COL380

Introduction to
Parallel & Distributed Programming

Agenda

- Instruction latency and overlap
- Core organization
- Inter-communication
- Parallel programming models

SIMD

```
float *d1, *d2;
Loop: d1[I] += d2[i];
```

```
movss  xmm0,DWORD PTR [rdi+rax*1]
addss  xmm0,DWORD PTR [rsi+rax*1]
movss  DWORD PTR [rdi+rax*1],xmm0
```

```
float *d1, *d2;
Loop: d1[I] += d2[i];
```

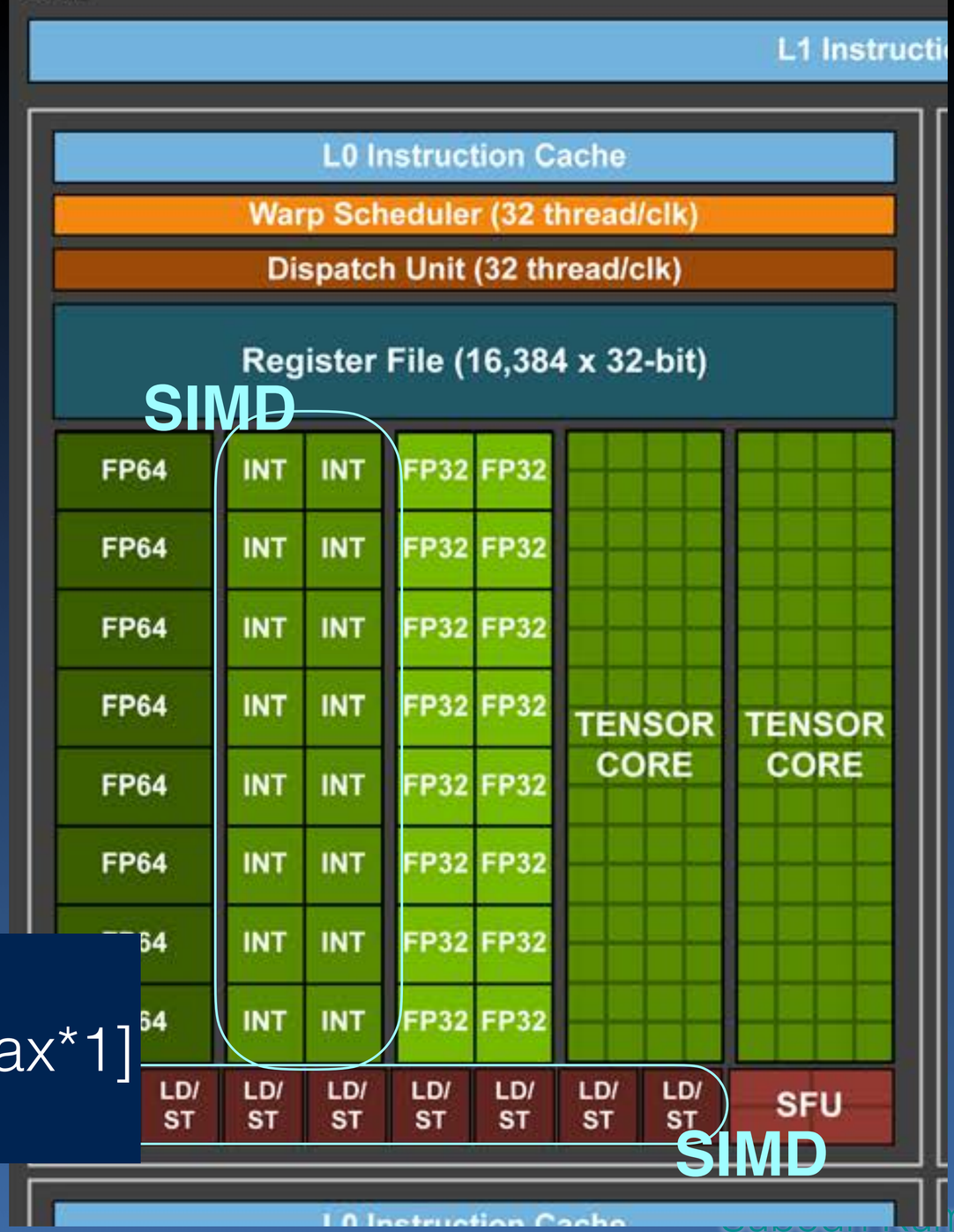
```
movss  xmm0,DWORD PTR [rdi+rax*1]
addss  xmm0,DWORD PTR [rsi+rax*1]
movss  DWORD PTR [rdi+rax*1],xmm0
```

```
float *d1, *d2;
Loop: d1[I] += d2[i];
```

