

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 (.grid e .phys.png).	--visualize
--verbose	Exibe logs detalhados de debug. Útil se algo der errado.	-v ou --verbose

Argumento	Função	
--balanced	Gera grafos balanceados (So para USE).	
--unbalanced	Gera grafos desbalanceados.	

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.	10

Argumento	Descrição	Exemplo
--obstacle-intensity	Define a porcentagem do grid coberta por obstáculos aleatórios (0.0 a 1.0). (Server para dar variedade ao método balanceado).	0.15

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:

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

2DDWAVE:

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

Sobre o arquivo grid gerado (legenda):

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

Legenda:

- ⑩ I: **Input** (Entrada do circuito).
- ⑩ OUT: **Output** (Saída final).
- ⑩ [0]: **Operação** (Majority/AND/OR).
- ⑩ X: **Crossover** (Cruzamento de fios - Coplanar).
- ⑩ +: **Routing** (Fio normal).
- ⑩ b: **Buffer** (Apenas no modo Balanced).
- ⑩ <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
```