



Engineering and Physical Sciences  
Research Council



# Language and Hardware Acceleration Backend for Graph Processing

Andrey Mokhov, Alessandro de Gennaro, Ghaith Tarawneh,  
Jonny Wray, Georgy Lukyanov, Sergey Mileiko, Joe Scott,  
Alex Yakovlev, Andrew Brown

*Verona, September 2017*

# POETS Project Team



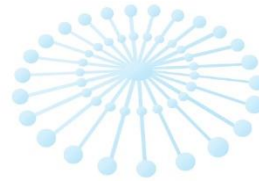
UNIVERSITY OF  
CAMBRIDGE

ARM



CCFE  
CULHAM CENTRE FOR  
FUSION ENERGY

Imperial College  
London



e-Therapeutics plc  
The Network Pharmacology Company




Newcastle  
University

Imagination

MAXELLER  
Technologies  
MAXIMUM PERFORMANCE COMPUTING

UNIVERSITY OF  
Southampton

nmi  
Semiconductors  
to Systems



nag®

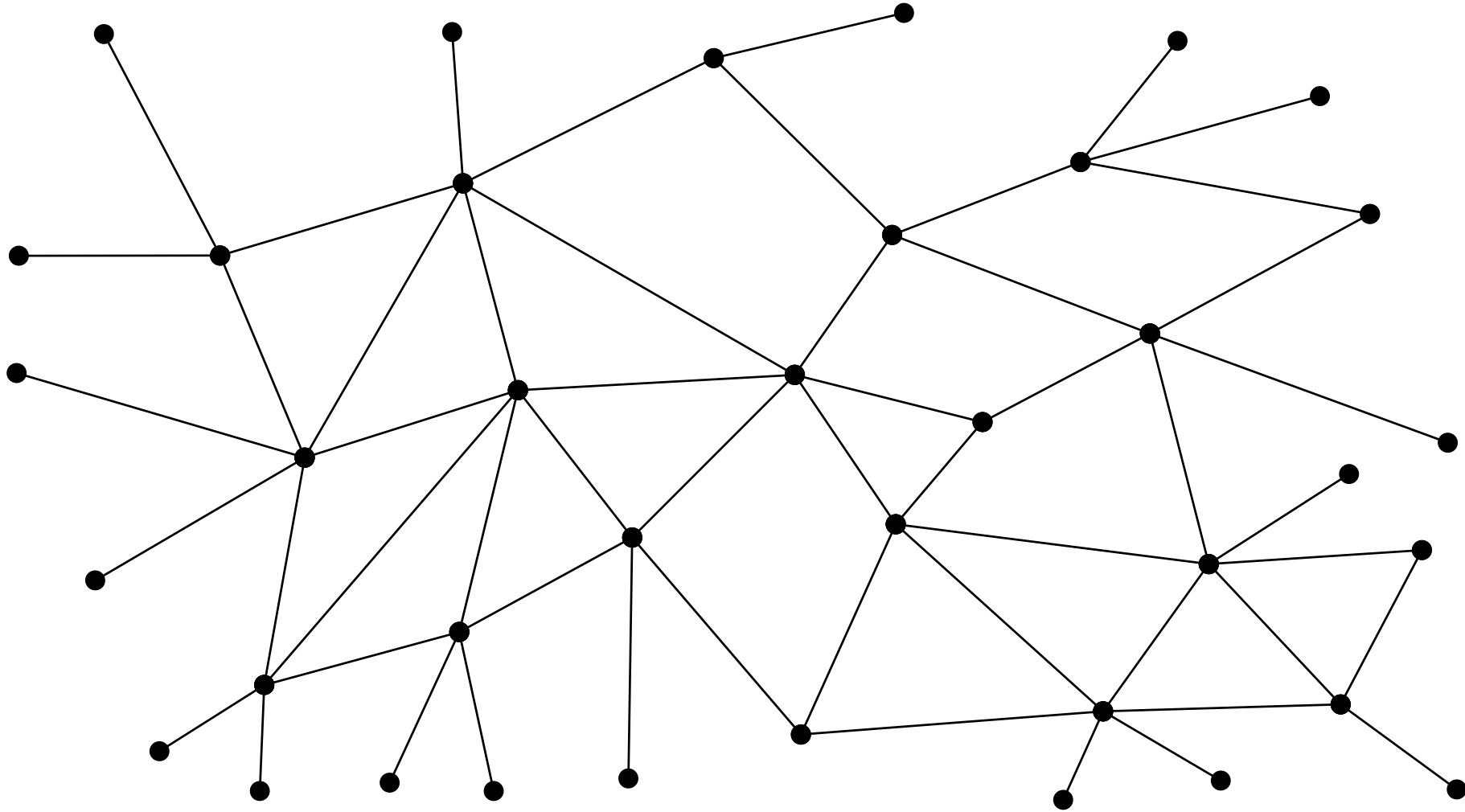
Industrial driver of the research: **e-Therapeutics**

- Drug discovery company (Oxford, UK)
- Using **computational** and mathematical approaches to discovery of new drugs as opposed to usual wet-lab, experimental approaches
- Key computational problem: **predicting the impact of drug** candidates

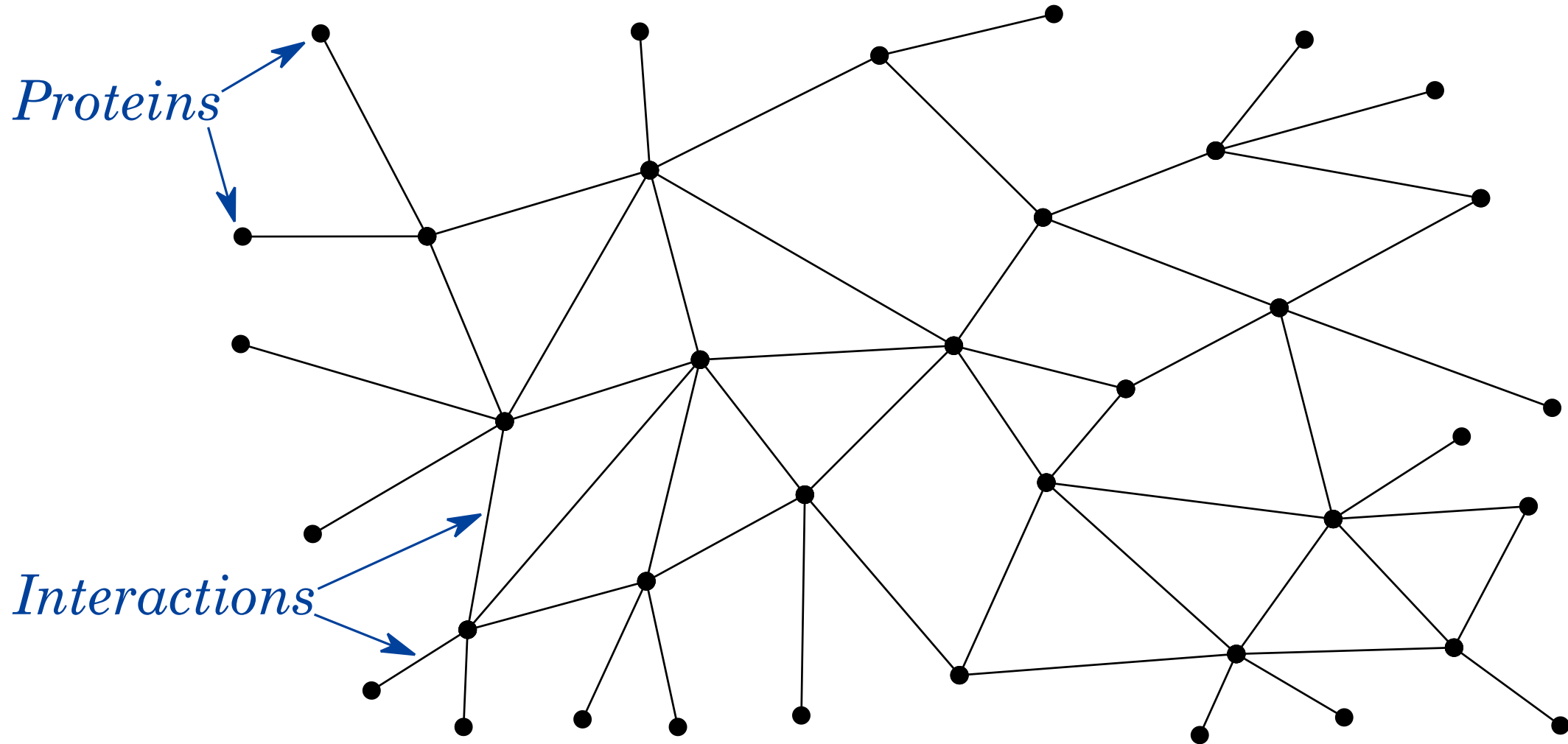
Presented results:

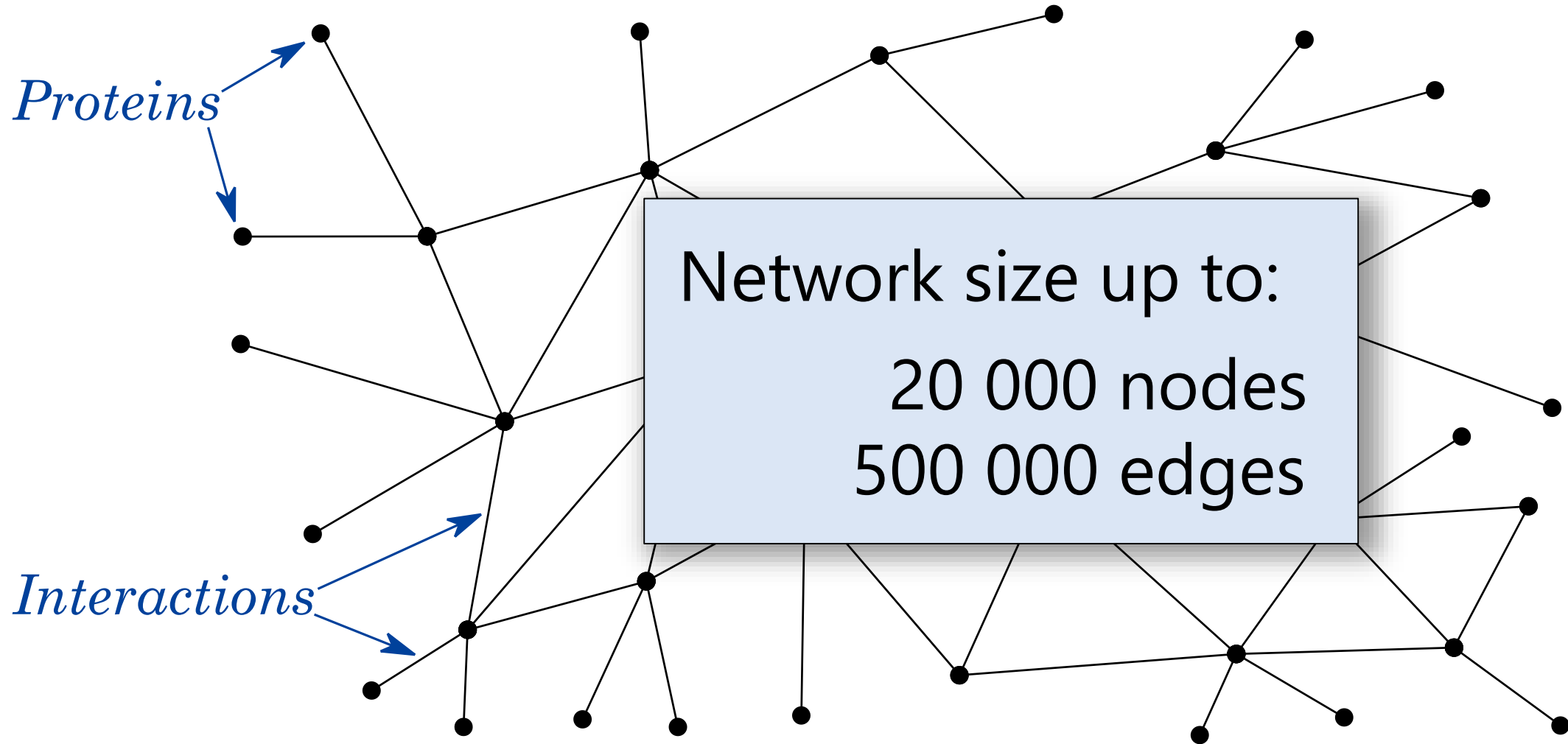
- **Language** for constructing and transforming protein-interaction networks
- **Hardware acceleration backend** for network analysis
- Achieved **1000x acceleration** compared to conventional analysis in software