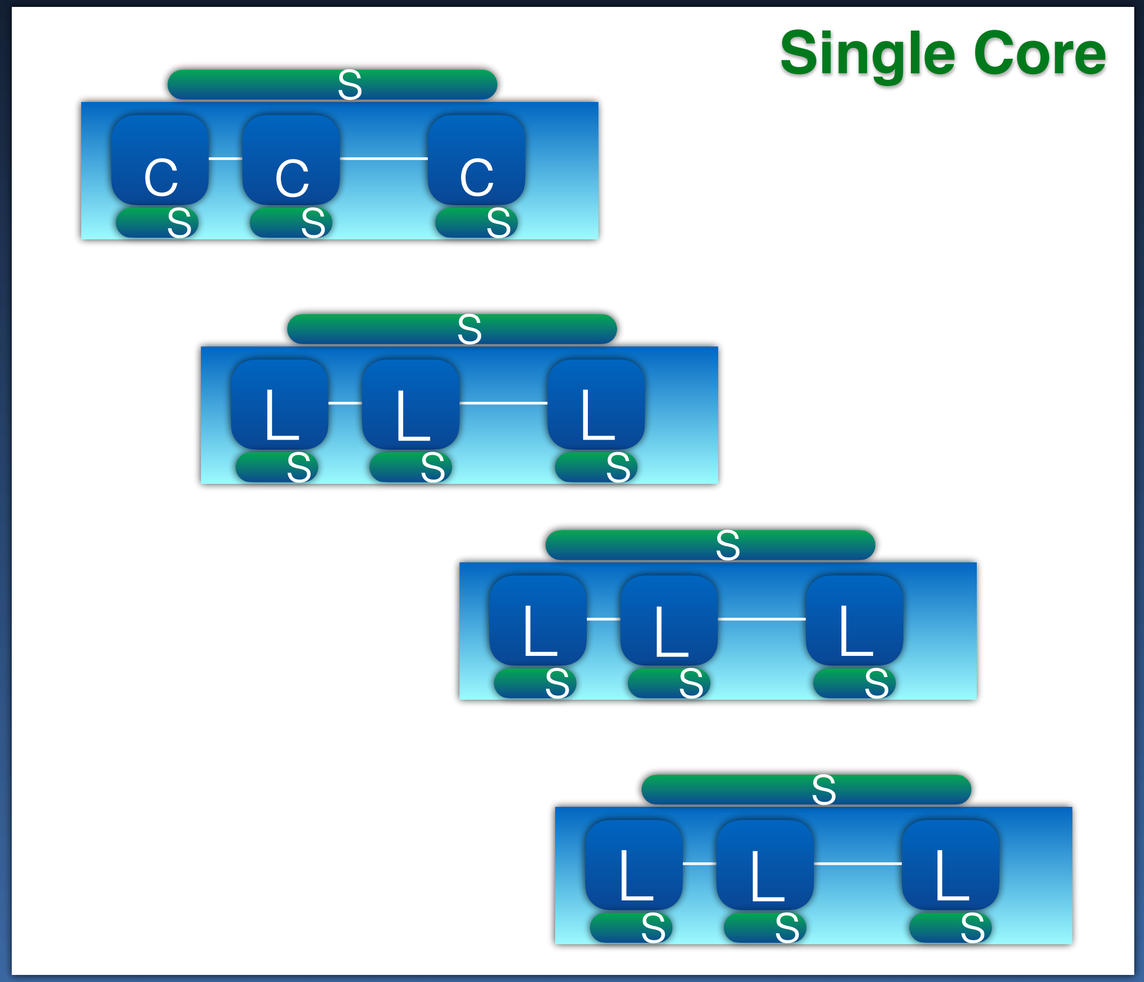
```
movss  xmm0,DWORD PTR [rdi+rax*1]
addss  xmm0,DWORD PTR [rsi+rax*1]
movss  DWORD PTR [rdi+rax*1],xmm0
```

```
float *d1, *d2;
```

```
vmovss xmm0,DWORD PTR [rdi+rax*1]
vaddss xmm0,xmm0,DWORD PTR [rsi+rax*1]
vmovss DWORD PTR [rdi+rax*1],xmm0
```



