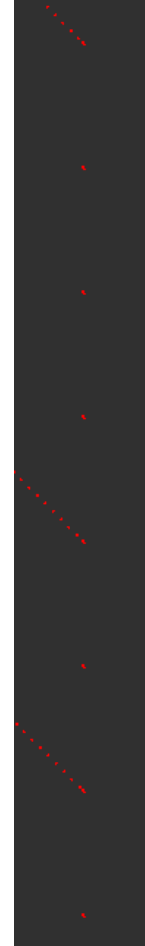
20

Passe Haut :



20

Passe Bande :

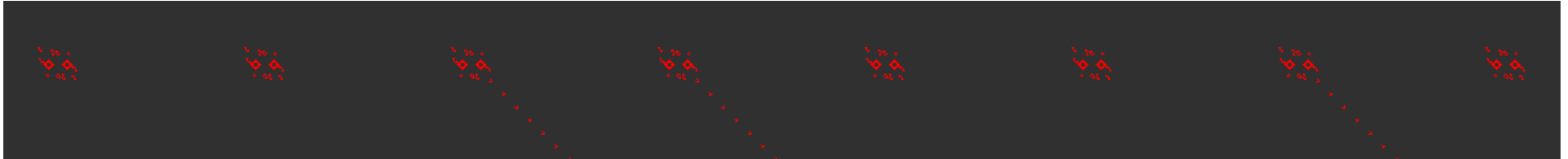


81

III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

50



Passe bas :

3



Passe Haut :

12



Passe Bande :

12



III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

138



Passe bas :

4



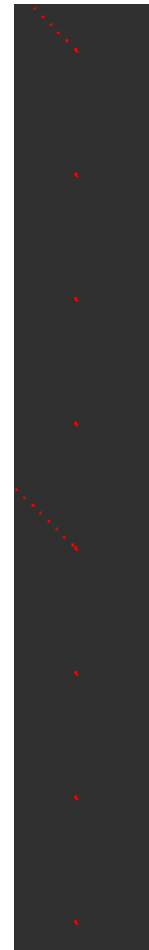
Passe Haut :

69



Passe Bande :

17



III. Experiences et implémentation en C

1. Essai de différentes fréquences fondamentales (Amplitude fixée)
appliqués à ces différents filtres

72



Passe bas :

1

Passe Haut :

72

Passe Bande :