# Protein-Interaction Networks

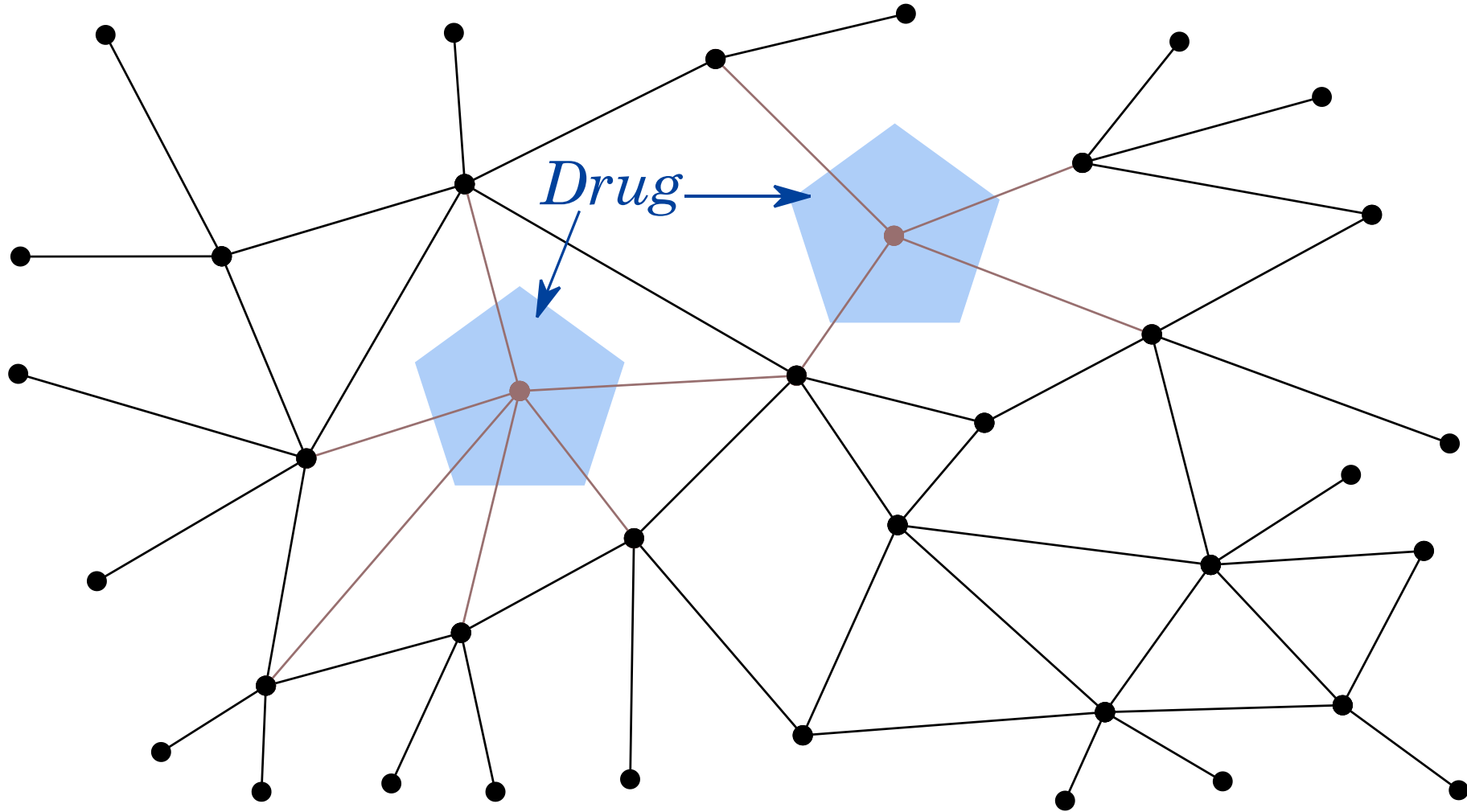


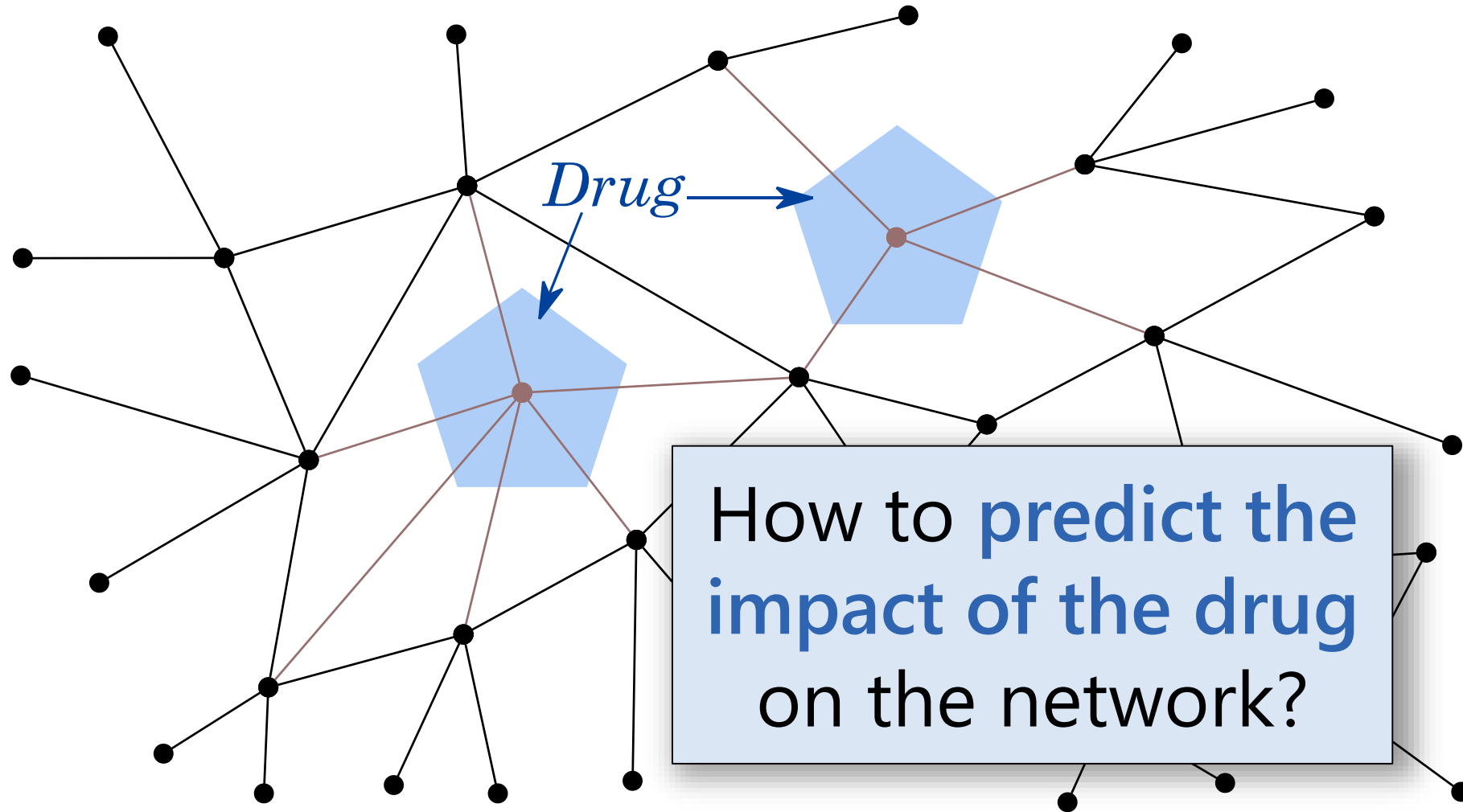
# Protein-Interaction Networks



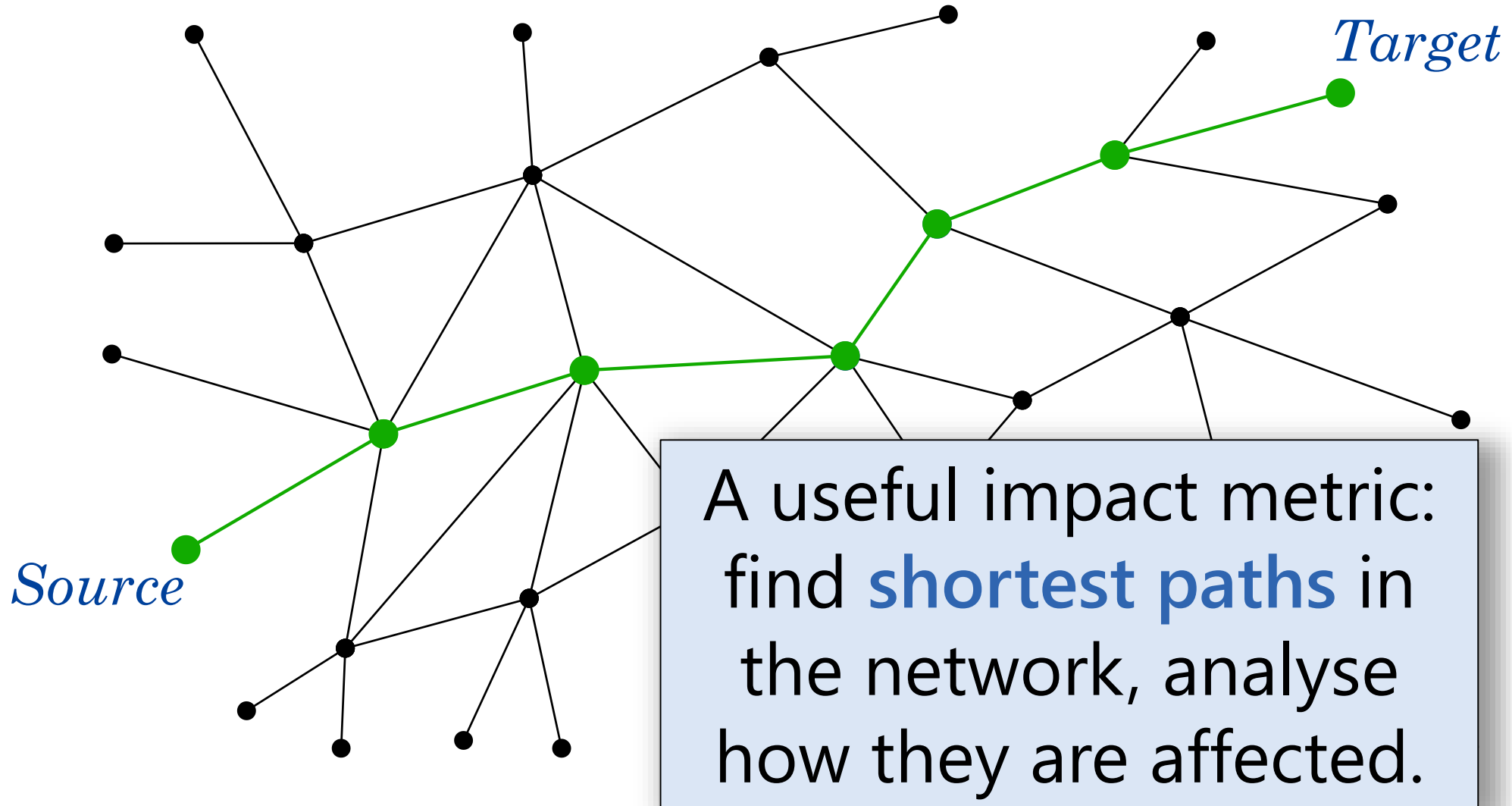


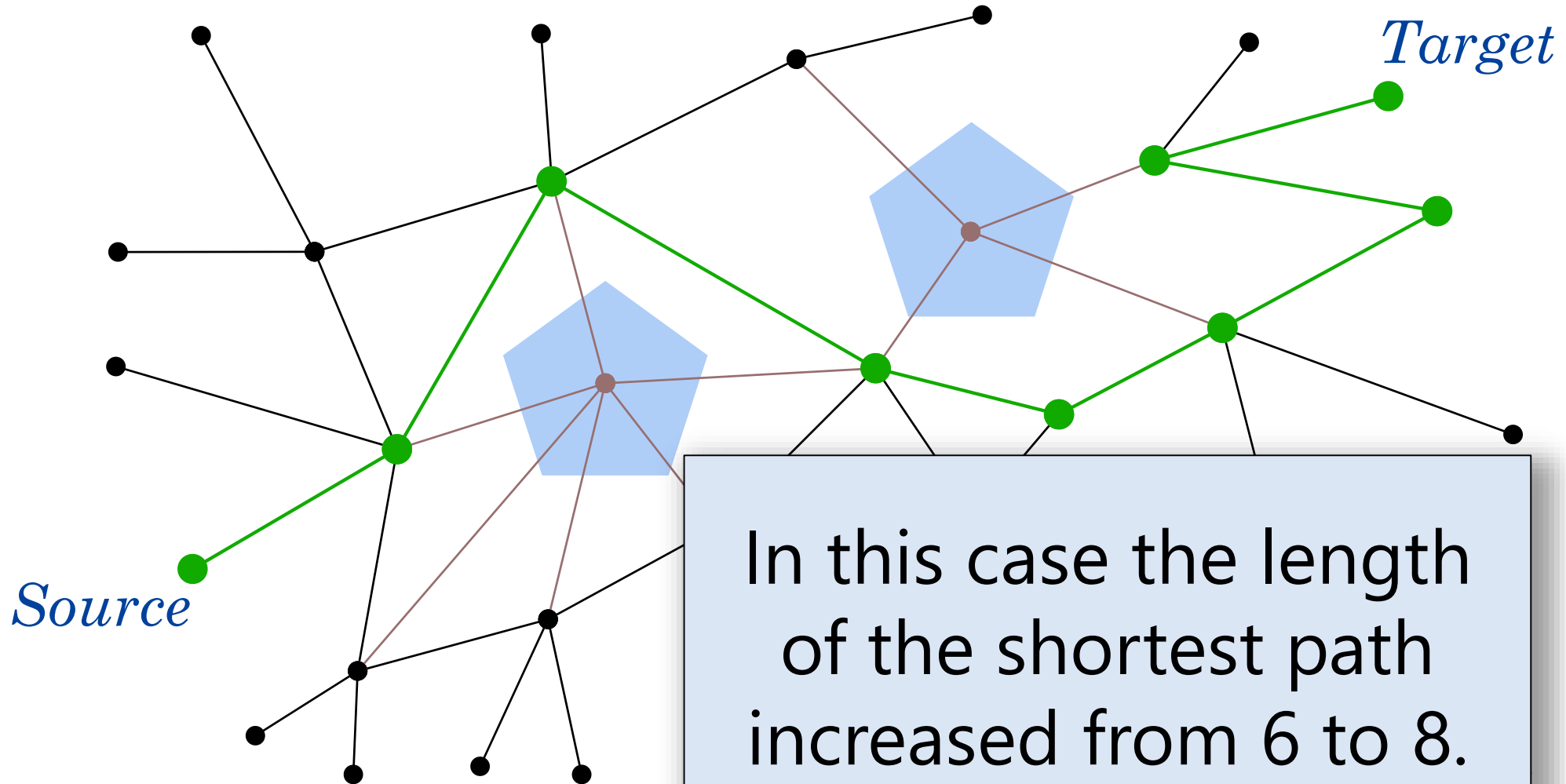
# Drug Impact Analysis

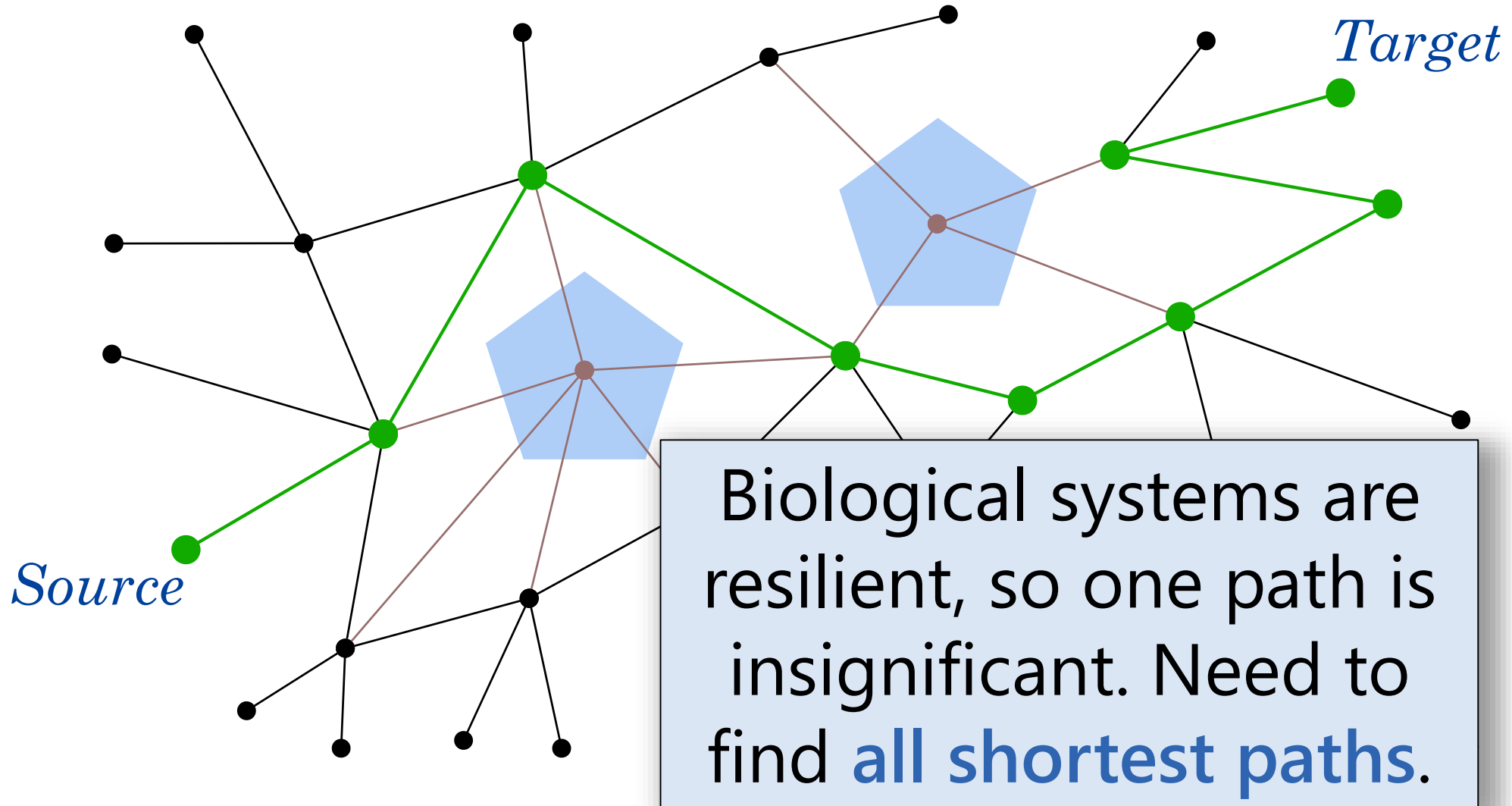












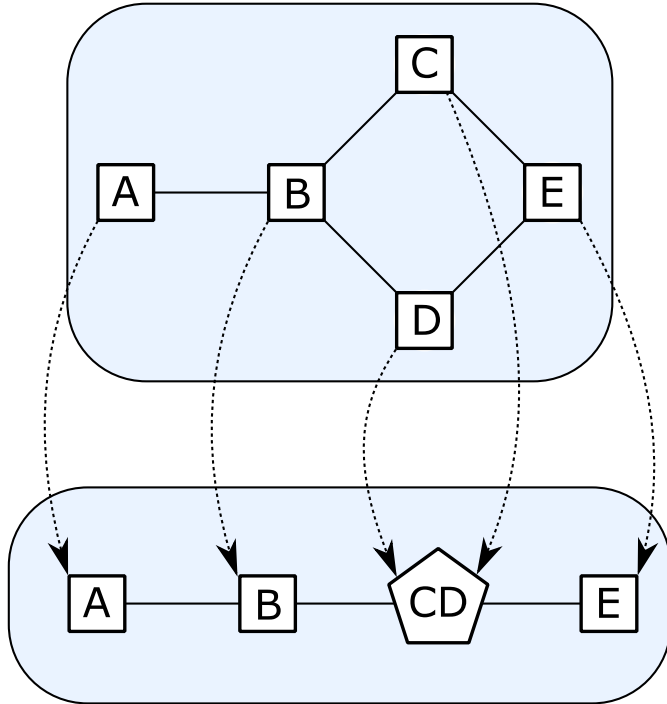
Impact analysis is a **computational bottleneck** for e-Therapeutics:

- Need to analyse multiple networks (potentially **thousands**)
- Need to analyse multiple drug candidates (potentially **millions**)