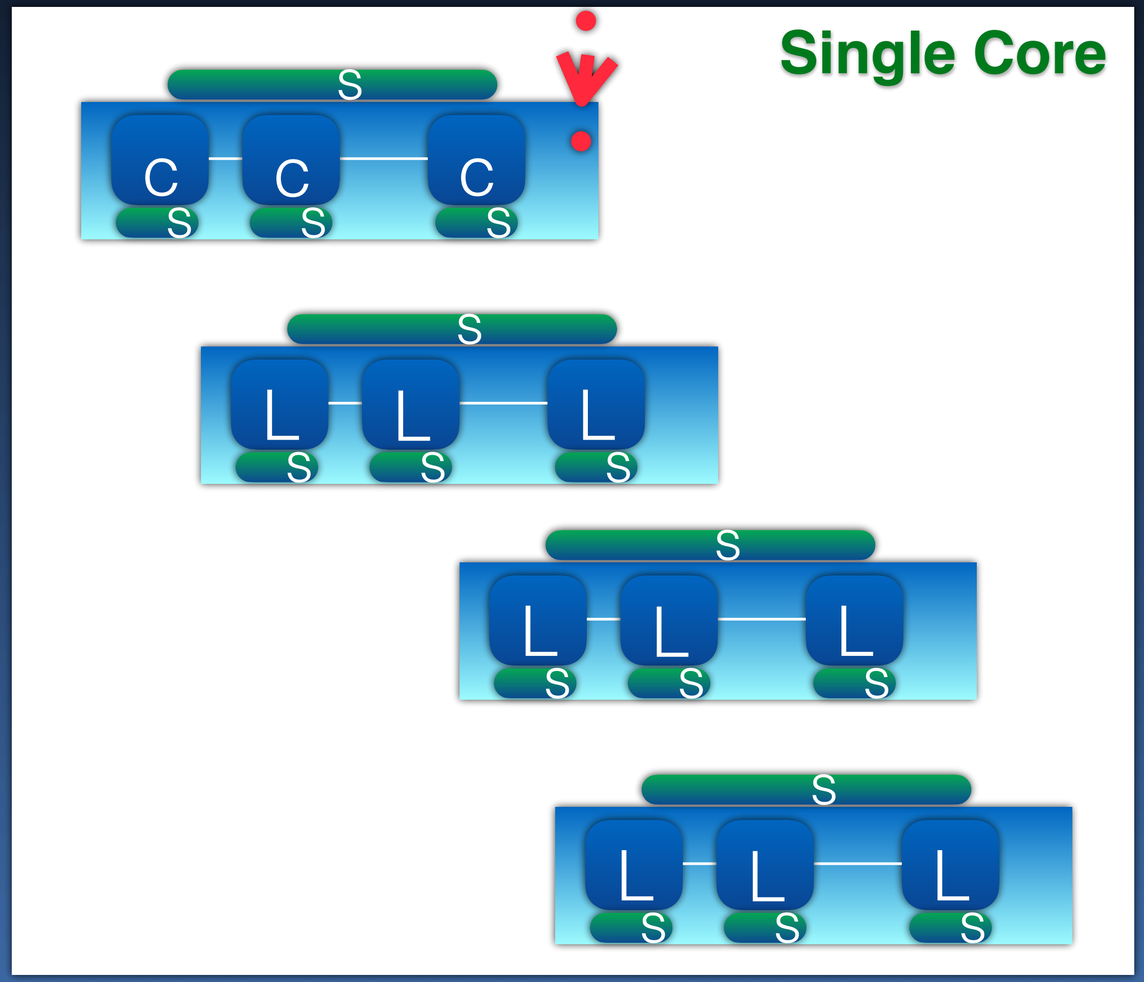
NOT Sequential



State

NOT Sequential

Instructions

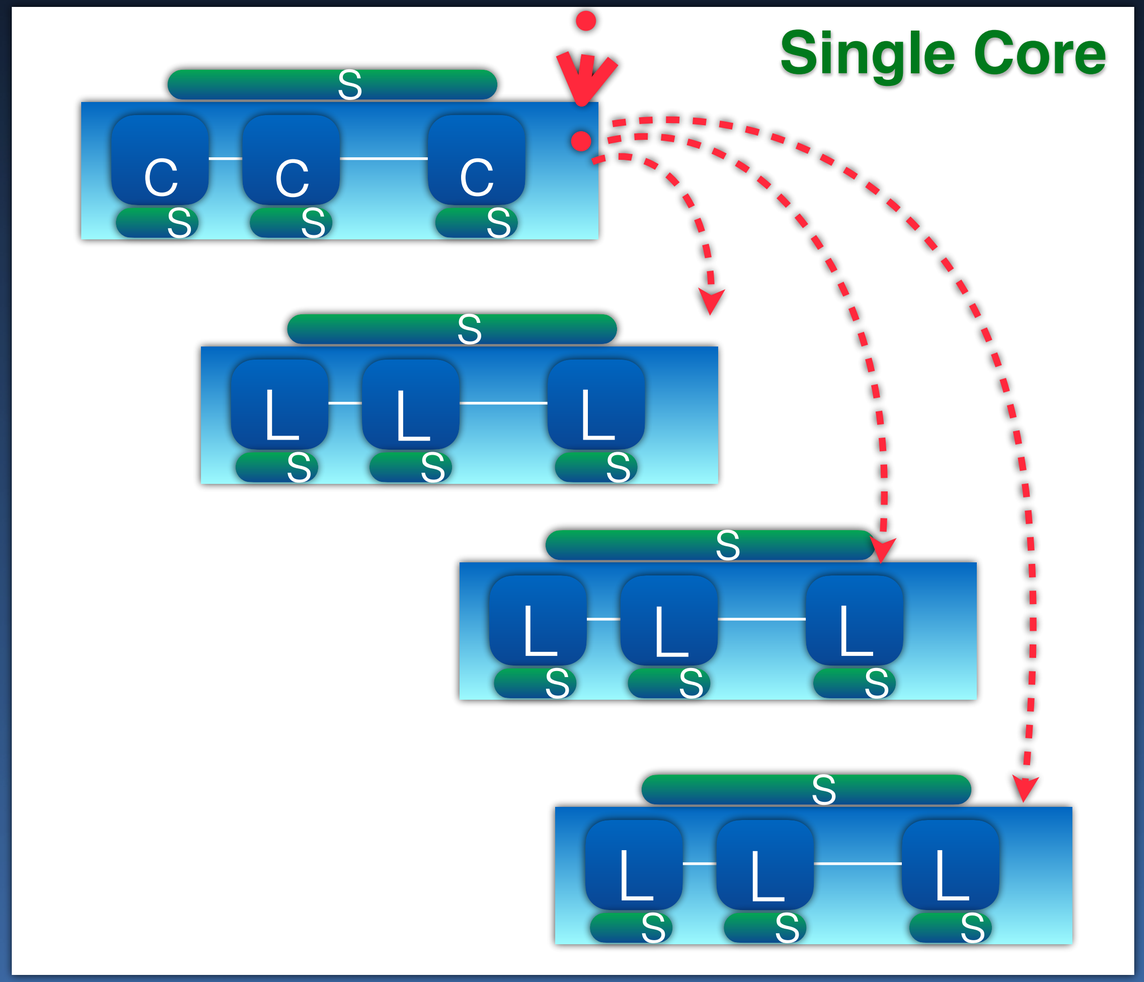


