

# Décomposition du schéma ClockTreeSchema

## 1. Structure racine

- **Type** : object
- **Propriété obligatoire** : tree
- **Aucune propriété supplémentaire** n'est autorisée à la racine (additionalProperties: false)

## 2. Propriété tree

- **Type** : object
- **Propriétés obligatoires** :
  - id (string, non vide)
  - schema\_version (string, format x.y.z)
  - elements (array d'objets)
  - transitions (array d'objets)
- **Aucune propriété supplémentaire** n'est autorisée dans tree

## 3. Propriété id

- **Type** : string
- **Non vide** (minLength: 1)

## 4. Propriété schema\_version

- **Type** : string
- **Format** : version x.y.z (ex: "1.0.0")

## 5. Propriété elements

- **Type** : array d'objets
- **Chaque élément** doit avoir :

- id (string, non vide, unique)
- name (string, non vide)
- position (objet {x, y})
- type (string, valeur parmi une liste)
- label (objet {align, text})
- default (type variable)
- outputTargets (array de string)
- size (objet {width, height

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "ClockTreeSchema",
  "type": "object",
  "required": ["tree"],
  "properties": {
    "tree": {
      "type": "object",
      "required": ["id", "schema_version", "elements", "transitions"],
      "properties": {
        "id": {
          "type": "string",
          "minLength": 1
        },
        "schema_version": {
          "type": "string",
          "pattern": "^\\d+\\.\\d+\\.\\d+$"
        },
        "elements": {
          "type": "array",
          "items": {
            "type": "object",
            "required": ["id", "name", "position", "type", "label", "default", "outputTargets", "size"],
            "properties": {
              "id": {
                "type": "string",
                "minLength": 1
              },

```

#### Détail des propriétés d'un élément

Propriété	Type / Contraintes
id	string, non vide

Propriété	Type / Contraintes
name	string, non vide
position	object `{x: number, y: number}`
type	string, enum (plusieurs valeurs possibles, ex: "fixedSource", "multiplexor", etc.)
label	object `{align: string, text: string}`
default	type variable (voir plus bas)
role	string ou null
block	string ou null
outputTargets	array de string
unit	object `{text: string, factor: string}` (optionnel, mais requis selon le type)
description	string ou null
size	object `{width: number, height: number}`
isTrustZone	boolean (optionnel)
oneOf	array (optionnel, voir plus bas)
min, max	number (optionnel)
outOfRange	boolean (optionnel)
fracDivisor	object `{base: number, power: number}` (optionnel)
multiplicatorFactor	array (optionnel)
possible_Input	array d'objets (optionnel, voir plus bas)

### Contraintes conditionnelles (allOf)

- **Si type est "fixedSource", "editableValue", "multiplier", "variableSource" :**
  - unit et outputTargets sont obligatoires
  - unit.text doit être "MHz" ou "KHz"
  - outputTargets doit avoir au moins 1 élément
- **Si type est "discreteValuesSource", "distinctFrequenceOscillator", "divider" :**
  - oneOf est obligatoire
- **Si type est "multiplexor" :**
  - possible\_Input est obligatoire
  - unit est interdit
- **Si type est "rectangularShape" :**
  - size est obligatoire
- **Si type est "fractionalValue" :**
  - min, max, outOfRange, fracDivisor, multiplicatorFactor sont obligatoires
- **Si oneOf existe :**
  - default doit être un nombre
- **Sinon :**
  - default peut être un nombre ou une string

#### Propriété oneOf

- **Type** : array
- **Chaque élément** : soit un nombre, soit un objet {const: number}

#### Propriété possible\_Input

- **Type** : array d'objets
- **Chaque objet** doit avoir :
  - available (boolean)
  - description (string)
  - input\_Id (string)
  - label (string)

- from (string)

## 6. Propriété transitions

- **Type** : array d'objets
- **Chaque transition** doit avoir :
  - id (string, format UUID)
  - sourceTaskId (string, non vide)
  - targetTaskId (string non vide OU array de strings non vides)
  - isVirtual (boolean)
  - transitionRoutingPoint (optionnel, array d'objets)

### Propriété transitionRoutingPoint

- **Type** : array d'objets
- **Chaque objet** doit avoir :
  - kind (string, "linear", "bezier" ou "arc")
  - x (number)
  - y (number)
  - pointIndex (integer >= 0)

### Exemple D'une Partie du Schema Code :

```

OPEN EDITORS
src > {} schemajson > {} properties > {} tree > {} properties > {} elements > {} items > {} properties > {} fracDivisor

{} sample.json data
TS index.ts src
{} VS.Clockjson C...
{} H7P.Clock Ijso...
X {} schemajson.s...
TS validator.ts src
SCHE... [ ] [ ] [ ] [ ] [ ]
  data
  {} sample.json
  errors
  sample_errors.txt
  > node_modules
  src
  TS index.ts
  {} schemajson
  TS validator.ts
  {} nodemon.json
  {} package-lock.json
  {} package.json
  tsconfig.json
  yarn.lock

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "ClockfreeSchema",
  "type": "object",
  "required": ["tree"],
  "properties": {
    "tree": {
      "type": "object",
      "required": ["id", "schema_version", "elements", "transitions"],
      "properties": {
        "id": {
          "type": "string",
          "minLength": 1
        },
        "schema_version": {
          "type": "string",
          "pattern": "^\\d+\\.\\d+\\.\\d+$"
        },
        "elements": {
          "type": "array",
          "items": {
            "type": "object",
            "required": ["id", "name", "position", "type", "label", "default", "outputTargets", "size"],
            "properties": {
              "id": {
                "type": "string",
                "minLength": 1
              },
              "name": {
                "type": "string",
                "minLength": 1
              },
              "position": {
                "type": "object",
                "required": ["x", "y"],
                "properties": {
                  "x": { "type": "number" },
                  "y": { "type": "number" }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

Exemple de Fixed source Conforme aux Regles du Schema :

```
{
  "tree": {
    "id": "clock_tree_001",
    "schema_version": "1.0.0",
    "elements": [
      {
        "id": "fixed_1",
        "name": "Fixed Source 1 ",
        "position": { "x": 10, "y": 20 },
        "type": "fixedSource",
        "label": { "align": "center", "text": "Fixed 1" },
        "default": 100,
        "role": "source",
        "block": "blockA",
        "outputTargets": ["editable_1", "multiplexor_1"],
        "unit": { "text": "MHz", "factor": "1" },
        "description": "Fixed frequency source",
        "size": { "width": 100, "height": 50 },
        "isTrustZone": false
      },
    ],
  },
}
```

```
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `npx ts-node ./src/index.ts`
🚀 JSON valide et conforme à toutes les règles.
[nodemon] clean exit - waiting for changes before restart
```