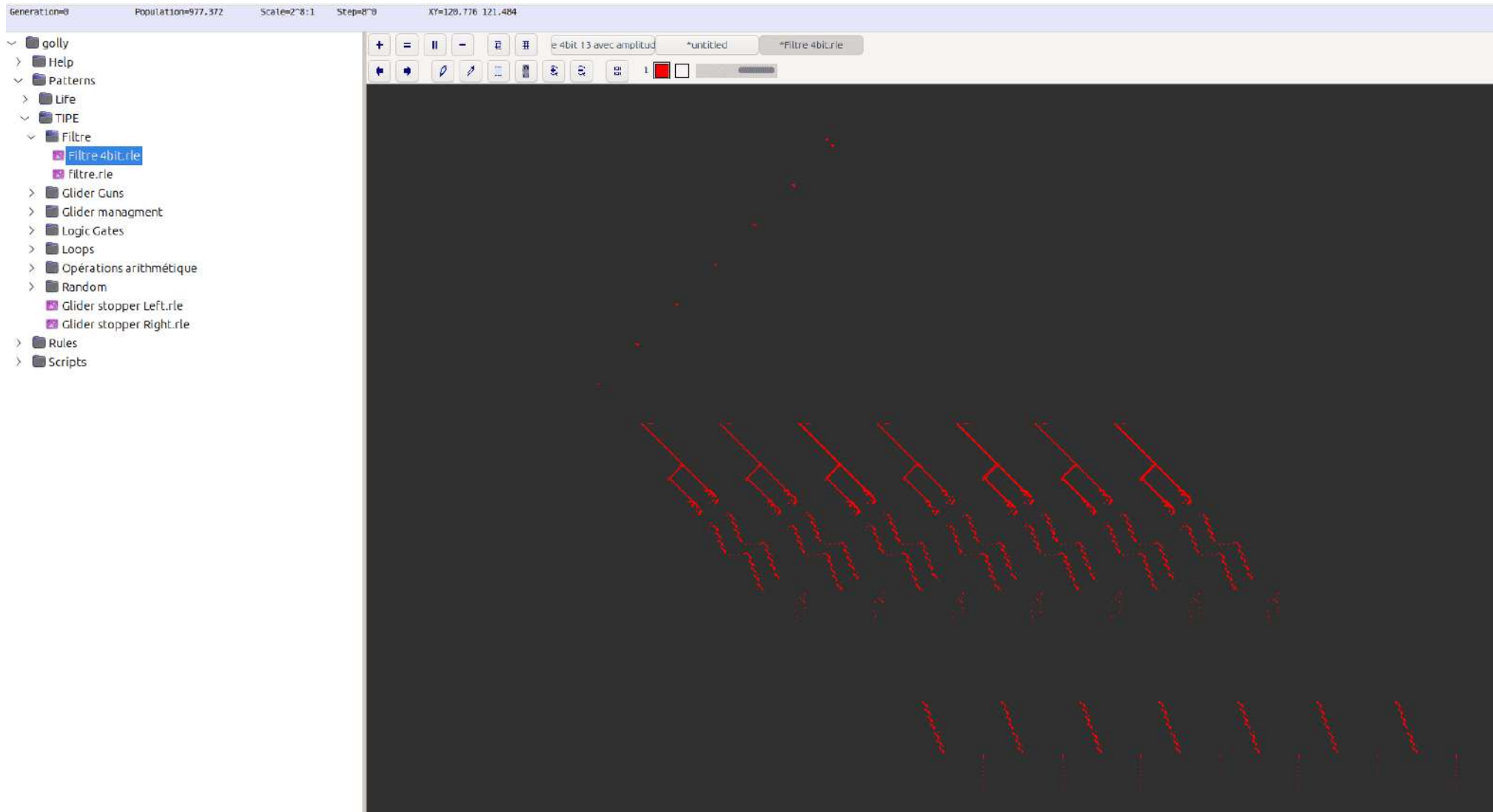
4

III. Experiences et implémentation en C

2. Nombre de générations
nécessaires pour arriver au resultat

III. Experiences et implémentation en C

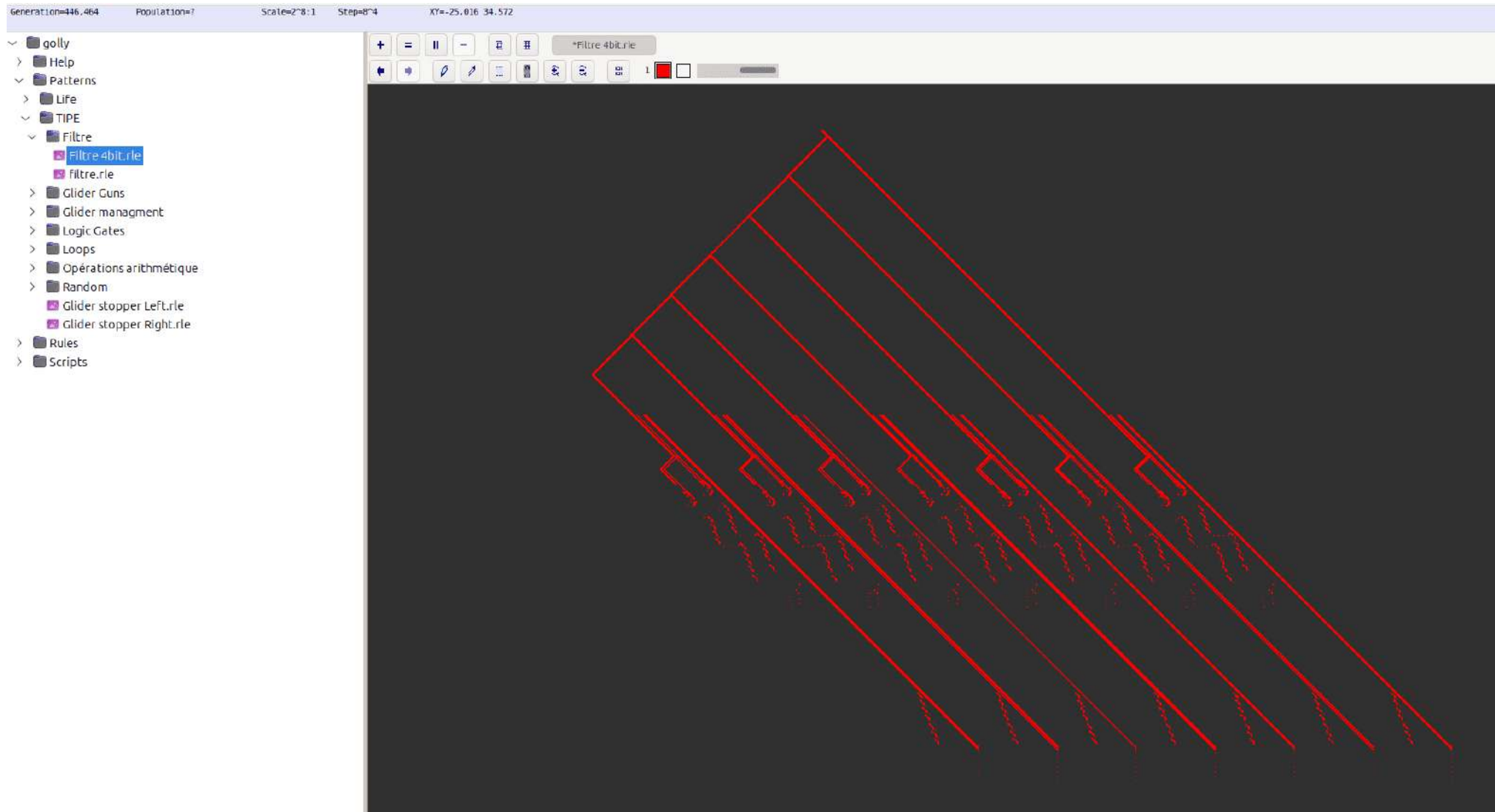
2. Nombre de générations nécessaires pour arriver au resultat



Golly

III. Experiences et implémentation en C

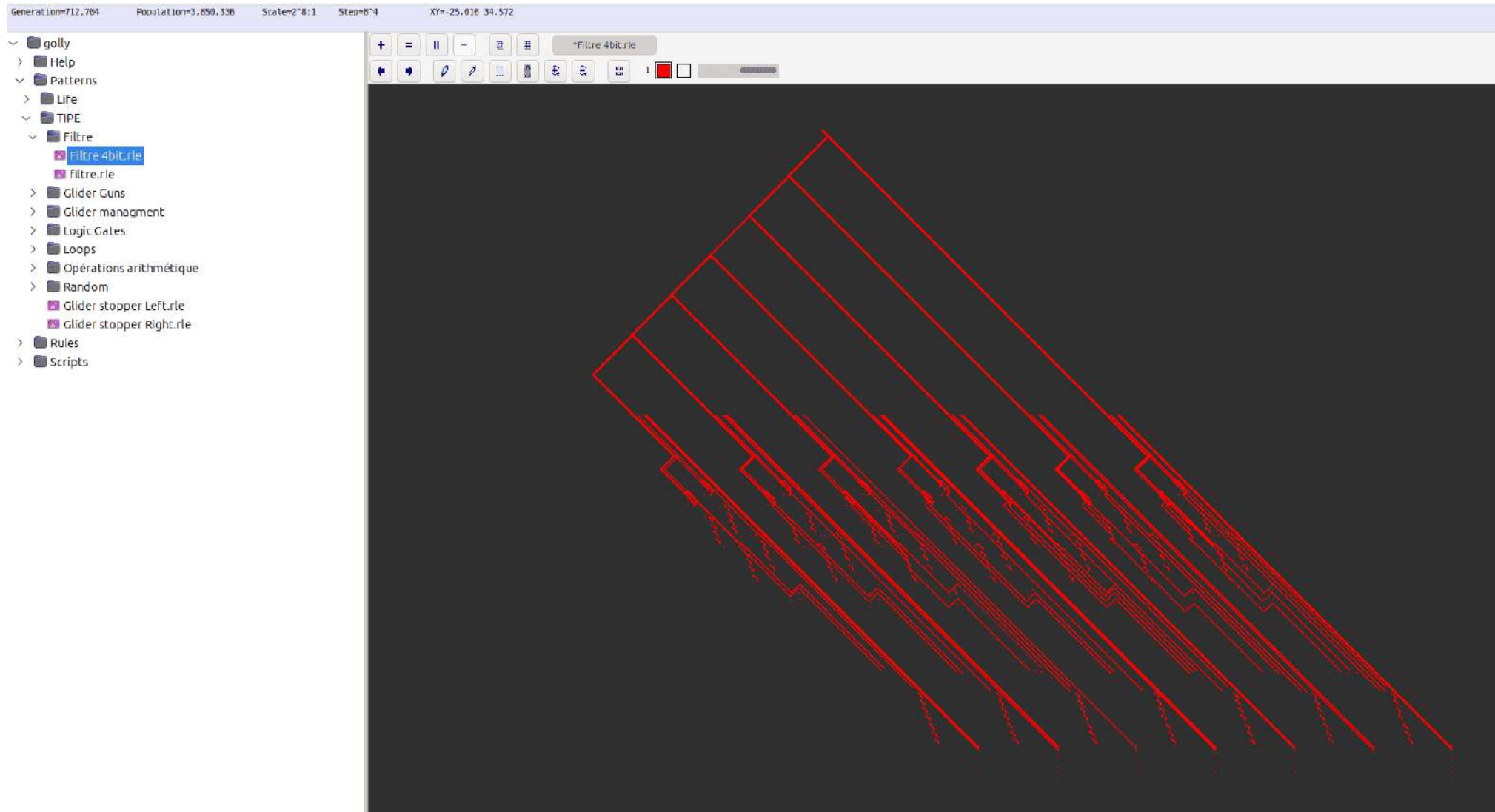
2. Nombre de générations nécessaires pour arriver au resultat



