

Argumento	Descrição	Exemplo
--tec	Define a tecnologia alvo. Use qca.	--tec qca
--k-graphs	Número de grafos a serem gerados.	--k-graphs 50
--output-dir	Pasta onde os resultados serão salvos.	--output-dir resultados
--visualize	Gera arquivos visuais do grid (.gride .phys.png).	--visualize
--verbose	Exibe logs detalhados de debug. Útil se algo der errado.	-vou --verbose

Argumento	Descrição	Exemplo
--arch-size R C	Dimensões do grid (Linhas Colunas).	4 4
--qca-arch	Tipo de zona de clock: U(USE), R(RES), T(2DDWave).	U
--num-inputs	Número de nós de entrada (inputs primários).	3
--num-derivations	Controla a profundidade/complexidade do grafo.	5
--obstacle-intensity	Define a porcentagem do grid coberta por obstáculos aleatórios (0.0 a 1.0). (Server para dar variedade ao metodo balanceado).	0.15

Argumento	Descrição	Exemplo
--force-grid-size R C	Força um tamanho fixo de grid. Impede a expansão automática.	--force-grid-size 40 40
--grammar-reconvergence	Define se o modo USE pode criar reconvergências (True) ou apenas árvores (False)	--grammar-reconvergence False
--balanced	Gera grafos balanceados (So para USE).	
--unbalanced	Controla a profundidade/complexidade do grafo.	Gera grafos desbalanceados.

Exemplos de comandos:

### Para gerar grafos balanceados:

USE:

```
python scripts/runner.py single --tec qca --balanced --k-graphs 10 --num-inputs 3 --obstacle-intensity 0.15 --visualize --output-dir datasets/results_balanceados
```

### Para gerar grafos desbalanceados:

USE:

(Com Reconvergencia):

```
python scripts/runner.py single --tec qca --qca-arch U --unbalanced --k-graphs 10 --arch-size 20 20 --num-inputs 3 --num-derivations 2 --grammar-reconvergence True --visualize --output-dir datasets/results_desbalanceados
```

(Sem Reconvergencia):

-Mais facil de Gerar.

```
python scripts/runner.py single --tec qca --qca-arch U --unbalanced --k-graphs 10 --arch-size 20 20 --num-inputs 3 --num-derivations 5 --grammar-reconvergence False --visualize --output-dir datasets/results_desbalanceados
```

2DDWAVE:

```
python scripts/runner.py single --tec qca --qca-arch T --unbalanced --k-graphs 10 --num-inputs 4 --num-derivations 3 --visualize --output-dir datasets/results_desbalanceados
```

### Sobre o arquivo grid gerado (legenda):

Um arquivo de texto simples para visualização rapida no terminal/editor.

#### Legenda:

- I: **Input** (Entrada do circuito).
- OUT: **Output** (Saida final).
- [O]: **Operação** (Majority/AND/OR).
- X: **Crossover** (Cruzamento de fios - Coplanar).
- +: **Routing** (Fio normal).
- b: **Buffer** (Apenas no modo Balanced).
- <C>: **Convergência** (Ponto onde caminhos se unem).

### Instalação do ambiente virtual:

Rode esses comandos no terminal:

Windows:

```
python -m venv venv
```

Linux:

```
python3 -m venv venv
```

**Sempre que for utilizar o gerador, precisa ativar o venv, utilize o comando:**

Windows:

```
.\venv\Scripts\Activate
```

Linux:

```
source venv/bin/activate
```

**Com o Ambiente ativado, você tem de instalar as dependencias (apenas na primeira vez):**

```
pip install -r requirements.txt
```