- A single analysis run currently takes time in the order of **seconds**

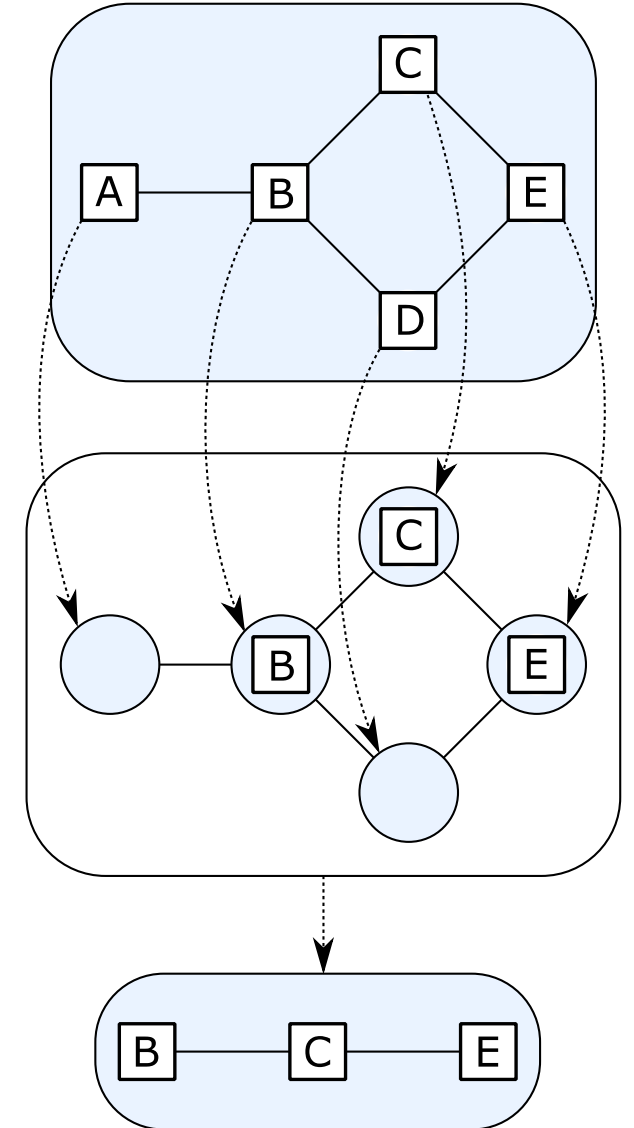
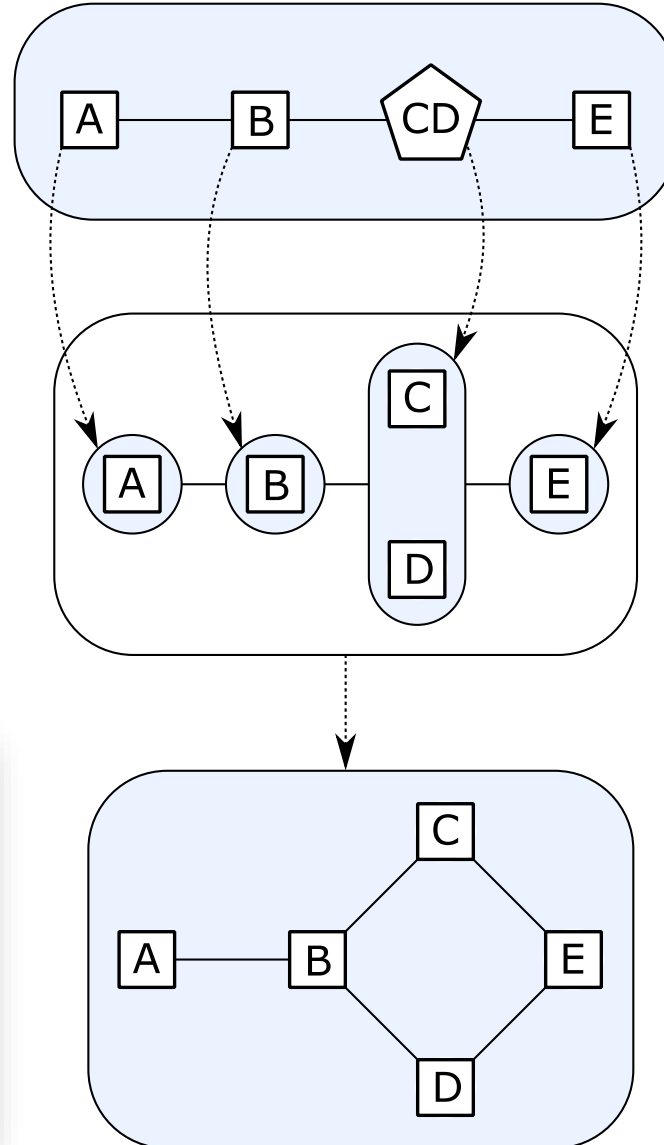
Conventional software implementation:

- Run **Breadth-First Search** (BFS) from all possible source nodes
- Computational complexity of a single BFS:  **$O(|V|+|E|)$**
- Large constant factors hidden in  **$O$**  due to irregular memory access

**Idea: Embed networks in FPGA and measure signal propagation time**

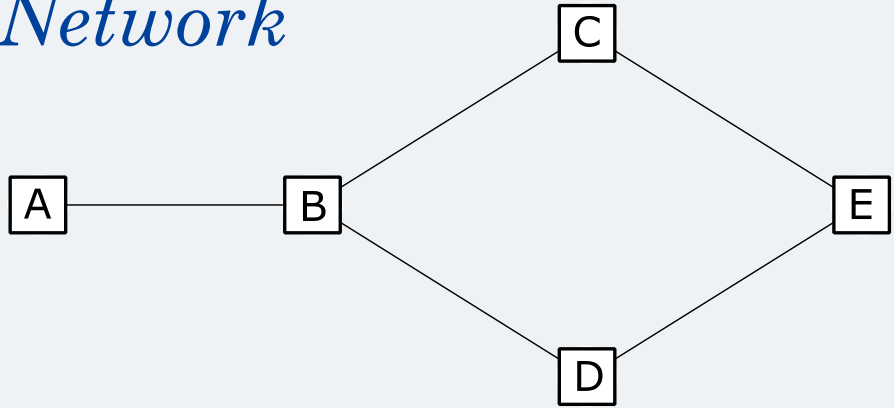


DSL embedded in  
**Haskell**: manipulate  
networks using familiar  
FP abstractions.