Golly

III. Experiences et implémentation en C

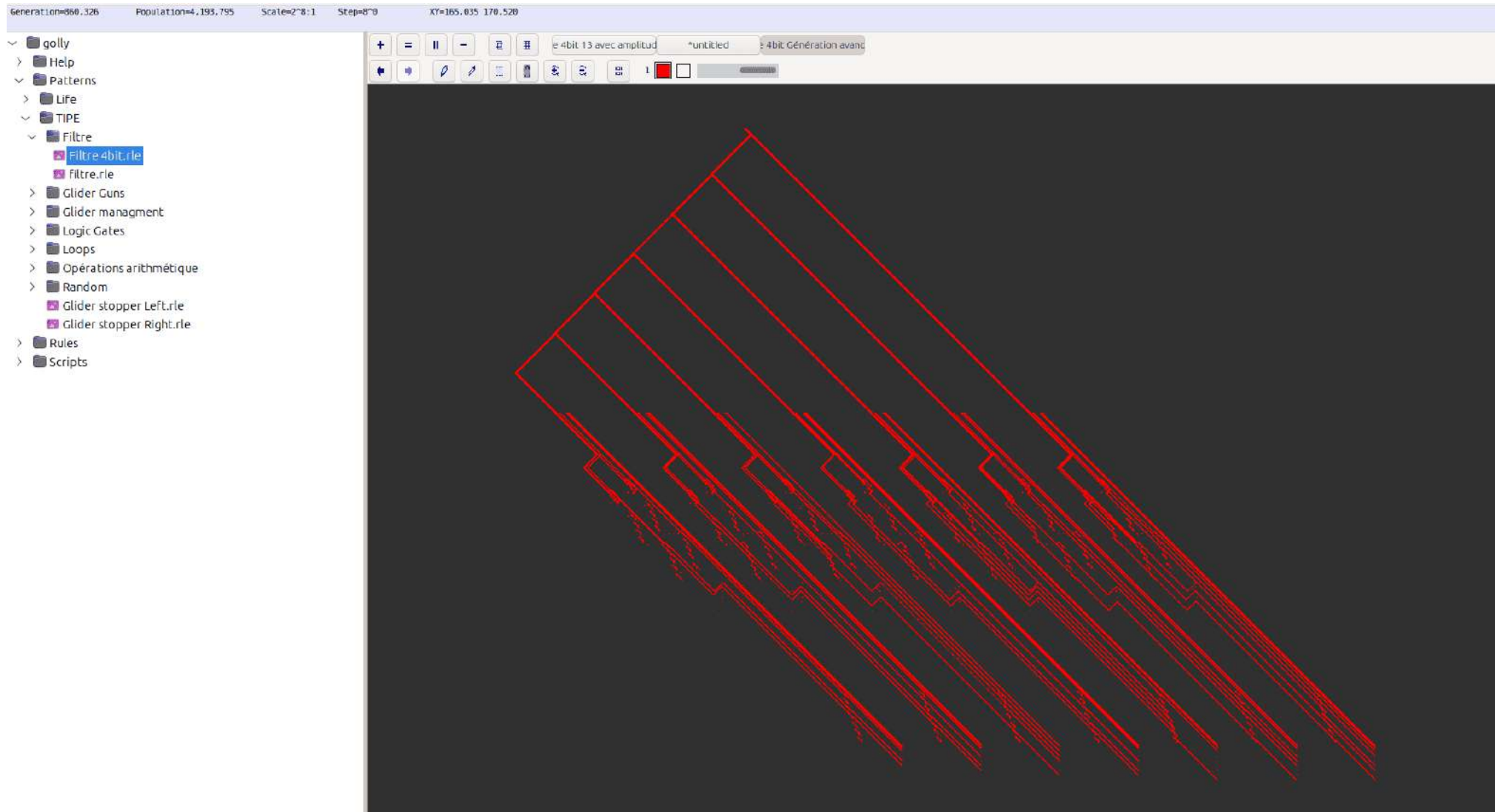
2. Nombre de générations nécessaires pour arriver au resultat



Golly

III. Experiences et implémentation en C

2. Nombre de générations nécessaires pour arriver au resultat



Générations : 860,326

Golly

III. Experiences et implémentation en C

3. Implémentation en C

III. Experiences et implémentation en C

3. Implémentation en C



III. Experiences et implémentation en C

3. Implémentation en C



```
typedef struct var{  
    ... SDL_Window *window;  
    ... SDL_Renderer *renderer;  
    ... SDL_Texture *texture;  
  
    ... int *KEY_DOWN_STATUS;  
    ... matrix *XY_CASE_MAT;  
    ... matrix *coordLivingCells;  
  
    ... SDL_Rect *camera;  
    ... SDL_Rect *gridDestRect;  
  
} Var;
```

III. Experiences et implémentation en C

3. Implémentation en C



```
void evolution(Var *var){  
    ... matrix NEW_XY_CASE_MAT = CREATE_MATRIX(var->XY_CASE_MAT->width, var->XY_CASE_MAT->height);  
    ... for (int i = 0; i < var->XY_CASE_MAT->width; i++){  
        ... for (int j = 0; j < var->XY_CASE_MAT->height; j++){ ...  
            ...  
            ... int nbreCellAutour = nbreCellulesAutour(*var->XY_CASE_MAT, i, j);  
            ... int cellVal = var->XY_CASE_MAT->mat[i][j];  
            ... if (nbreCellAutour == 3)  
            {  
                ... NEW_XY_CASE_MAT.mat[i][j] = 1;  
                ... if (cellVal != 1){  
                    ... DRAW_RECT_EVOLUTION_DISPLAY(var, i, j, 1);  
                }  
            }  
            ... else if (nbreCellAutour == 2)  
            {  
                ... NEW_XY_CASE_MAT.mat[i][j] = cellVal;  
            }  
            ... else if (cellVal == 1){  
                ... DRAW_RECT_EVOLUTION_DISPLAY(var, i, j, 0);  
            }  
        }  
    }  
    ... *var->XY_CASE_MAT = NEW_XY_CASE_MAT;  
}
```


III. Experiences et implémentation en C

3. Implémentation en C

```
elio@Elio-Vivobook:~/Documents/Prepa/MPI/TIPE/Programme$ gcc -o main Lib/*.c main.c -l SDL2_ttf $(sd  
l2-config --cflags --libs) && ./main  
Version : 2.30.0  
ERREUR : Texture dimensions are limited to 16384x16384
```

Filtre :

270,051 x 204,492

FIN

Main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6
7  /* !!! ATTENTION !!! */
8  /* ---- Toutes les fichiers sont dépendants ---- */
9  /* ---- Changer l'ordre mènerait à la perte de toute logique ---- */
10 #include "Lib/GlobalDef.h"
11 #include "Lib/Error.h"
12 #include "Lib/Array.h"
13 #include "Lib/Matrix.h"
14 #include "Lib/Var.h"
15 #include "Lib/Grid.h"
16 #include "Lib/Conversion_RLE.h"
17 #include "Lib/Evolution.h"
18 #include "Lib/eventGestion.h"
19
20
21
22 int main(int argc, char **argv){
23     // ----- Vérifie si il y des erreur sur les définitions ----- //
24     DefinitionError();
25
26     // ----- Initialisation de la fenetre, du rendu et des textures ----- //
27     SDL_Window *window = NULL;
28     SDL_Renderer *renderer = NULL;
29     SDL_Texture *texture = NULL;
30
31
32
33     SDL_version nb;
34     SDL_VERSION(&nb);
35     printf("Version : %d.%d.%d \n", nb.major, nb.minor, nb.patch);
36
37     // Initialisation
38     VERIF_SDL_COMMAND(SDL_Init(SDL_INIT_VIDEO), "INIT_VIDEO"); // SDL_INIT_VIDEO | SDL_INIT_AUDIO
39 }
```

Main.c