State

Subodh Kumar

NOT Sequential

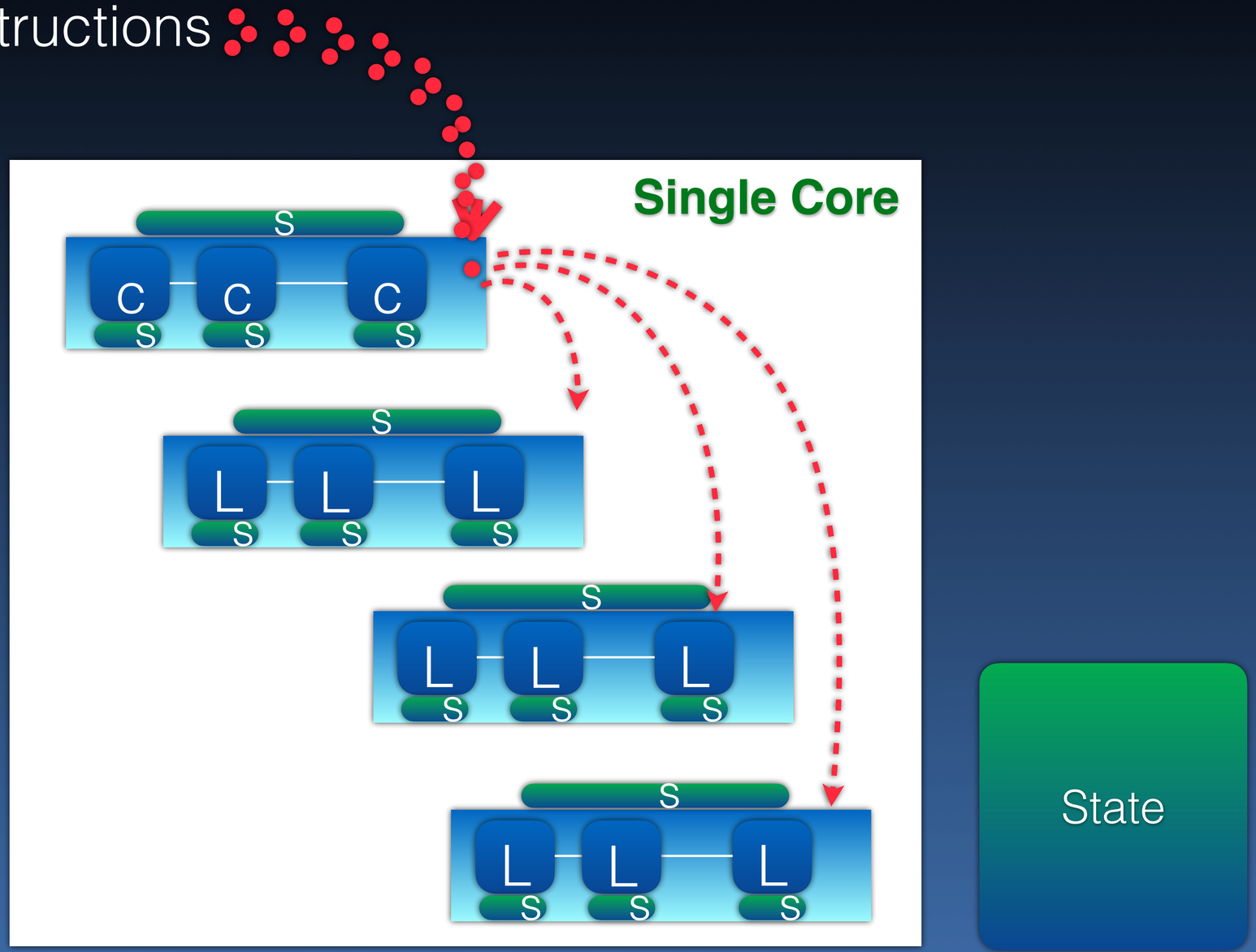
Instructions



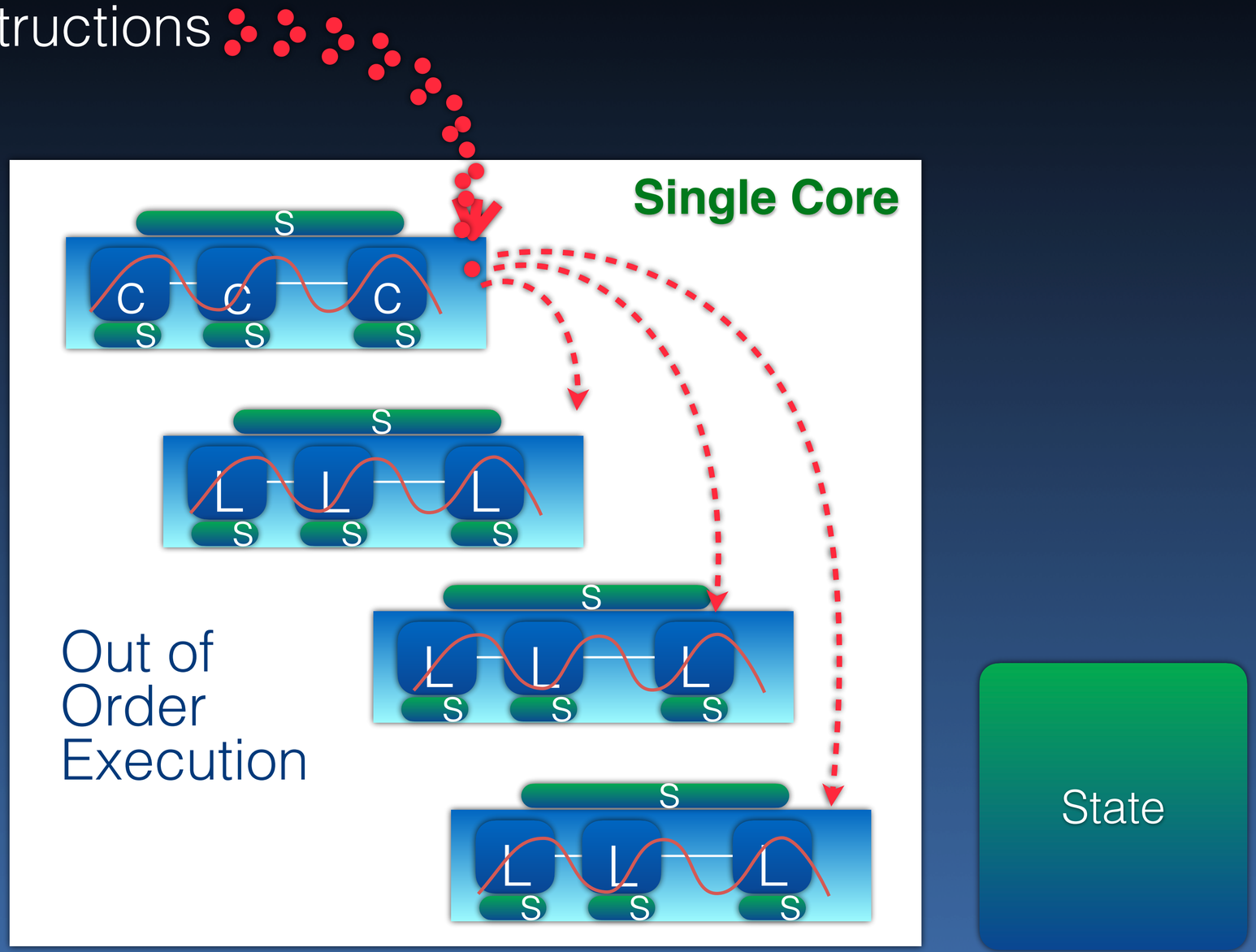