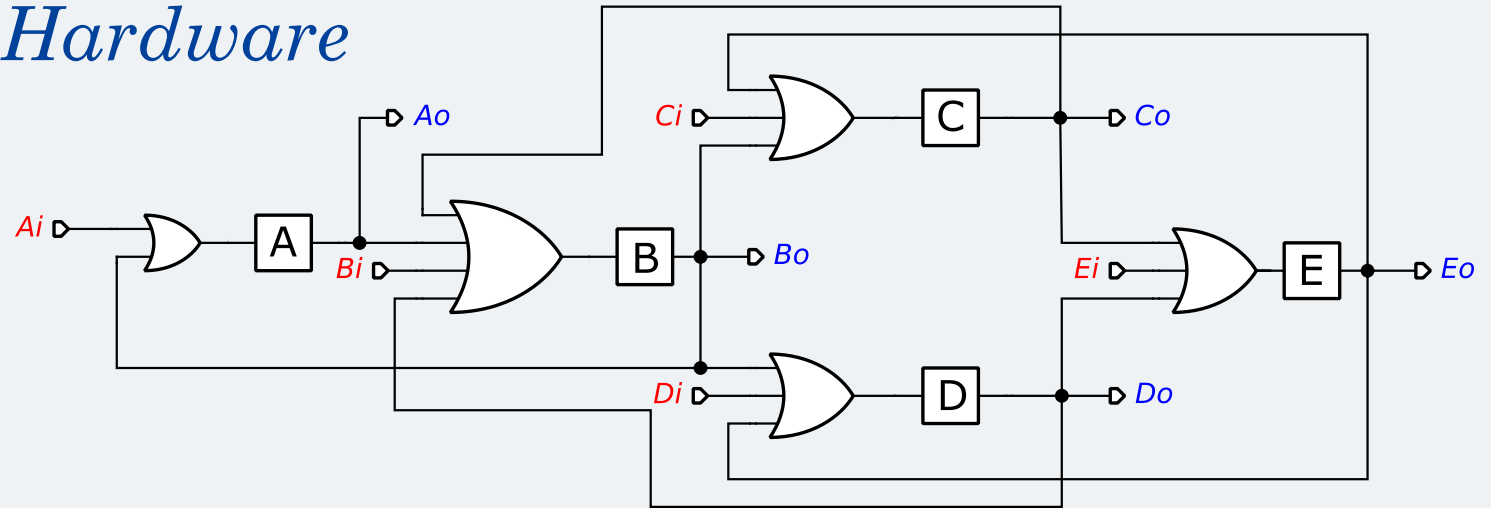
# Impact Analysis Acceleration

*Network*



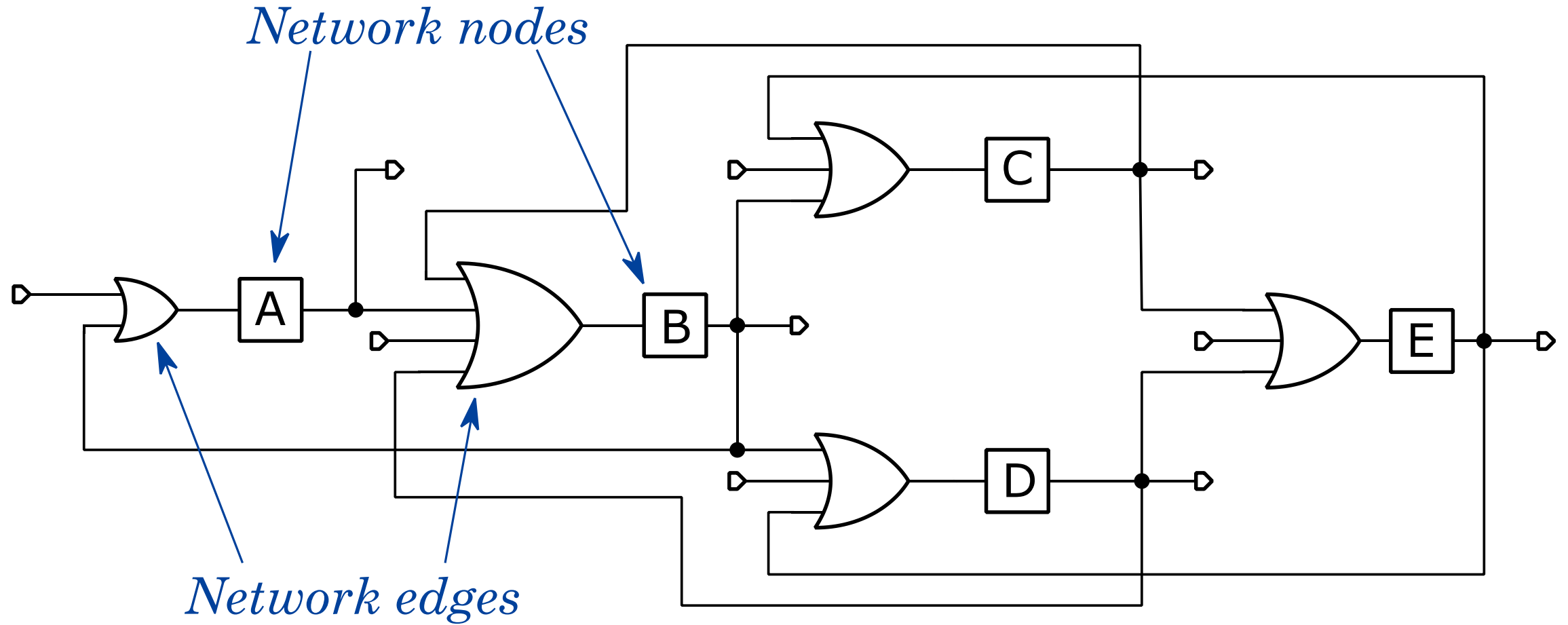
*Compiler from network  
DSL to hardware*