```
45 //----- Creation de la fenetre :-----//
46
47
48 window = SDL_CreateWindow("Fenêtre Titree", 70, 0, RENDER_WIDTH, RENDER_HEIGHT, 0); // Pour le dernier on peut mettre un flag : SDL_WINDOW_FULLSCREEN par exemple
49 if(window == NULL){
50     ExitWithError("Window creation failed");
51 }
52
53 //----- Creation du rendu ----- //:
54 renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_TARGETTEXTURE);
55 if(renderer == NULL){
56     ExitWithError("Renderer creation failed");
57 }
58
59
60 /* Flags (pour le deuxième argument de create renderer) :
61 SDL_RENDERER_SOFTWARE (plus le proc)
62 SDL_RENDERER_ACCELERATED (plus pour la carte graphique MDRRR)
63 SDL_RENDERER_PRESENTVSYNC (TT est dans le nom)
64 SDL_RENDERER_TARGETTEXTURE (jsp)
65 */
66
67 /*
68 ----- Sinon pour la création de la fenetre et du rendu :-----
69 if(SDL_CreateWindowAndRender(800, 600, &window, &renderer)!=0){
70     SDL_ExitWithError("Window or renderer creation failed");
71 }
72 */
73
74 //----- Création de la texture ----- //
75 texture = SDL_CreateTexture(renderer, PIXEL_FORMAT, TEXTURE_ACCESS, TEXTURE_WIDTH, TEXTURE_HEIGHT);
76 if(texture == NULL){
77     ExitWithError(SDL_GetError());
78 }
79
80
81 //----- Création des rectangles ----- //
82 SDL_Rect *camera = malloc(sizeof(SDL_Rect));
83 SDL_Rect *gridDestRect = malloc(sizeof(SDL_Rect));
84
85
86
87 camera->x = 0;
88 camera->y = 0;
89 camera->w = GRID_DISP_WIDTH;
90 camera->h = GRID_DISP_HEIGHT;
91
92 gridDestRect->x = 0;
93 gridDestRect->y = 0;
94 gridDestRect->w = GRID_DISP_WIDTH;
95 gridDestRect->h = GRID_DISP_HEIGHT;
96
```

Main.c

```
98 ... // ----- Déclaration des variables ----- //
99 ... SDL_bool program_launched = SDL_TRUE;
100
101 ... // Tableau des touches abaissées
102 ... int * KEY_DOWN_STATUS = NULL;
103 ... KEY_DOWN_STATUS = CREATE_TAB_0(315);
104
105
106 ... // Variable pour savoir si la souris est pressée
107 ... bool* isMouseButtonPressed = malloc(sizeof(bool));
108 ... *isMouseButtonPressed = false;
109
110 ... // savoir si la souris à précédement été bougée
111 ... bool* MOUSE_MOVING = malloc(sizeof(bool));
112 ... *MOUSE_MOVING = false;
113
114
115 ... // Matrice des cases de la grille
116
117 ... matrix matmp = CREATE_MATRIX(CASE_NUMBER_WIDTH, CASE_NUMBER_HEIGHT);
118 ... matrix *XY_CASE_TAB = malloc(sizeof(matrix));
119 ... XY_CASE_TAB = &matmp;
120
121
122
123 ... // ----- Var ----- //
124 ... Var *var = malloc(sizeof(Var));
125
126 ... MALLOC_VAR(var);
127 ... var->>window = window;
128 ... var->renderer = renderer;
129 ... var->texture = texture;
130 ... var->KEY_DOWN_STATUS = KEY_DOWN_STATUS;
131 ... var->XY_CASE_MAT = XY_CASE_TAB;
132 ... var->camera = camera;
133 ... var->gridDestRect = gridDestRect;
134 ...
```


Main.c

```
123  ...//-----Var-----//
124  ...Var *var = malloc(sizeof(var));
125  ...
126  ...MALLOC_VAR(var);
127  ...var->window = window;
128  ...var->renderer = renderer;
129  ...var->texture = texture;
130  ...var->KEY_DOWN_STATUS = KEY_DOWN_STATUS;
131  ...var->XY_CASE_MAT = XY_CASE_TAB;
132  ...var->camera = camera;
133  ...var->gridDestRect = gridDestRect;
134  ...
135
136  ...while (program_launched)
137  ...{
138  ...
139  ...    SDL_Event *event = malloc(sizeof(SDL_Event));
140  ...    program_launched = GESTION(var, *event, isMouseButtonPressed, MOUSE_MOVING);
141  ...
142  ...    // Actualise le rendu
143  ...    VERIF_SDL_COMMAND(SDL_RenderCopy(var->renderer, var->texture, var->camera, var->gridDestRect), "RenderCopy");
144  ...    SDL_RenderPresent(var->renderer);
145  ...
146  ...
147  ...}
148  ...printf("quit.\n");
149
150
151  ...
152  ...//----- Clear le rendu + vérif erreur -----//
153  ...VERIF_SDL_COMMAND(SDL_RenderClear(var->renderer), "RenderClear");
154  ...
155
156
157
158  ...// FREE //
159  ...DESTROY_VAR(var);
160  ...
161
162  ...SDL_Quit();
163  ...return EXIT_SUCCESS;
164  }
165
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
```

Array.h

```
1  #ifndef ARRAY_H  
2  #define ARRAY_H  
3  int * CREATE_TAB_0(const int len);  
4  
5  #endif
```

Array.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16 int * CREATE_TAB_0(const int len){
17     int *tab = malloc(len * sizeof(int));
18     if(tab == NULL){
19         ExitWithError("Space allocation failed");
20     }
21
22     for(int i = 0; i < len; i++)
23     {
24         tab[i] = 0;
25     }
26
27     return tab;
28 }
```


Conversion_RLE.h

```
1  #ifndef CONVERSION_RLE_H
2  #define CONVERSION_RLE_H
3
4  void ajouter_fin_chaine(char *chaine, char caractere);
5  int string_to_int(char *chaine);
6  void AJOUTER_CELLULE(Var *var, int CASE_X, int CASE_Y);
7  void CONVERSION_RLE(Var *var, char* file_name);
8
9  #endif
```

Conversion_RLE.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16 void ajouter_fin_chaine(char *chaine, char caractere){
17     /*Ajoute caractere à la fin de chaine*/
18     size_t len = strlen(chaine);
19     chaine[len] = caractere;
20     chaine[len+1] = '\0';
21 }
22
23 int string_to_int(char *chaine){
24     /*Converti chaine en un entier qui est renvoyé*/
25     int nombre;
26     nombre = atoi(chaine);
27
28     return nombre;
29 }
30
31 void AJOUTER_CELLULE(Var *var, int CASE_X, int CASE_Y){
32     /*Ajoute une cellule vivante dans la grille et dans le tableau*/
33     if ((0 > CASE_X || CASE_X > CASE_NUMBER_WIDTH) || (0 > CASE_Y || CASE_Y > CASE_NUMBER_HEIGHT))
34     {
35         ExitWithError("AJOUTER_CELLULE ");
36     }
37     else
38     {
39         CASE_CLICK_CASE_DISPLAY(var, CASE_X, CASE_Y);
40     }
41
42
43 }
44
45
46
```

Conversion_RLE.c