State

Subodh Kumar

NOT Sequential



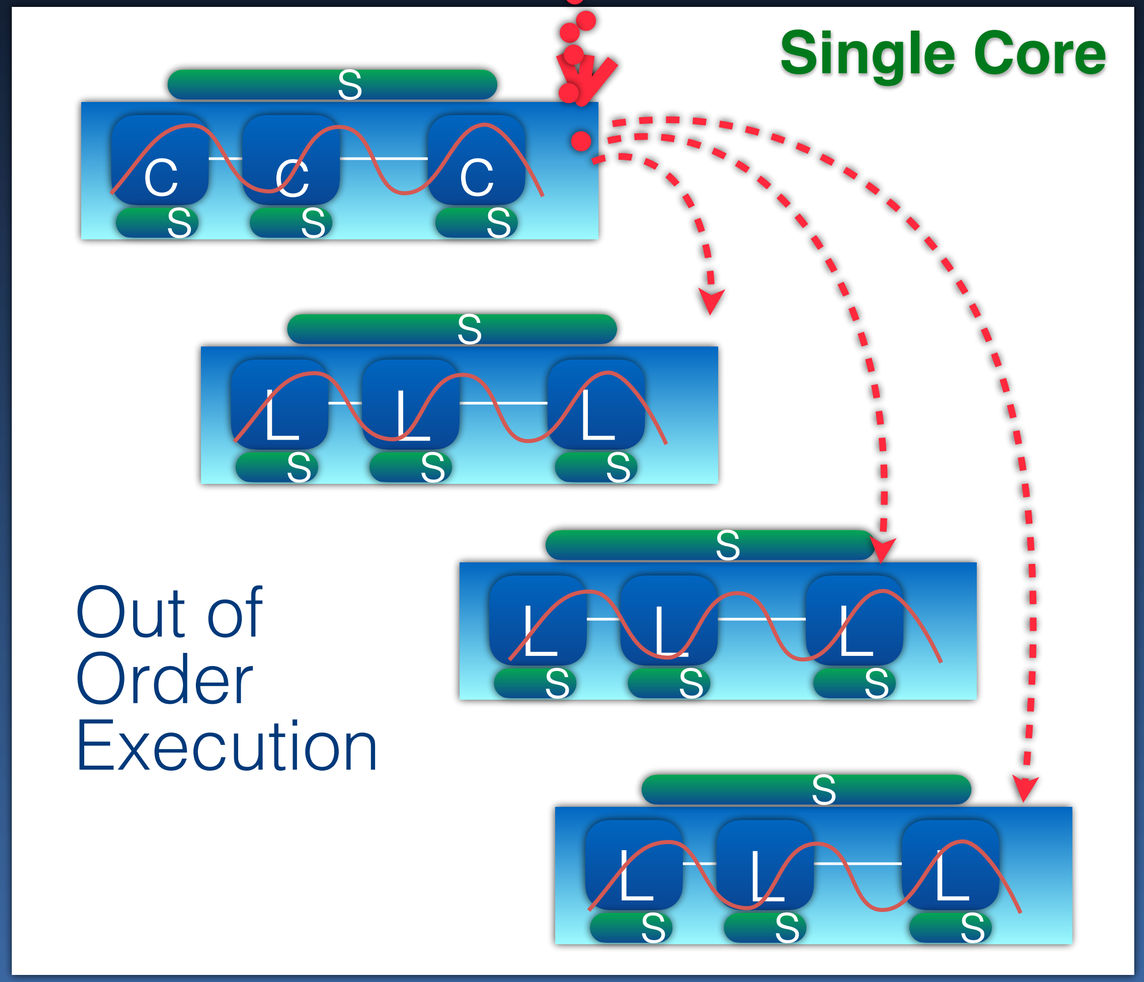
NOT Sequential



Subodh Kumar

NOT Sequential

Instructions