*Hardware*

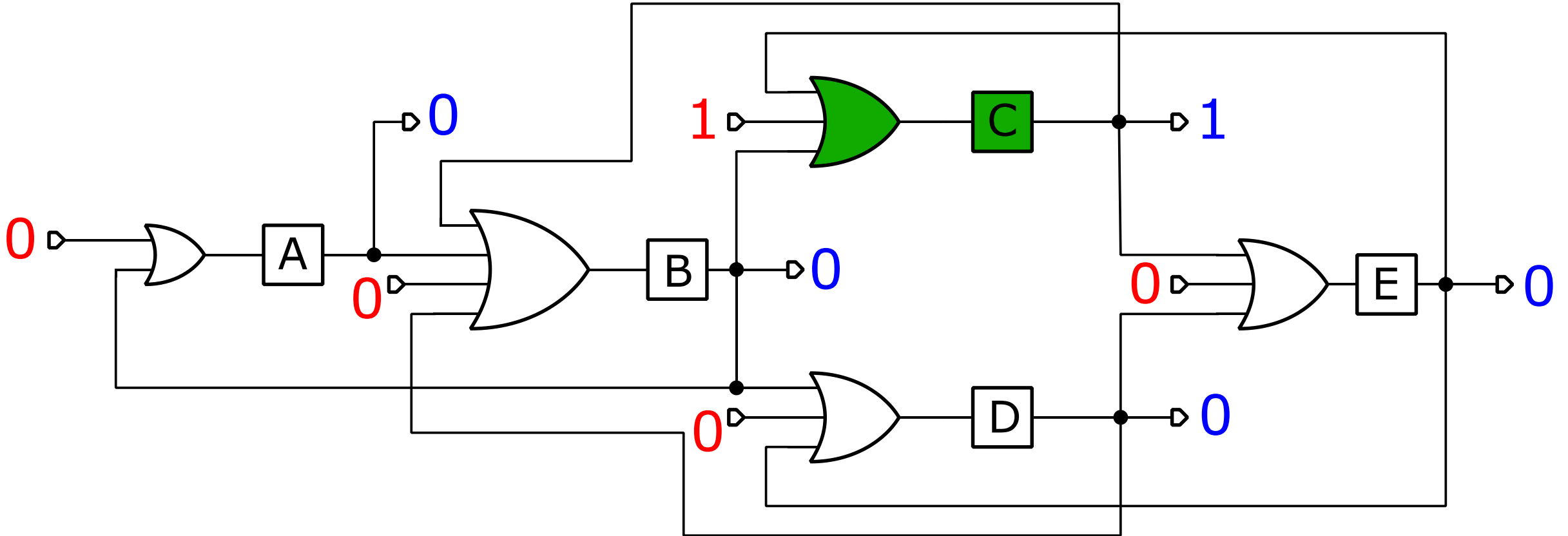


*Network  
analysis results*

# Embedding Networks in FPGAs

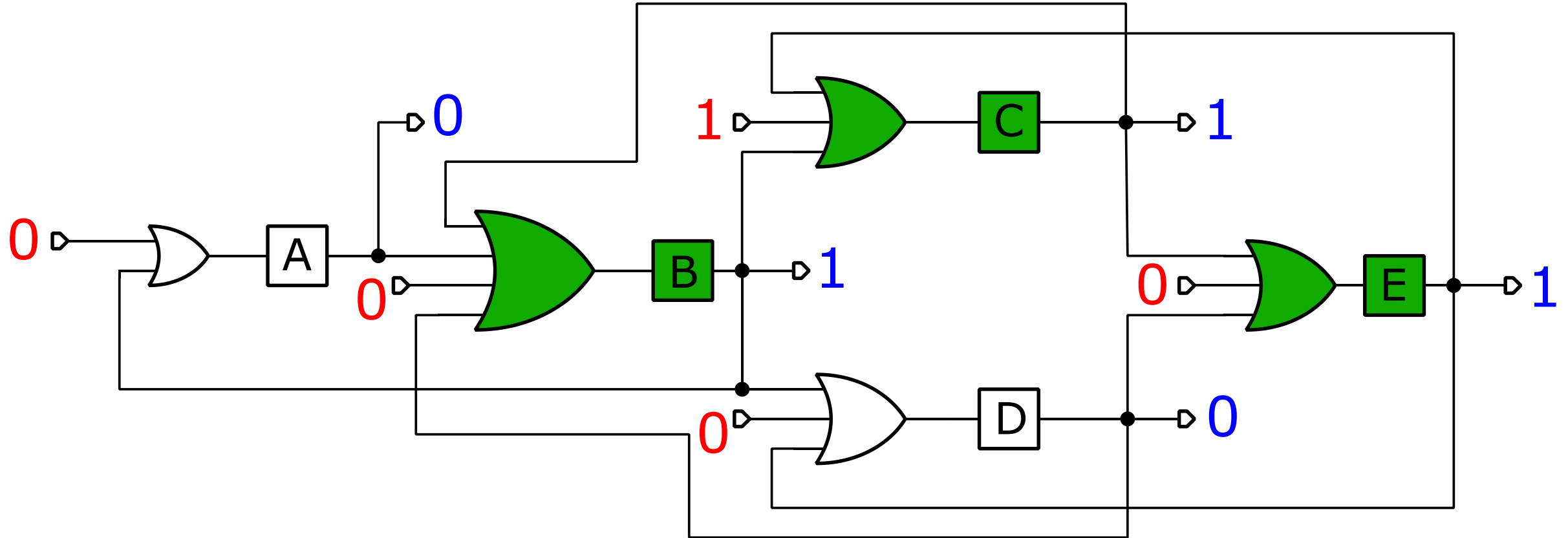


# Breadth-First Search in FPGA

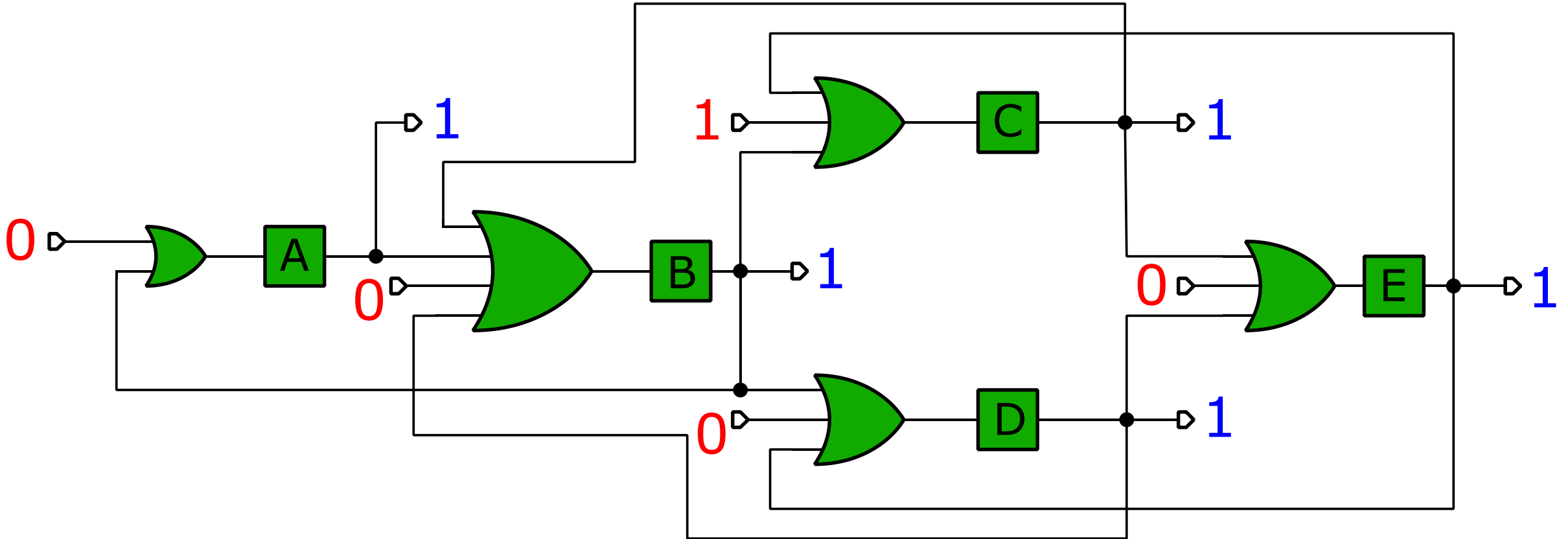


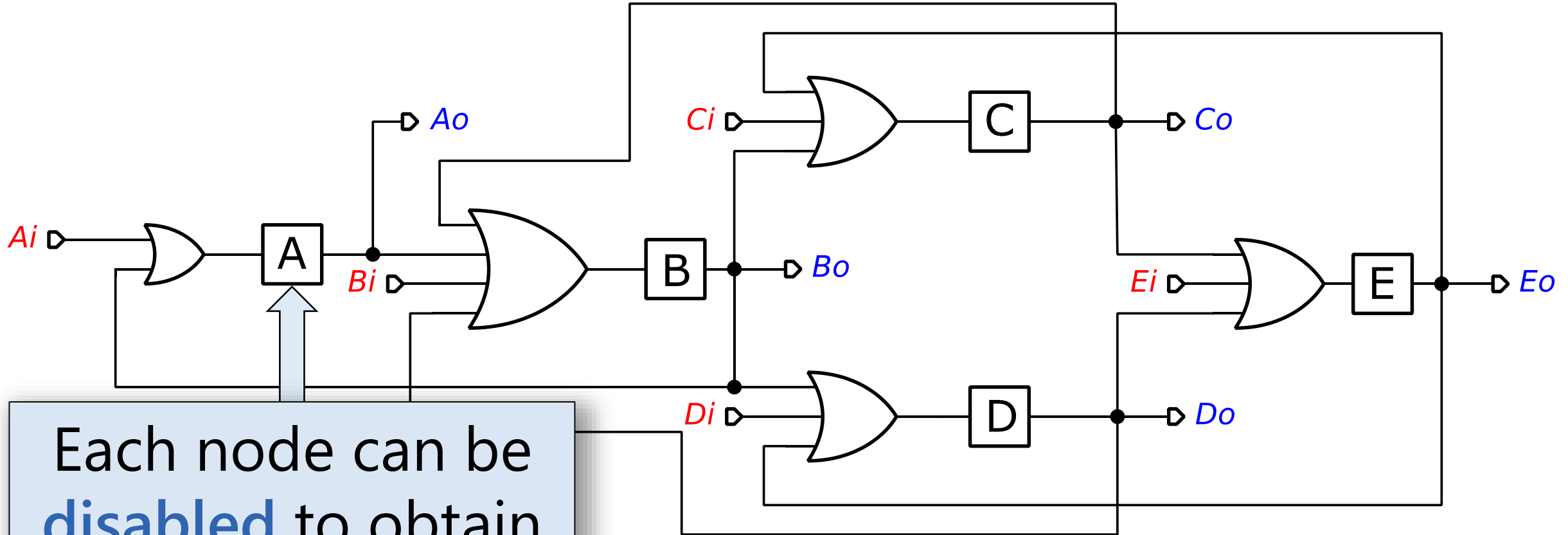


# Breadth-First Search in FPGA



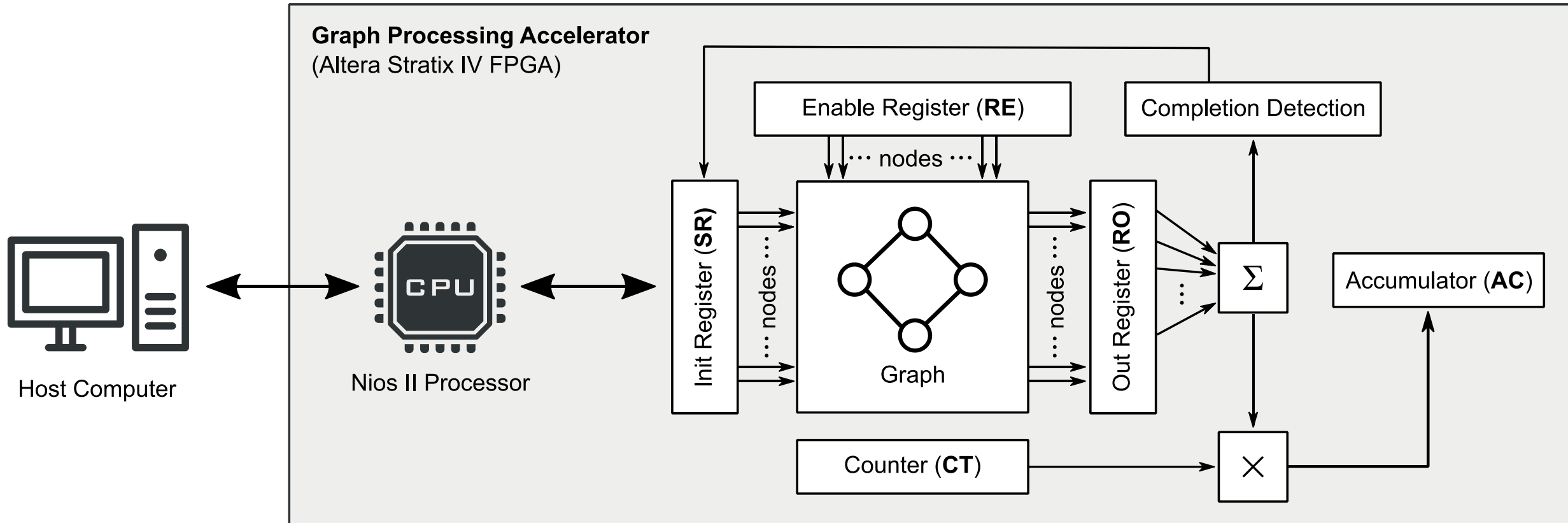
# Breadth-First Search in FPGA





Each node can be **disabled** to obtain a subnetwork.

# Embedding Networks in FPGAs



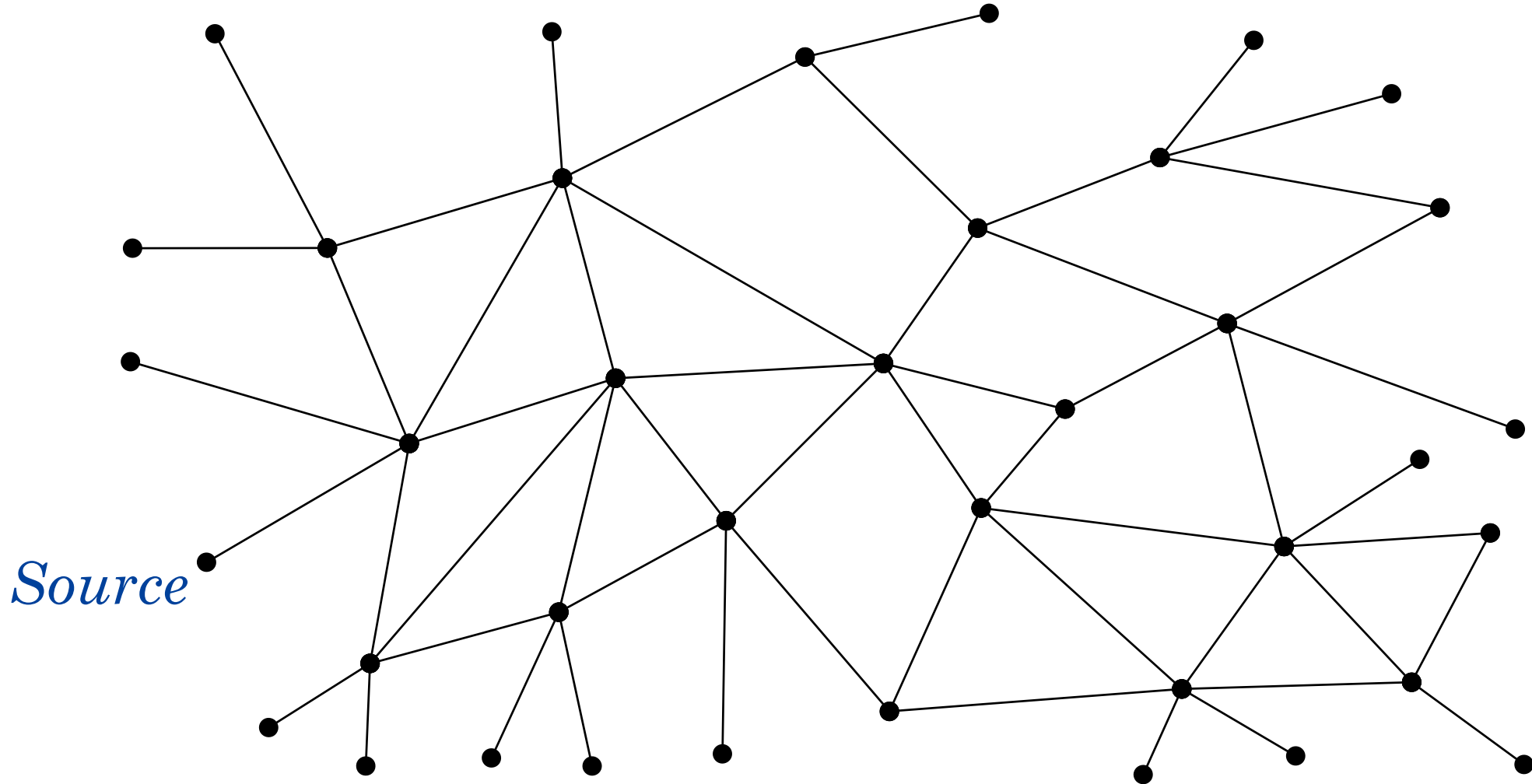
Key idea: we only need to analyse **induced subnetworks** of the original network. All such subnetworks are obtained by disabling some nodes. **No resynthesis!**

# Benchmarks

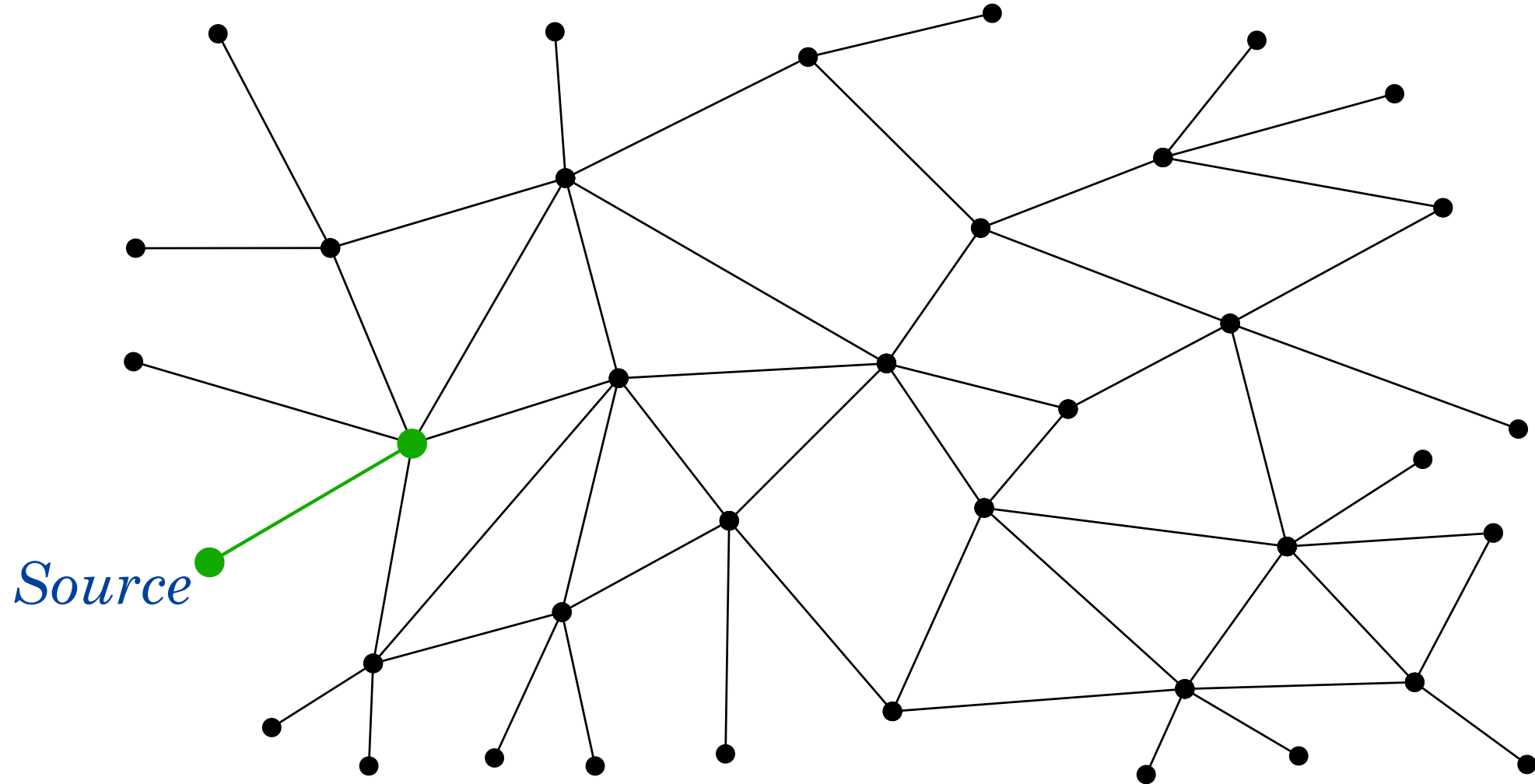
Network	n1	n2	n3	n4	n5
Number of vertices	15	87	349	1628	3487
Number of edges	42	804	6456	53406	115898
Logic utilisation (LUT + memory)	<1%	1%	2%	11%	25%
Peak interconnect usage	<1%	2%	16%	<b>75%</b>	<b>91%</b>
Maximum frequency (MHz)	283	246	159	131	107
Software throughput (nets/sec)	$> 10^4$	1176	56	3	0.88
FPGA throughput (nets/sec)	$> 10^6$	204290	33186	5489	2205
Acceleration factor	100x	173x	592x	<b>1829x</b>	<b>2505x</b>