```
47 void CONVERSION_RLE(Var *var, char* file_name){
48     /*
49     ... Algorithme de conversion d'un fichier RLE jusqu'à l'implémentation en C
50     ... file_name : nom du fichier à ouvrir avec son extension
51
52     ... ATTENTION NE PAS METTRE D'ESPACE DANS LA CHAÎNE file_name
53     ... */
54     if (file_name == NULL){
55         ExitWithError("NOM DU FICHIER RLE NON DÉFINI");
56     }
57
58     FILE *file = fopen(file_name, "r");
59     if (file == NULL){
60         ExitWithError("FICHIER RLE NON OUVERT");
61     }
62     else{
63         char caractere;
64         char ligne[255];
65         int nombre;
66         int CASE_X = 10;
67         int CASE_Y = 10;
68         caractere = fgetc(file);
69
70
71         while (caractere != '!'){
72             if(caractere == 'x' || caractere == '#'){
73                 fgets(ligne, 255, file);
74                 caractere = fgetc(file);
75
76             }
77             else{
78                 char nombre_str[1000] = "0";
79
80                 while (caractere != 'o' && caractere != 'b' && caractere != '$' && caractere != '\n' && caractere != '!')
81                 {
82                     ajouter_fin_chaine(nombre_str, caractere);
83                     caractere = fgetc(file);
84                 }
85             }
86         }
87     }
88 }
```

Conversion_RLE.c

```
87 .....
88 ..... nombre = string_to_int(nombre_str);
89 .....
90 .....
91 ..... if (caractere == 'o'){
92 .....     if (nombre == 0)
93 .....     {
94 .....         AJOUTER_CELLULE(var, CASE_X, CASE_Y);
95 .....         CASE_X++;
96 .....     }
97 .....     else{
98 .....         for (int i = 0; i < nombre; i++)
99 .....         {
100 .....             AJOUTER_CELLULE(var, CASE_X, CASE_Y);
101 .....             CASE_X++;
102 .....         }
103 .....     }
104 .....
105 ..... }
106 ..... else if (caractere == 'b'){
107 .....     if (nombre == 0)
108 .....     {
109 .....         CASE_X++;
110 .....     }
111 .....     else
112 .....     {
113 .....         CASE_X += nombre;
114 .....     }
115 ..... }
116 ..... else if (caractere == '$'){
117 .....     CASE_X = 10;
118 .....     if (nombre == 0){
119 .....         CASE_Y++;
120 .....     }
121 .....     else{
122 .....         CASE_Y += nombre;
123 .....     }
124 ..... }
125 .....
126 ..... caractere = fgetc(file);
127 .....
128 .....
129 .....
130 .....
131 ..... }
132 .....
133 ..... }
134 .....
135 ..... fclose(file);
136 ..... }
137 .....
138 .....
139 ..... }
140 .....
```

Error.h

```
1  #ifndef ERROR_H
2  #define ERROR_H
3
4  void SDL_ExitWithError(const char *message);
5  void ExitWithError(const char *message);
6  void VERIF_SDL_COMMAND(int command, char *nameCommand);
7  void DefinitionError();
8
9  #endif
```

Error.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16 void SDL_ExitWithError(const char *message){
17     SDL_Log("ERREUR : %s -> %s\n", message, SDL_GetError());
18     exit(EXIT_FAILURE);
19 }
20
21 void ExitWithError(const char *message){
22     printf("ERREUR : %s\n", message);
23     exit(EXIT_FAILURE);
24 }
25
26
27 void VERIF_SDL_COMMAND(int command, char *nameCommand){
28     if(command != 0){
29         SDL_ExitWithError(strcat("%s failed !\n", nameCommand));
30     }
31 }
32
33
34
35 void DefinitionError(){
36     if(TEXTURE_HEIGHT < RENDER_HEIGHT){
37         ExitWithError("La taille du rendu est supérieure au nombre de carrés que l'on veut afficher (Effet : strech texture)");
38     }
39 }
40
```

eventGestion.h

```
1  #ifndef EVENT_GESTION_H
2  #define EVENT_GESTION_H
3
4  SDL_bool GESTION(Var *var, SDL_Event event, bool *isMouseButtonPressed, bool *MOUSE_MOVING);
5
6  #endif
```



```

16 int time;
17 int velocity = 500;
18
19
20 SDL_bool GESTION(Var *var, SDL_Event event, bool* isMouseButtonPressed, bool* MOUSE_MOVING){
21     while (SDL_PollEvent(&event)){ // Elle va lire tout les évènements
22         //on a aussi : SDL_WaitEvent(&event) mais ca bloque la fenêtre*/
23         switch (event.type) // Le switch c'est comme un match en ocaml mais que pour les int
24         {
25
26             case SDL_KEYDOWN :
27
28                 switch (event.key.keysym.sym)
29                 {
30
31                     case SDLK_SPACE:
32                         var->KEY_DOWN_STATUS[SDLK_b] = 1;
33                         evolution(var);
34                         continue;
35
36                     case SDLK_RETURN:
37                         // Trigger
38                         if (var->KEY_DOWN_STATUS[SDLK_RETURN] == 0){
39                             var->KEY_DOWN_STATUS[SDLK_RETURN] = 1;
40                         }
41                         else{
42                             var->KEY_DOWN_STATUS[SDLK_RETURN] = 0;
43                         }
44                         // INITIALISATION DU TIMESTAMP
45                         time = SDL_GetTicks();
46
47                         continue;
48
49                     case SDLK_KP_PLUS:
50                         if(velocity >= 50){
51                             velocity -= 50;
52                         }
53                         continue;
54
55
56                     case SDLK_KP_MINUS:
57                         velocity += 50;
58                         continue;
59
60                     case SDLK_c:
61                         var->KEY_DOWN_STATUS[SDLK_c] = 1;
62                         GRID_DISPLAY_CREATION(var);
63                         continue;
64
65                     case SDLK_o:
66                         CONVERSION_RLE(var, "RLE_FILES/main.rle");
67                         continue;
68
69                     case SDLK_LCTRL:
70                         var->KEY_DOWN_STATUS[306] = 1; // SDLK_LCTRL = 306
71                         continue;
72

```

eventGestion.c


```

73 ..... case SDLK_z:
74 .....     var->KEY_DOWN_STATUS[306] = 1;
75 .....     if (var->KEY_DOWN_STATUS[306] == 1){
76 .....         // REMETTRE LA GRILLE AU TABLEAU ENTREGISTRÉ LORS de l'activation de ENTER
77 .....     }
78 .....     continue;
79 .....
80 .....
81 .....     default:
82 .....
83 .....         continue;
84 .....     }
85 .....
86 ..... case SDLK_KEYUP :
87 .....     switch (event.key.keysym.sym)
88 .....     {
89 .....
90 .....         case SDLK_b:
91 .....             var->KEY_DOWN_STATUS[SDLK_b] = 0;
92 .....             continue;
93 .....
94 .....         case SDLK_c:
95 .....             var->KEY_DOWN_STATUS[SDLK_c] = 0;
96 .....             continue;
97 .....
98 .....
99 .....         case SDLK_q:
100 .....             var->KEY_DOWN_STATUS[SDLK_q] = 0;
101 .....             continue;
102 .....
103 .....         case SDLK_d:
104 .....             var->KEY_DOWN_STATUS[SDLK_d] = 0;
105 .....             continue;
106 .....
107 .....         case SDLK_z:
108 .....             var->KEY_DOWN_STATUS[SDLK_z] = 0;
109 .....             continue;
110 .....
111 .....         case SDLK_s:
112 .....             var->KEY_DOWN_STATUS[SDLK_s] = 0;
113 .....             continue;
114 .....
115 .....         case SDLK_LCTRL:
116 .....             var->KEY_DOWN_STATUS[306] = 0;
117 .....             continue;
118 .....
119 .....
120 .....         default:
121 .....             continue;
122 .....     }
123 .....
124 .....
125 ..... case SDL_MOUSEMOTION :
126 .....     //printf("Coordonnée : (%d, %d) \n", event.motion.x, event.motion.y);
127 .....     //printf("Vitesse : (%d, %d) \n", event.motion.xrel, event.motion.yrel);
128 .....