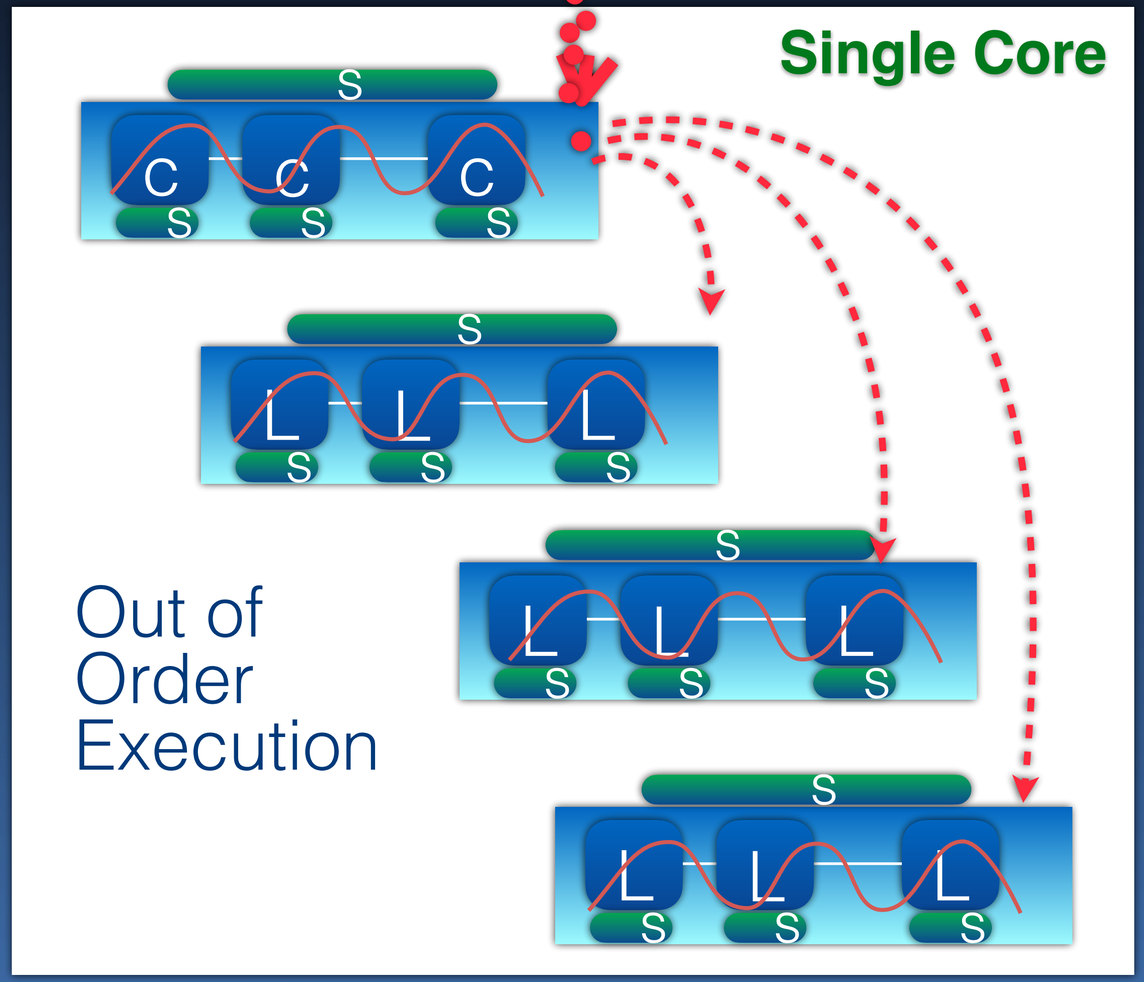
R1 = x
R2 = y
Z = R1+R2

State

Subodh Kumar

NOT Sequential

Instructions



Access x

.

.