**CPU:** Intel i7-6700HQ 2.60GHz, 6MB cache  
**FPGA:** Altera Stratix IV

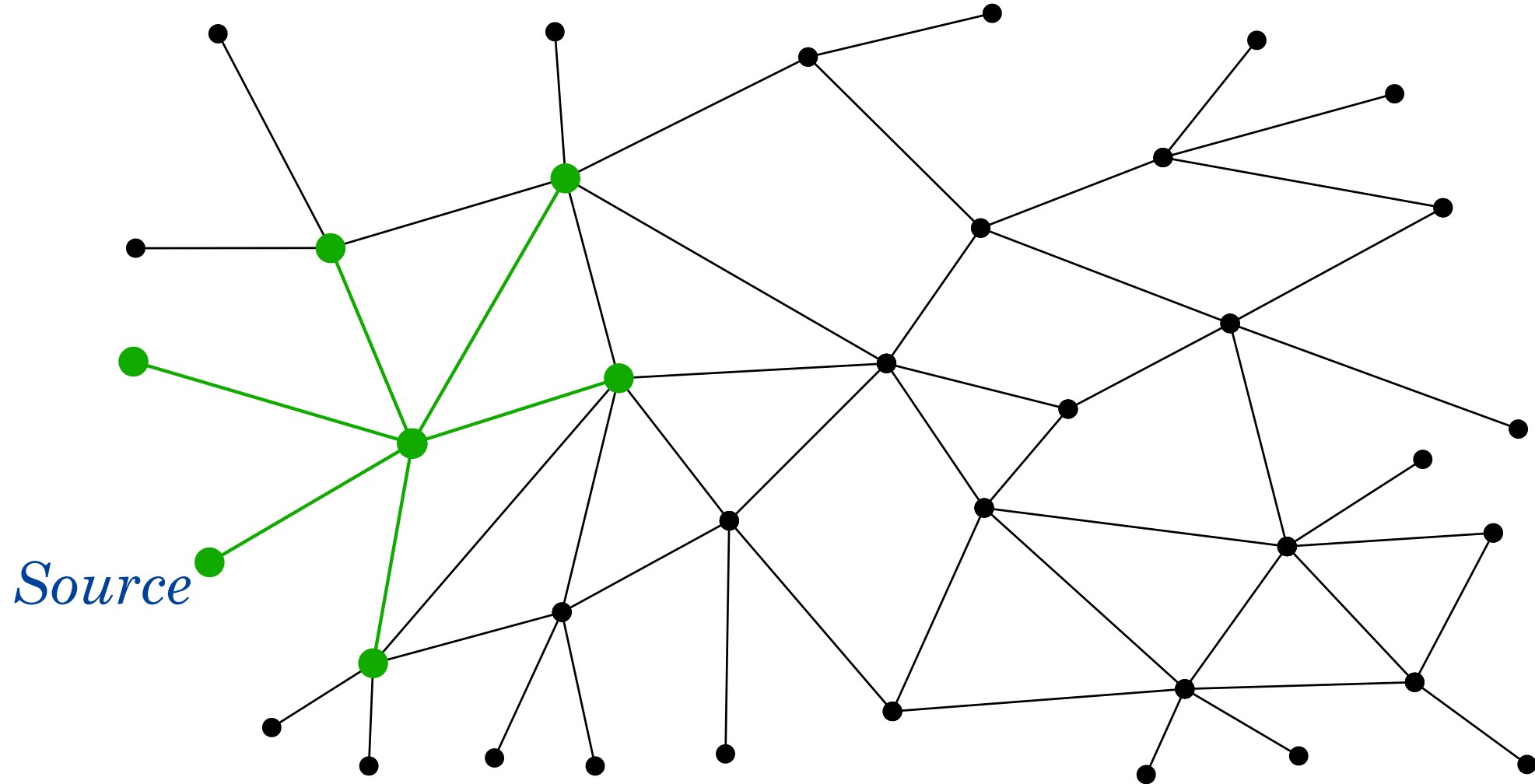
# Why Is It So Fast?