```

eventGestion.c

```

124 .....
125 ..... case SDL_MOUSEMOTION :
126 ..... //printf("Coordonnée : (%d, %d) \n", event.motion.x, event.motion.y);
127 ..... //printf("Vitesse : (%d, %d) \n", event.motion.xrel, event.motion.yrel);
128 .....
129 ..... // ----- Actualisation du déplacement de la caméra avec la souris ----- //
130 ..... if(*isMouseButtonPressed && event.motion.x <= GRID_DISP_WIDTH && event.motion.y <
131 ..... MOVE_CAMERA_MOUSE(var->renderer, var->texture, var->camera, event.motion.xrel
132 ..... *MOUSE_MOVING = true;
133 ..... }
134 .....
135 ..... continue;
136 .....
137 ..... case SDL_MOUSEBUTTONDOWN :
138 ..... *isMouseButtonPressed = true;
139 ..... //printf("x : %d, y : %d \n", event.motion.x, event.motion.y);
140 ..... //printf("Case x : %d, Case y : %d \n", GET_CASE_FROM_COORD_X(event.motion.x), GE
141 .....
142 ..... continue;
143 .....
144 ..... case SDL_MOUSEBUTTONUP :
145 ..... *isMouseButtonPressed = false;
146 ..... if(!(*MOUSE_MOVING))
147 ..... {
148 ..... CASE_CLICK_DISPLAY(var, event.motion.x, event.motion.y);
149 ..... }
150 ..... *MOUSE_MOVING = false;
151 .....
152 ..... continue;
153 .....
154 ..... case SDL_QUIT :
155 ..... return SDL_FALSE;
156 ..... break;
157 .....
158 .....
159 ..... default:
160 ..... break;
161 .....
162 .....
163 ..... }
164 .....
165 .....
166 .....
167 ..... }
168 .....
169 ..... /* Evolution continue */
170 ..... if(var->KEY_DOWN_STATUS[SDLK_RETURN] == 1 && (SDL_GetTicks() - time > velocity)){.....
171 ..... evolution(var);
172 ..... time = SDL_GetTicks();
173 ..... }
174 .....
175 .....
176 .....
177 ..... return SDL_TRUE;
178 ..... }
179 .....

```

eventGestion.c

Evolution.h

```
1  #ifndef EVOLUTION_H
2  #define EVOLUTION_H
3
4  void evolution(Var *var);
5  int nbreCellulesAutour(matrix mat, int X, int Y);
6  void test(Var *var);
7  ...
8  #endif
```

Evolution.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4 #include <string.h>
5 #include <SDL.h>
6 #include "GlobalDef.h"
7 #include "Error.h"
8 #include "Array.h"
9 #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16 int nbreCellulesAutour(matrix mat, int X, int Y){
17     int nbre = 0;
18     for (int l = -1; l < 2; l++){
19         {
20             if ((0 <= X+l) && (X+l < mat.width))
21             {
22                 for (int k = -1; k < 2; k++){
23                     {
24                         if ((0 <= Y+k) && (Y+k < mat.height))
25                         {
26                             if (!(l == 0 && k == 0)){
27                                 nbre += mat.mat[X+l][Y+k];
28                             }
29                         }
30                     }
31                 }
32             }
33         }
34     }
35     return nbre;
36 }
37
38 void evolution(Var *var){
39     matrix NEW_XY_CASE_MAT = CREATE_MATRIX(var->XY_CASE_MAT->width, var->XY_CASE_MAT->height);
40     for (int i = 0; i < var->XY_CASE_MAT->width; i++){
41         for (int j = 0; j < var->XY_CASE_MAT->height; j++){
42             int nbreCellAutour = nbreCellulesAutour(*var->XY_CASE_MAT, i, j);
43             int cellVal = var->XY_CASE_MAT->mat[i][j];
44             if (nbreCellAutour == 3)
45             {
46                 NEW_XY_CASE_MAT.mat[i][j] = 1;
47                 if (cellVal != 1){
48                     DRAW_RECT_EVOLUTION_DISPLAY(var, i, j, 1);
49                 }
50             }
51             else if (nbreCellAutour == 2)
52             {
53                 NEW_XY_CASE_MAT.mat[i][j] = cellVal;
54             }
55             else if (cellVal == 1){
56                 DRAW_RECT_EVOLUTION_DISPLAY(var, i, j, 0);
57             }
58         }
59     }
60     *var->XY_CASE_MAT = NEW_XY_CASE_MAT;
61 }
62 }
```

```

1  #ifndef GLOBAL_DEF_H
2  #define GLOBAL_DEF_H
3
4  /*----- Variables Globales et Définition préprocesseur -----*/
5  /* Pour que le rendu s'adapte parfaitement aux cases (qu'elles soient pas coupés)
6  --- Pour 20 pixel d'espace mettre : 1840, 1120
7  --- Pour 25 pixel d'espace mettre : 1850, 1125
8  --- Pour 30 pixel d'espace mettre : 1830, 1110 */
9
10 // Longueur et hauteur du rendu
11 #ifndef RENDER_WIDTH
12 #define RENDER_WIDTH 3775
13 #endif
14
15 #ifndef RENDER_HEIGHT
16 #define RENDER_HEIGHT 1180
17 #endif
18
19 // Nombre de pixels de large pour une case
20 #ifndef SQUARE_WIDTH
21 #define SQUARE_WIDTH 10
22 #endif
23
24 // Longueur et hauteur de l'affichage de la grille
25 #ifndef GRID_DISP_WIDTH
26 #define GRID_DISP_WIDTH RENDER_WIDTH
27 #endif
28
29 #ifndef GRID_DISP_HEIGHT
30 #define GRID_DISP_HEIGHT RENDER_HEIGHT
31 #endif
32
33
34 // Nombre de cases en longueur de l'écran
35 #ifndef CASE_NUMBER_WIDTH
36 #define CASE_NUMBER_WIDTH 1638
37 #endif
38
39 // Nombre de cases en hauteur de l'écran
40 #ifndef CASE_NUMBER_HEIGHT
41 #define CASE_NUMBER_HEIGHT 1638
42 #endif
43
44 // Longueur et hauteur de la texture
45 #ifndef TEXTURE_WIDTH
46 #define TEXTURE_WIDTH (CASE_NUMBER_WIDTH*SQUARE_WIDTH)
47 #endif
48
49 #ifndef TEXTURE_HEIGHT
50 #define TEXTURE_HEIGHT (CASE_NUMBER_HEIGHT*SQUARE_WIDTH)
51 #endif
52
53 // Variables générales de la texture
54 #ifndef PIXEL_FORMAT
55 #define PIXEL_FORMAT SDL_PIXELFORMAT_RGBA32
56 #endif
57
58 #ifndef TEXTURE_ACCESS
59 #define TEXTURE_ACCESS SDL_TEXTUREACCESS_TARGET
60 #endif
61
62 #endif

```

GlobalDef.h

Grid.h