R1 = x

R2 = y

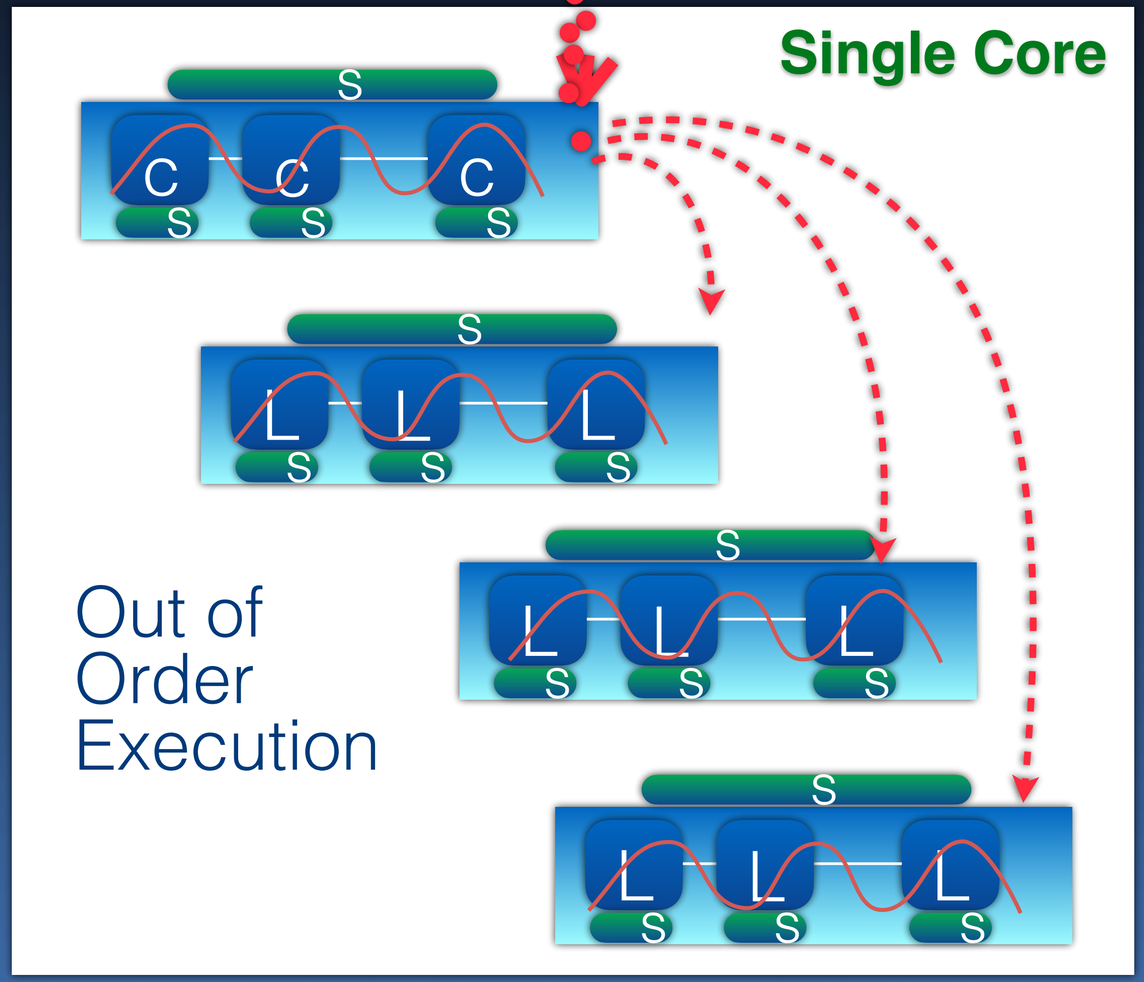
Z = R1+R2

State

Subodh Kumar

NOT Sequential

Instructions



volatile int x;

Access x

.

.