# After 1 clock cycle

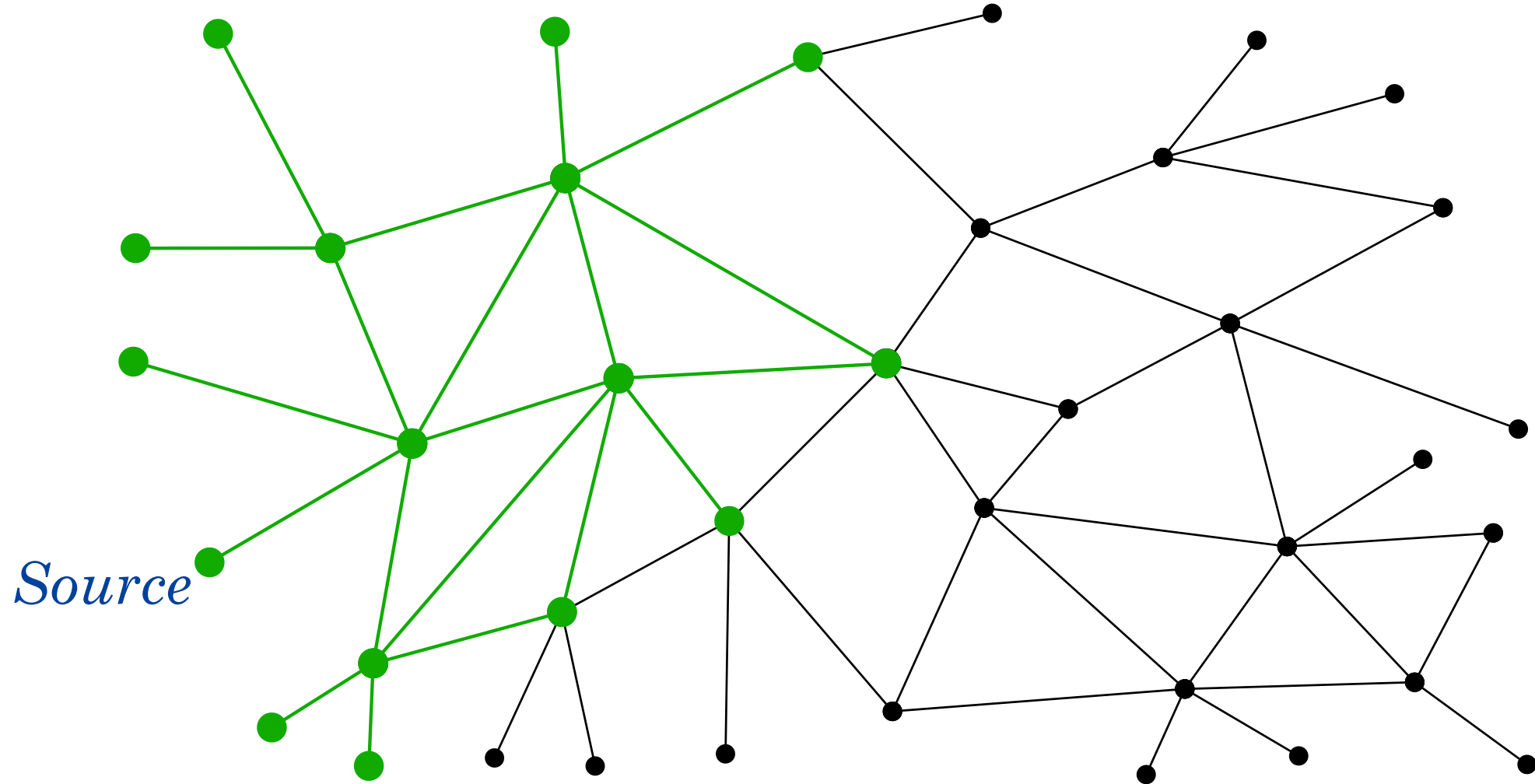


# After 2 clock cycles

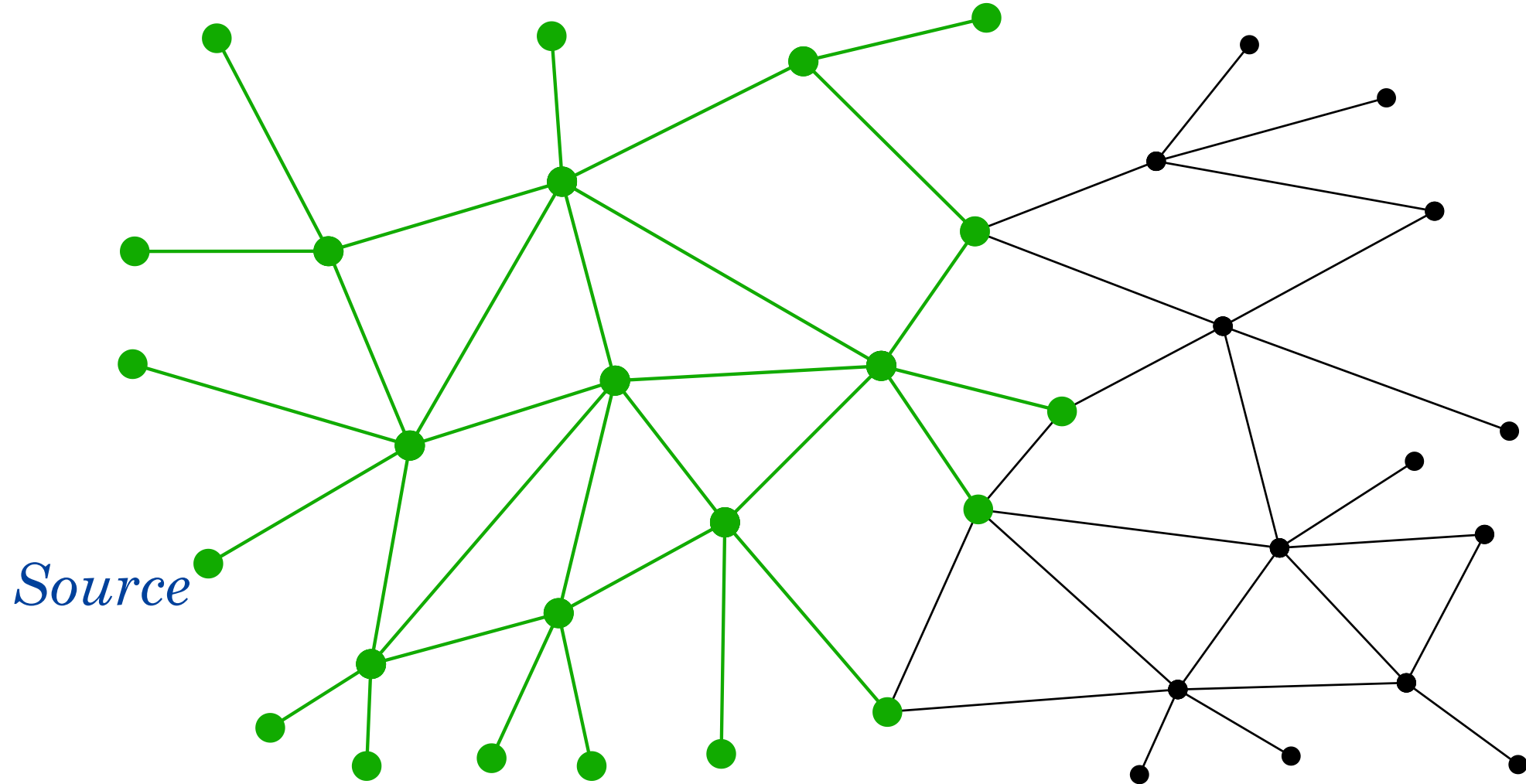




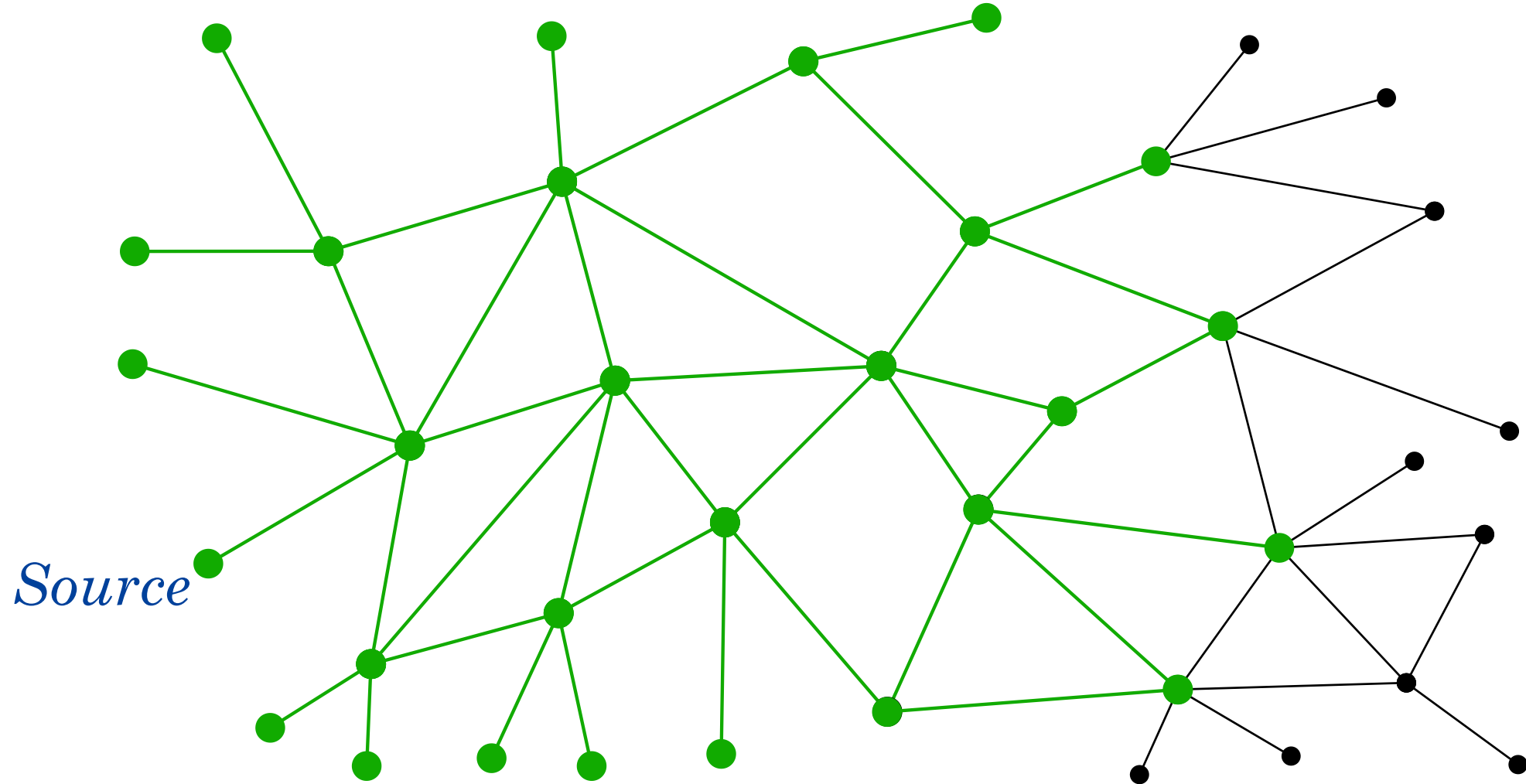
# After 3 clock cycles



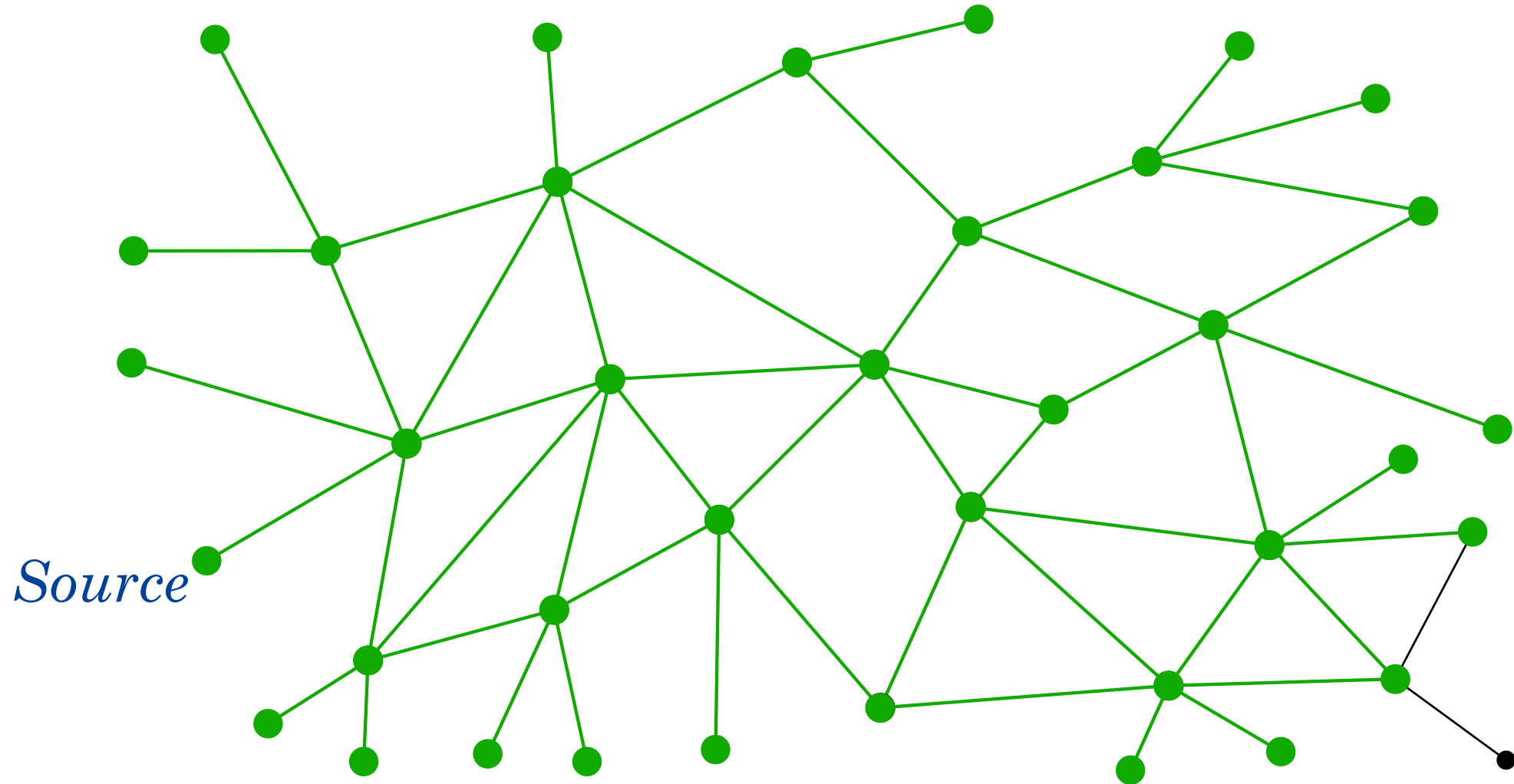
# After 4 clock cycles



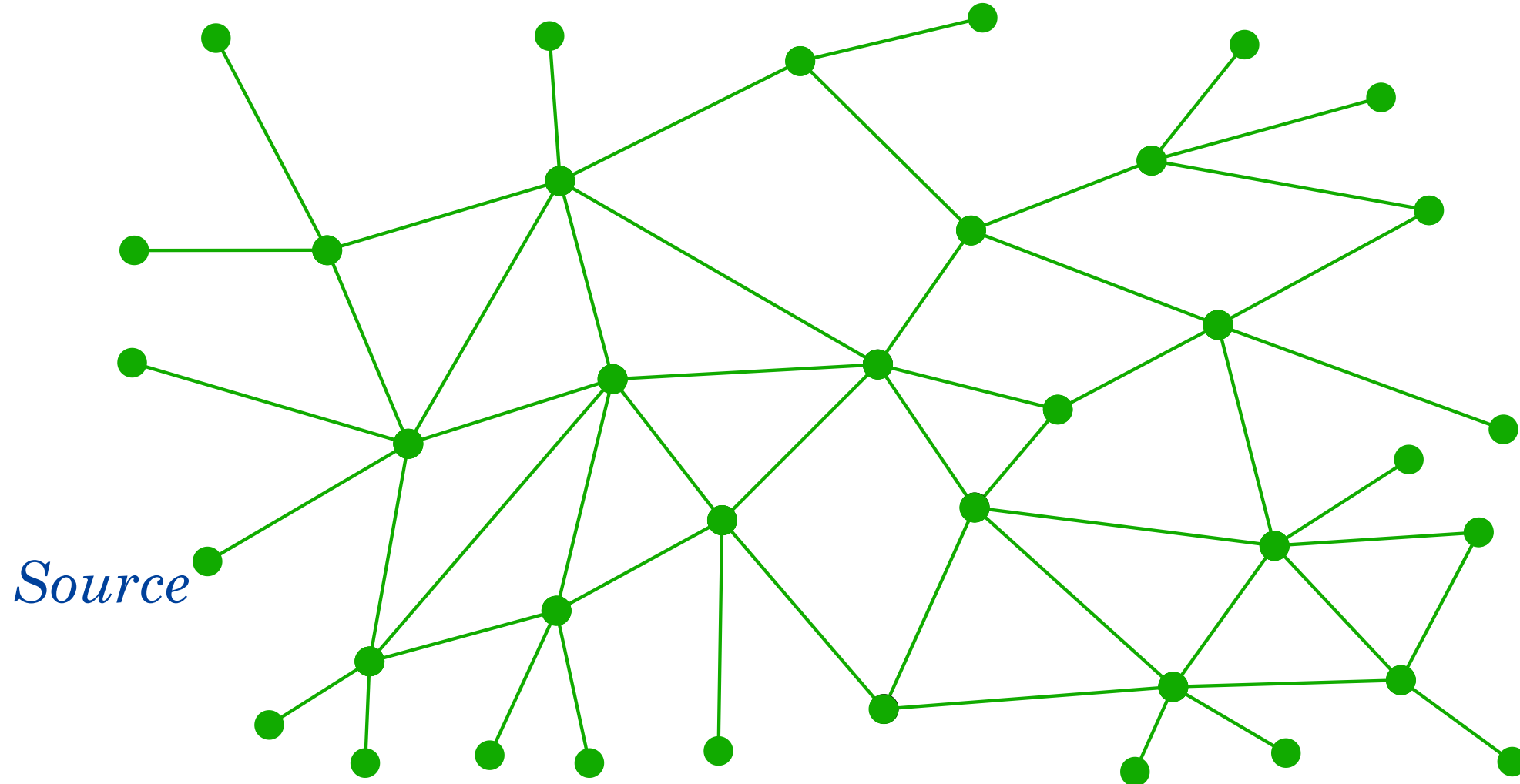
# After 5 clock cycles



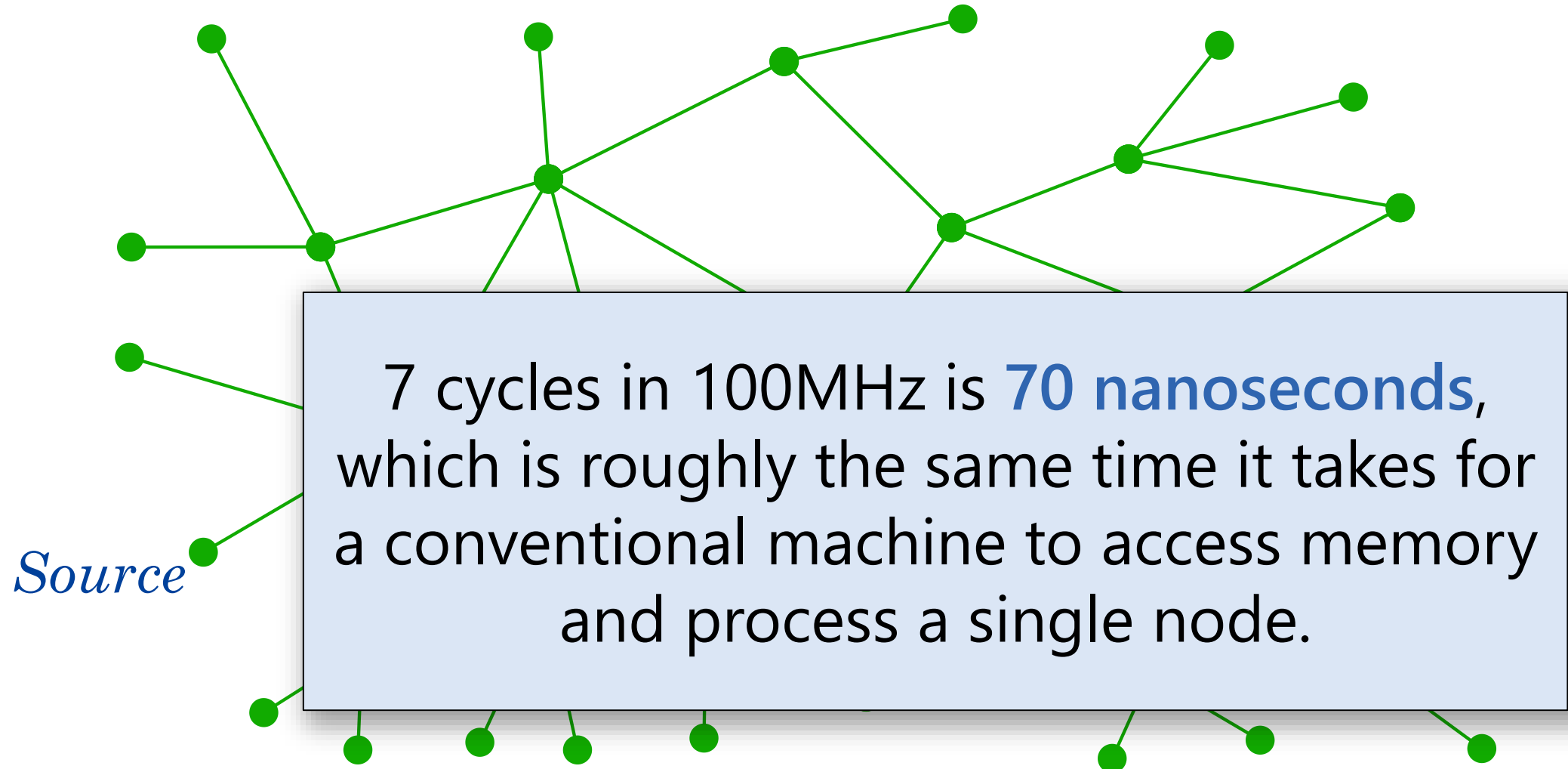
# After 6 clock cycles



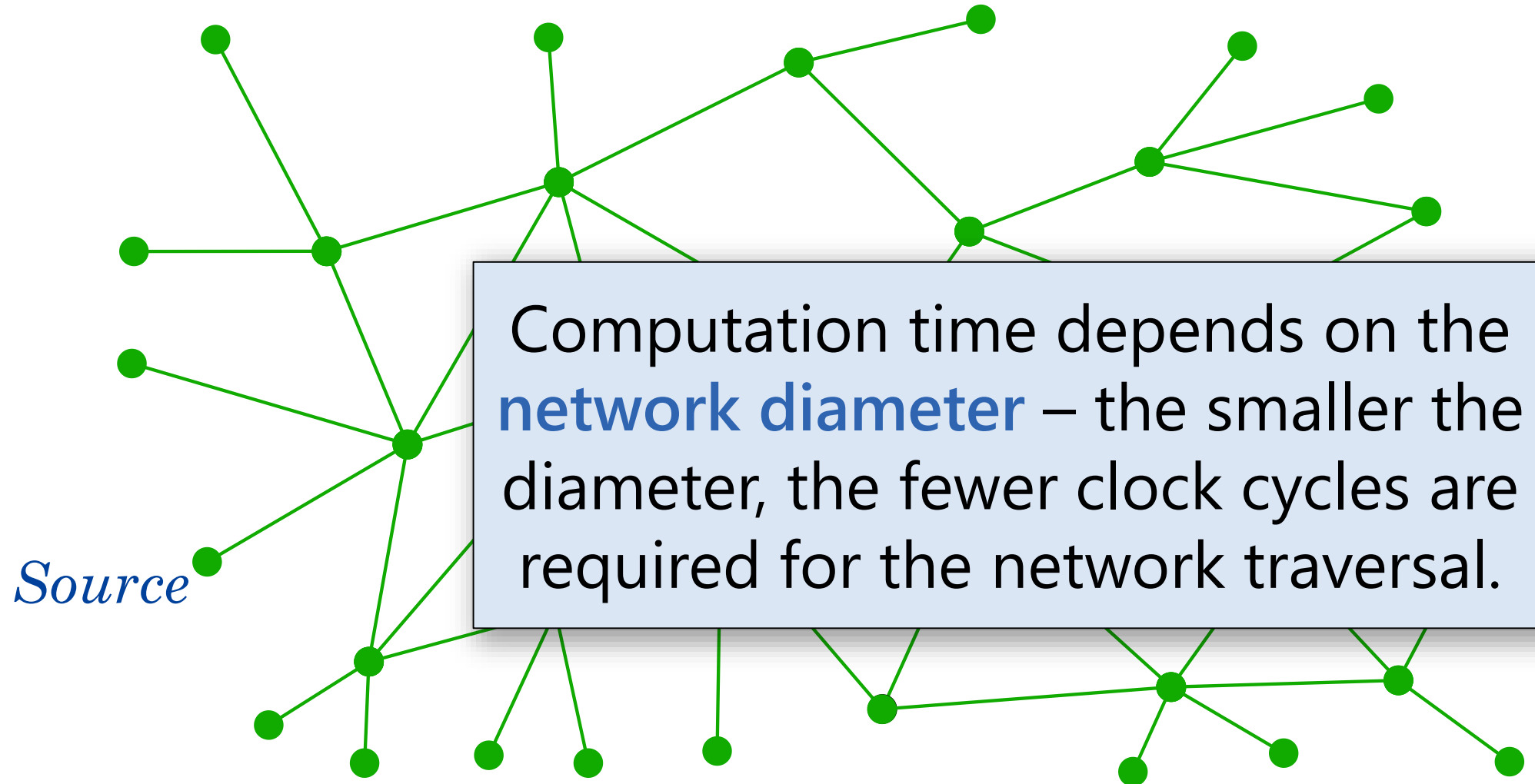
# After 7 clock cycles... Done!



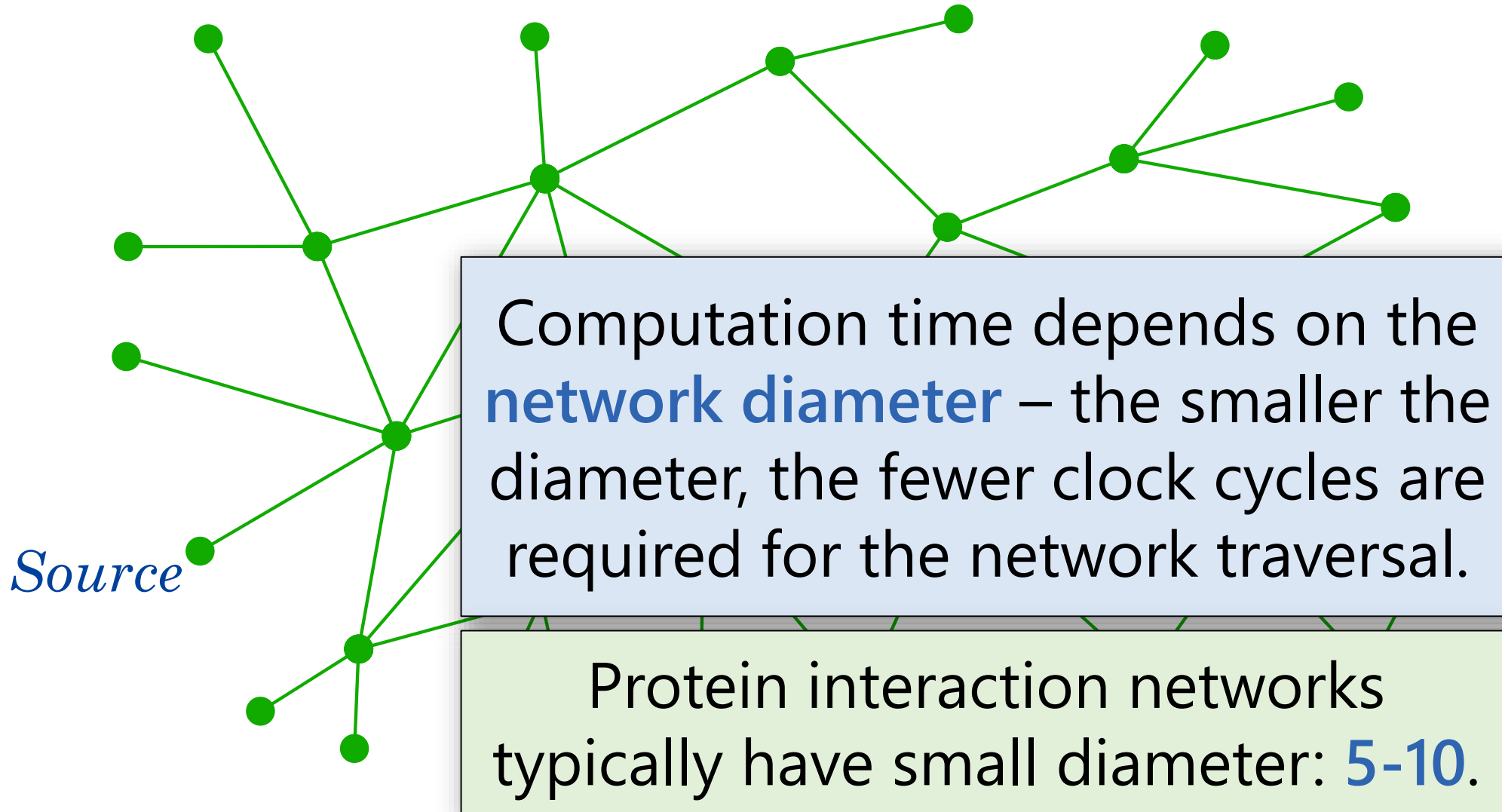
# After 7 clock cycles... Done!



# After 7 clock cycles... Done!

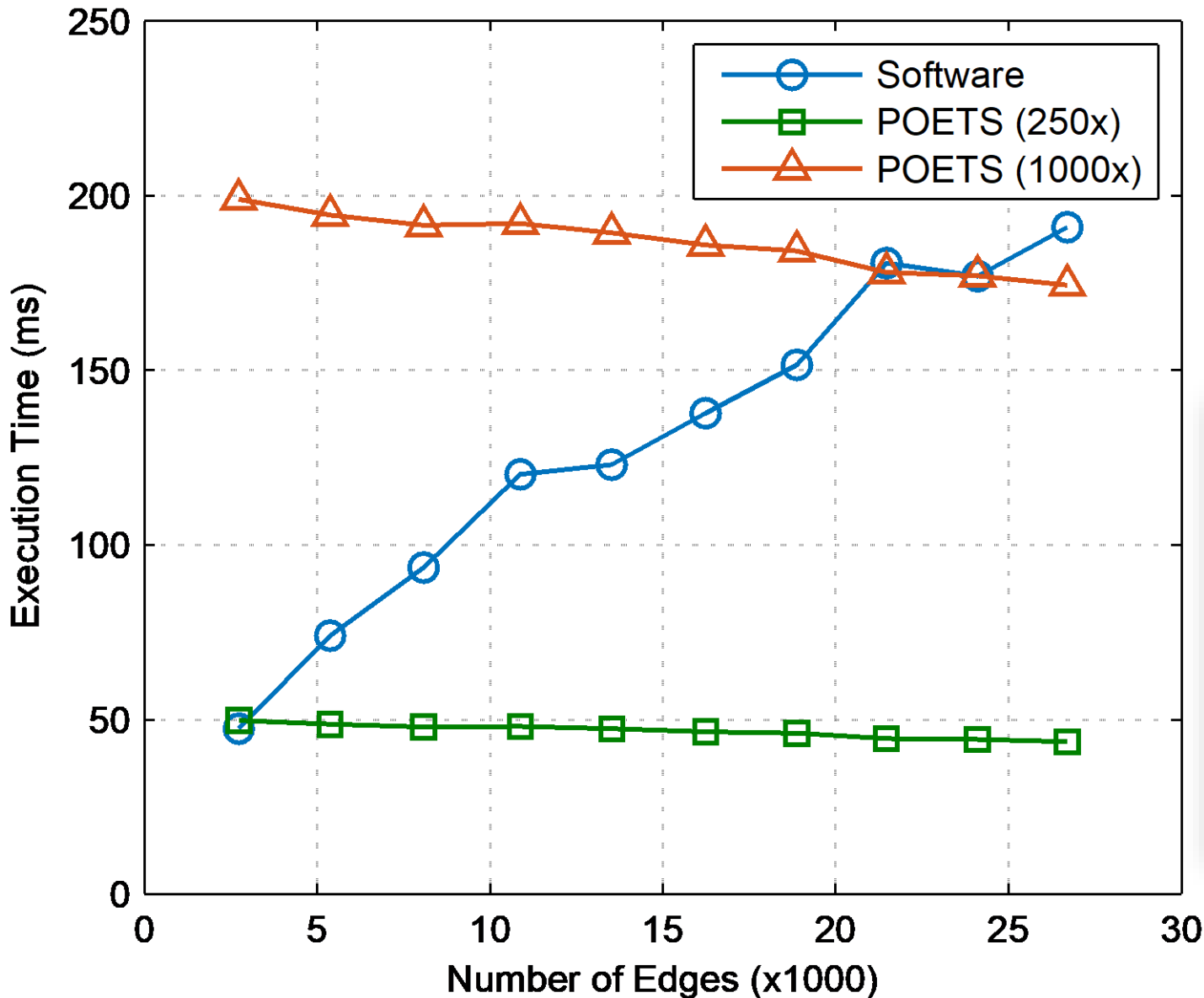


# After 7 clock cycles... Done!





# Unusual Performance Scaling



15% decrease

The accelerator is  
**faster on denser  
networks** because  
of smaller diameter.

**Main idea:** keep your application graphs in FPGA instead of memory

Achieved **1000x** acceleration of impact analysis on real biological networks

Ongoing integration into industrial drug discovery workflows

## Limitations:

- Fixed underlying network
- Scaling beyond one FPGA is difficult (but possible with POETS architecture)
- Weighted graphs require more sophisticated cores

## Future work:

- Fit the complete human protein-interaction network in FPGA
- Explore other application areas, e.g. energy grid or transportation networks

# Thank you!

[andrey.mokhov@ncl.ac.uk](mailto:andrey.mokhov@ncl.ac.uk)

Algebraic graphs: <https://github.com/snowleopard/alga>

Centrifuge project: <https://github.com/tuura/centrifuge>