```
1  #ifndef GRID_H
2  #define GRID_H
3
4  /* Fonction qui crée la grille de départ */
5  void CREATE_GRID(SDL_Renderer *renderer, const int X, const int Y);
6
7  /* Fonction qui crée la grille de départ à partir des cases */
8  void CREATE_GRID_CASE(SDL_Renderer *renderer, const int caseX, const int caseY);
9
10 /* Fonction qui affiche la grille sur la texture du rendu voulue */
11 void GRID_DISPLAY_CREATION(Var *var);
12
13 /* Fonction qui affiche la case en blanc ou en noir quand on clique dessus */
14 void CASE_CLICK_DISPLAY(Var *var, const int x, const int y);
15
16 /* Bouge la grille quand on click + bouge avec la souris */
17 void MOVE_CAMERA_MOUSE(SDL_Renderer *renderer, SDL_Texture *texture, SDL_Rect *camera, int xRel, int yRel);
18
19 void CASE_CLICK_COORDS(Var *var, const int x, const int y);
20 void CASE_CLICK_DISPLAY(Var *var, const int x, const int y);
21
22 void CASE_CLICK_CASE(Var *var, const int CASE_X, const int CASE_Y);
23 void CASE_CLICK_CASE_DISPLAY(Var *var, const int CASE_X, const int CASE_Y);
24
25 void DRAW_RECT_EVOLUTION(Var *var, const int CASE_X, const int CASE_Y, int STATE);
26 void DRAW_RECT_EVOLUTION_DISPLAY(Var *var, const int CASE_X, const int CASE_Y, int STATE);
27
28 int GET_CASE_FROM_COORD_X(int x);
29 int GET_CASE_FROM_COORD_Y(int y);
30
31 #endif
```

Grid.c

```
16 int color = 0;
17
18 void CREATE_GRID(SDL_Renderer *renderer, const int X, const int Y){
19     ... VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(renderer, 255-color, 255-color, 255-color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor");
20     ... SDL_RenderClear(renderer);
21     ... VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(renderer, color, color, color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor"); // On peut mettre 255 a la place de SDL opaque
22     ...
23     ...
24
25     ... for (int i = 0; i <= TEXTURE_WIDTH/X; i++){
26         ... VERIF_SDL_COMMAND(SDL_RenderDrawLine(renderer, i*X, 0, i*X, TEXTURE_HEIGHT), "RenderDrawLine");
27     }
28
29     ... for (int j = 0; j <= TEXTURE_HEIGHT/Y; j++){
30         ... VERIF_SDL_COMMAND(SDL_RenderDrawLine(renderer, 0, j*Y, TEXTURE_WIDTH, j*Y), "RenderDrawLine");
31     }
32
33     ... SDL_RenderPresent(renderer);
34 }
35
36
37 void CREATE_GRID_CASE(SDL_Renderer *renderer, const int caseX, const int caseY){
38     ... CREATE_GRID(renderer, SQUARE_WIDTH, SQUARE_WIDTH);
39
40 }
41
42
43 void GRID_DISPLAY_CREATION(Var *var){
44     ...
45     ... // On met en cible du dessin la texture
46     ... VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, var->texture), "SetRenderTarget");
47     ... // On nettoie entièrement la texture du rendu
48     ... VERIF_SDL_COMMAND(SDL_RenderClear(var->renderer), "RenderClear");
49
50     ... CREATE_GRID_CASE(var->renderer, CASE_NUMBER_WIDTH, CASE_NUMBER_HEIGHT);
51
52     ... // On remet la cible du rendu sur la fenêtre en entier
53     ... VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, NULL), "SetRenderTarget");
54 }
55
56
57 // Variables globales utilisées : RENDER_WIDTH, CASE_NUMBER_WIDTH
58 int GET_CASE_FROM_COORD_X(int x){
59     ... /* Fonction qui renvoie le numéro de la case à la coordonnée x */
60     ... return x/SQUARE_WIDTH;
61 }
62
63 // Variables globales utilisées : RENDER_HEIGHT, CASE_NUMBER_HEIGHT
64 int GET_CASE_FROM_COORD_Y(int y){
65     ... /* Fonction qui renvoie le numéro de la case à la coordonnée y */
66     ... return y/SQUARE_WIDTH;
67
68 }
69
```

Grid.c

```
70 void DRAW_RECT_EVOLUTION(Var *var, const int CASE_X, const int CASE_Y, int STATE){
71     /* Fonction spécifique à l'évolution qui dessine un rectangle dans la case
72     ...
73     ... CASE_X
74     ... CASE_Y
75     ... STATE : État de la cellule a la case : (CASE_X, CASE_Y) que l'on veut dessiner
76     ... */
77     if(STATE == 0){
78         // On change la couleur du rectangle en noir
79         VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(var->renderer, 255-color, 255-color, 255-color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor"); // On peut mettre 255 a la place de SDL opaque
80     }
81     else{
82         // On change la couleur du rectangle en blanc
83         VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(var->renderer, color, color, color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor"); // On peut mettre 255 a la place de SDL opaque
84     }
85     ...
86
87     // Création graphique du rectangle
88     // Les +-1 servent a ne pas redessiner sur la ligne de la grille
89     SDL_Rect *rectangle = malloc(sizeof(SDL_Rect));
90     rectangle->x = CASE_X*SQUARE_WIDTH+1;
91     rectangle->y = CASE_Y*SQUARE_WIDTH+1;
92     rectangle->w = SQUARE_WIDTH-1;
93     rectangle->h = SQUARE_WIDTH-1;
94
95     VERIF_SDL_COMMAND(SDL_RenderFillRect(var->renderer, rectangle), "RenderFillRect");
96
97     SDL_RenderPresent(var->renderer);
98
99     free(rectangle);
100 }
101
102
103 void DRAW_RECT_EVOLUTION_DISPLAY(Var *var, const int CASE_X, const int CASE_Y, int STATE){
104     /* Fonction spécifique à l'évolution qui affiche un rectangle dans la case*/
105     // On met en cible du dessin la texture
106     VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, var->texture), "SetRenderTarget");
107
108     DRAW_RECT_EVOLUTION(var, CASE_X, CASE_Y, STATE);
109
110     // On remet la cible du rendu sur la fenêtre en entier
111     VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, NULL), "SetRenderTarget");
112
113 }
114
```


Grid.c

```
117 void CASE_CLICK_CASE(Var.*var, const int CASE_X, const int CASE_Y){
118     ..../* Fonction qui dessine un rectangle dans la case */
119     ....
120
121     ....// Ajout de la case dans la matrice
122     ....if((0 <= CASE_X <= (var->XY_CASE_MAT->width)) && (0 <= CASE_Y <= (var->XY_CASE_MAT->height))){
123     ....    int Case = var->XY_CASE_MAT->mat[CASE_X][CASE_Y];
124     ....    if(Case == 0){
125     ....        ....var->XY_CASE_MAT->mat[CASE_X][CASE_Y] = 1;
126
127
128
129     ....        ....// On change la couleur du rectangle en blanc (ou noir suivant le background)
130     ....        ....VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(var->renderer, color, color, color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor"); // On peut mettre 255 a la place de SDL opaque
131
132     ....    }
133     ....    else{
134     ....        ....var->XY_CASE_MAT->mat[CASE_X][CASE_Y] = 0;
135
136     ....        ....// On change la couleur du rectangle en noir (ou blanc suivant le background)
137     ....        ....VERIF_SDL_COMMAND(SDL_SetRenderDrawColor(var->renderer, 255-color, 255-color, 255-color, SDL_ALPHA_OPAQUE), "SetRenderDrawColor"); // On peut mettre 255 a la place de SDL opaque
138     ....    }
139
140
141
142     ....// Création graphique du rectangle
143     ....// Les +-1 servent a ne pas redessiner sur la ligne de la grille
144     ....SDL_Rect.*rectangle = malloc(sizeof(SDL_Rect));
145     ....rectangle->x = CASE_X*SQUARE_WIDTH+1;
146     ....rectangle->y = CASE_Y*SQUARE_WIDTH+1;
147     ....rectangle->w = SQUARE_WIDTH-1;
148     ....rectangle->h = SQUARE_WIDTH-1;
149
150     ....VERIF_SDL_COMMAND(SDL_RenderFillRect(var->renderer, rectangle), "RenderFillRect");
151
152     ....SDL_RenderPresent(var->renderer);
153
154     ....free(rectangle);
155     ....}
156
157     ....else{
158     ....        ....ExitWithError("Out of range of the matrix");
159     ....    }
160
161
162 }
163
```

Grid.c

```
164 // Fonction qui dessine un rectangle dans la case
165 // Variables globales utilisées : RENDER_WIDTH, RENDER_HEIGHT, SQUARE_WIDTH
166 void CASE_CLICK_COORDS(Var *var, const int x, const int y){
167     /* Fonction qui dessine un rectangle dans la case */
168     // Case de la texture sur lequel on est :
169     int CASE_X = GET_CASE_FROM_COORD_X(x);
170     int CASE_Y = GET_CASE_FROM_COORD_Y(y);
171
172     if((0 <= x <= TEXTURE_WIDTH) && (0 <= y <= TEXTURE_HEIGHT)){
173         CASE_CLICK_CASE(var, CASE_X, CASE_Y);
174     }
175     else{
176         ExitWithError("Out of range of the matrix");
177     }
178 }
179
180
181
182 void CASE_CLICK_DISPLAY(Var *var, const int x, const int y){
183     if (x <= GRID_DISP_WIDTH && y <= GRID_DISP_HEIGHT){
184         // On met en cible du dessin la texture
185         VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, var->texture), "SetRenderTarget");
186
187         CASE_CLICK_COORDS(var, x+(var->camera->x), y+(var->camera->y));
188
189         // On remet la cible du rendu sur la fenêtre en entier
190         VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, NULL), "SetRenderTarget");
191     }
192 }
193
194 void CASE_CLICK_CASE_DISPLAY(Var *var, const int CASE_X, const int CASE_Y){
195     if (CASE_X <= CASE_NUMBER_WIDTH && CASE_Y <= CASE_NUMBER_HEIGHT){
196         // On met en cible du dessin la texture
197         VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, var->texture), "SetRenderTarget");
198
199         CASE_CLICK_CASE(var, CASE_X, CASE_Y);
200
201         // On remet la cible du rendu sur la fenêtre en entier
202         VERIF_SDL_COMMAND(SDL_SetRenderTarget(var->renderer, NULL), "SetRenderTarget");
203     }
204     else{
205         ExitWithError("CASE_CLICK_CASE_DISPLAY : Out of range");
206     }
207 }
208
209
210 void MOVE_CAMERA_MOUSE(SDL_Renderer *renderer, SDL_Texture *texture, SDL_Rect *camera, int xRel, int yRel){
211     int x = camera->x + xRel;
212     int y = camera->y + yRel;
213
214     if(0 <= x && x <= TEXTURE_WIDTH-RENDER_WIDTH && 0 <= y && y <= TEXTURE_HEIGHT-RENDER_HEIGHT){
215         camera->x = x;
216         camera->y = y;
217     }
218
219 }
```

Matrix.h

```
1  #ifndef MATRIX_H
2  #define MATRIX_H
3
4  typedef struct Matrix{
5      int width;
6      int height;
7      int **mat;
8  } matrix;
9
10 matrix CREATE_MATRIX(int width, int height);
11 void DESTROY_MATRIX(matrix mat);
12 void DESTROY_POINTER_MATRIX(matrix * mat);
13
14 #endif
```

Matrix.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16
17 matrix CREATE_MATRIX(int width, int height){
18     ... /* Create a matrix */
19     ... matrix matr;
20     ... matr.width = width;
21     ... matr.height = height;
22
23     ... matr.mat = malloc(width * sizeof(int*));
24
25     ... for (int i = 0; i < width; i++) {
26     ...     ... matr.mat[i] = CREATE_TAB_0(height);
27     ... }
28
29     ... return matr;
30 }
31
32 void DESTROY_MATRIX(matrix mat){
33     ... for (int i = 0; i < mat.width; i++)
34     ... {
35     ...     ... free(mat.mat[i]);
36     ... }
37     ... free(mat.mat);
38 }
39
40 void DESTROY_POINTER_MATRIX(matrix * mat){
41     ... DESTROY_MATRIX(*mat);
42     ... free(mat);
43 }
```

Var.h

```
1  ✓ #ifndef VAR_H
2  #define VAR_H
3
4  //----- Structure des variables importantes -----//
5  ✓ /* Liste :
6
7      ..... window
8      ..... renderer
9      ..... texture
10
11      ..... KEY_DOWN_STATUS
12      ..... XY_CASE_TAB
13
14      ..... camera
15      ..... gridDestRect
16  */
17  ✓ typedef struct var{
18      ..... SDL_Window *window;
19      ..... SDL_Renderer *renderer;
20      ..... SDL_Texture *texture;
21
22      ..... int *KEY_DOWN_STATUS;
23      ..... matrix *XY_CASE_MAT;
24      ..... matrix *coordLivingCells;
25
26      ..... SDL_Rect *camera;
27      ..... SDL_Rect *gridDestRect;
28
29
30  } Var;
31
32  void MALLOC_VAR(Var *var);
33
34  void DESTROY_VAR(Var *var);
35
36  #endif
```


Var.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5  #include <SDL.h>
6  #include "GlobalDef.h"
7  #include "Error.h"
8  #include "Array.h"
9  #include "Matrix.h"
10 #include "Var.h"
11 #include "Grid.h"
12 #include "Conversion_RLE.h"
13 #include "Evolution.h"
14 #include "eventGestion.h"
15
16 void MALLOC_VAR(Var *var){
17     ... var->KEY_DOWN_STATUS = malloc(sizeof(int*));
18     ... var->XY_CASE_MAT = malloc(sizeof(matrix));
19     ... var->coordLivingCells = malloc(sizeof(matrix));
20     ... var->camera = malloc(sizeof(SDL_Rect));
21     ... var->gridDestRect = malloc(sizeof(SDL_Rect));
22 }
23
24 void DESTROY_VAR(Var *var){
25     ... SDL_DestroyWindow(var->window);
26     ... SDL_DestroyRenderer(var->renderer);
27     ... SDL_DestroyTexture(var->texture);
28     ... free(var->KEY_DOWN_STATUS);
29     ... DESTROY_MATRIX(*var->XY_CASE_MAT);
30     ... DESTROY_MATRIX(*var->coordLivingCells);
31     ... free(var->camera);
32     ... free(var->gridDestRect);
33
34 }
```