R1 = x

R2 = y

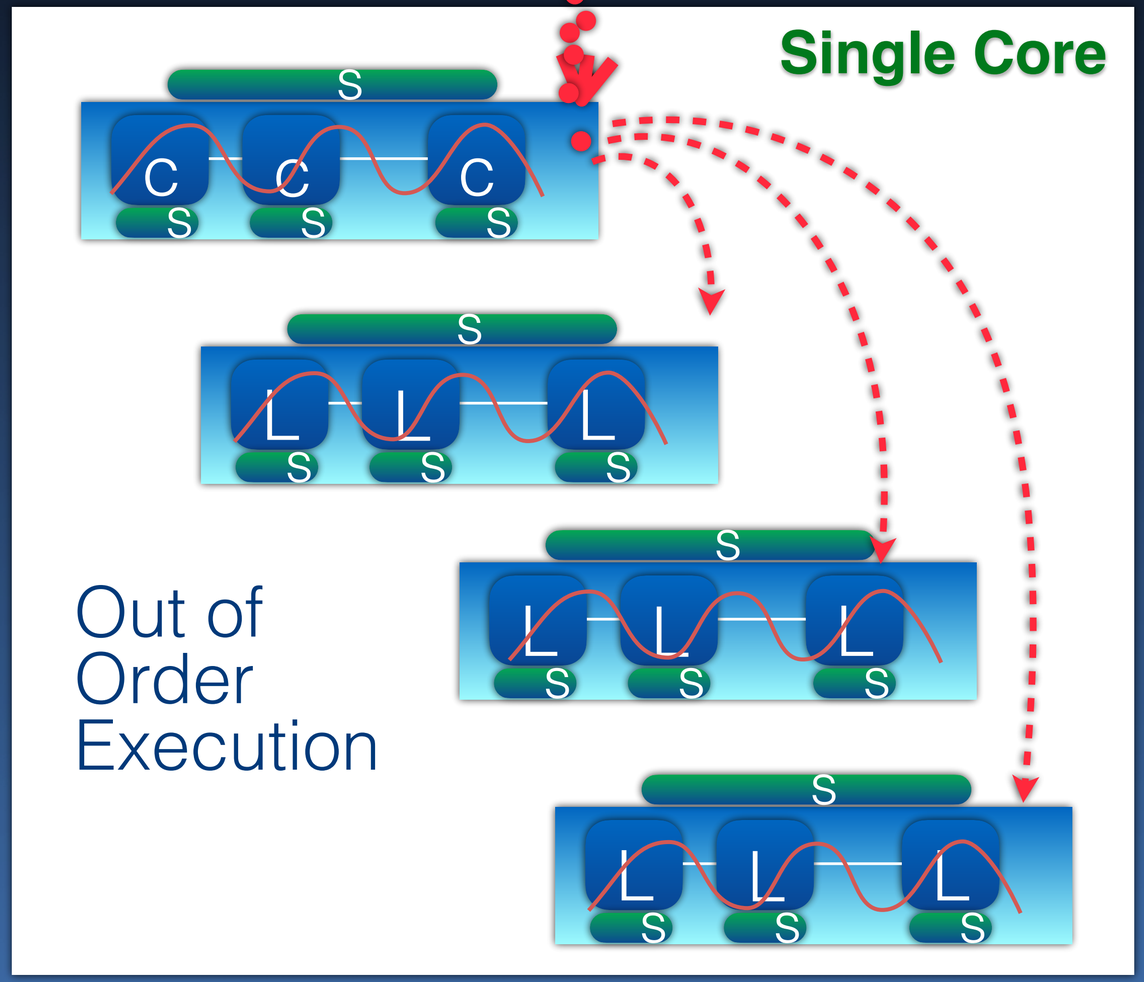
Z = R1+R2

State

Subodh Kumar

NOT Sequential

Instructions



volatile int x;

Access x

.

.

R1 = x

R2 = y

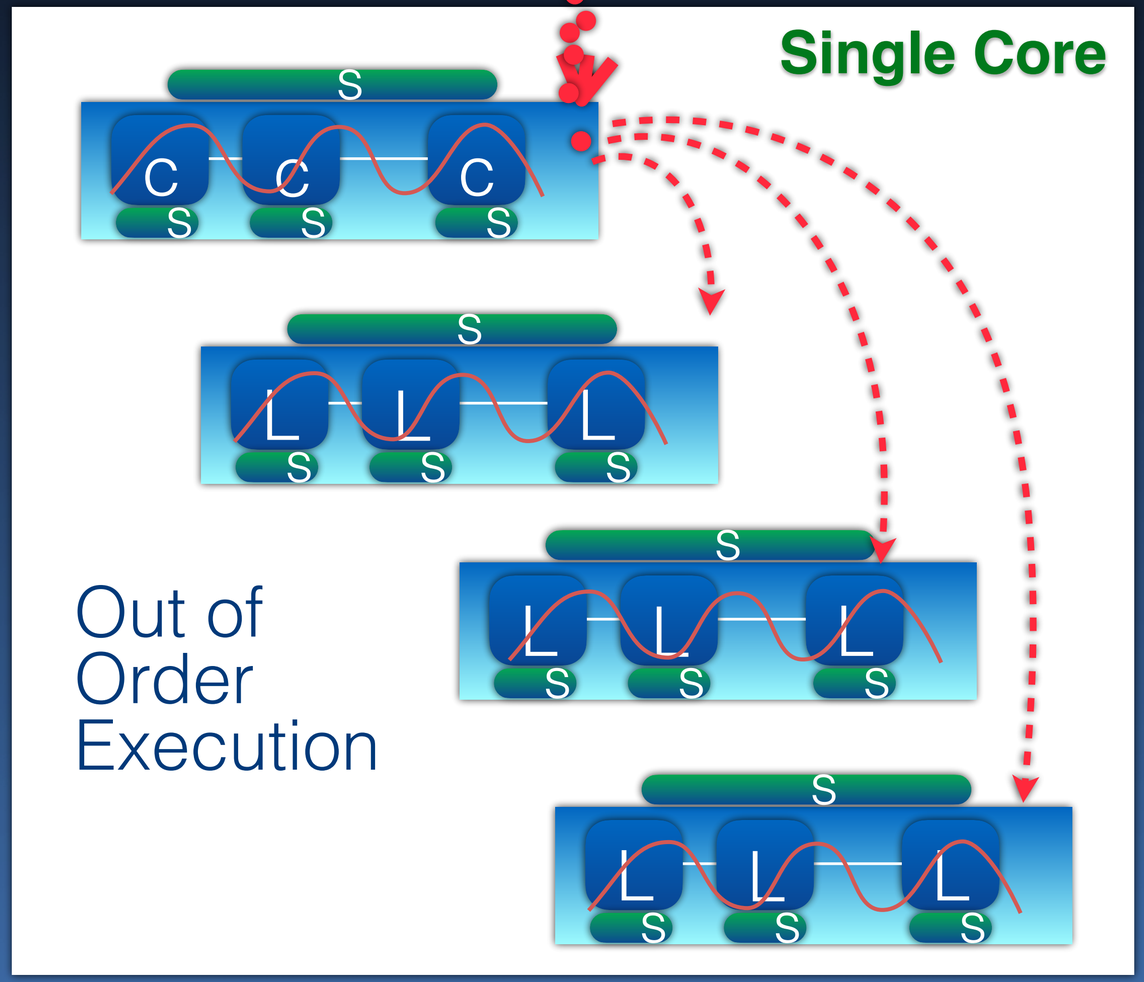
Z = R1+R2

State

Subodh Kumar

NOT Sequential

Instructions



volatile int x;

Access x

.

.

R1 = x

R2 = y

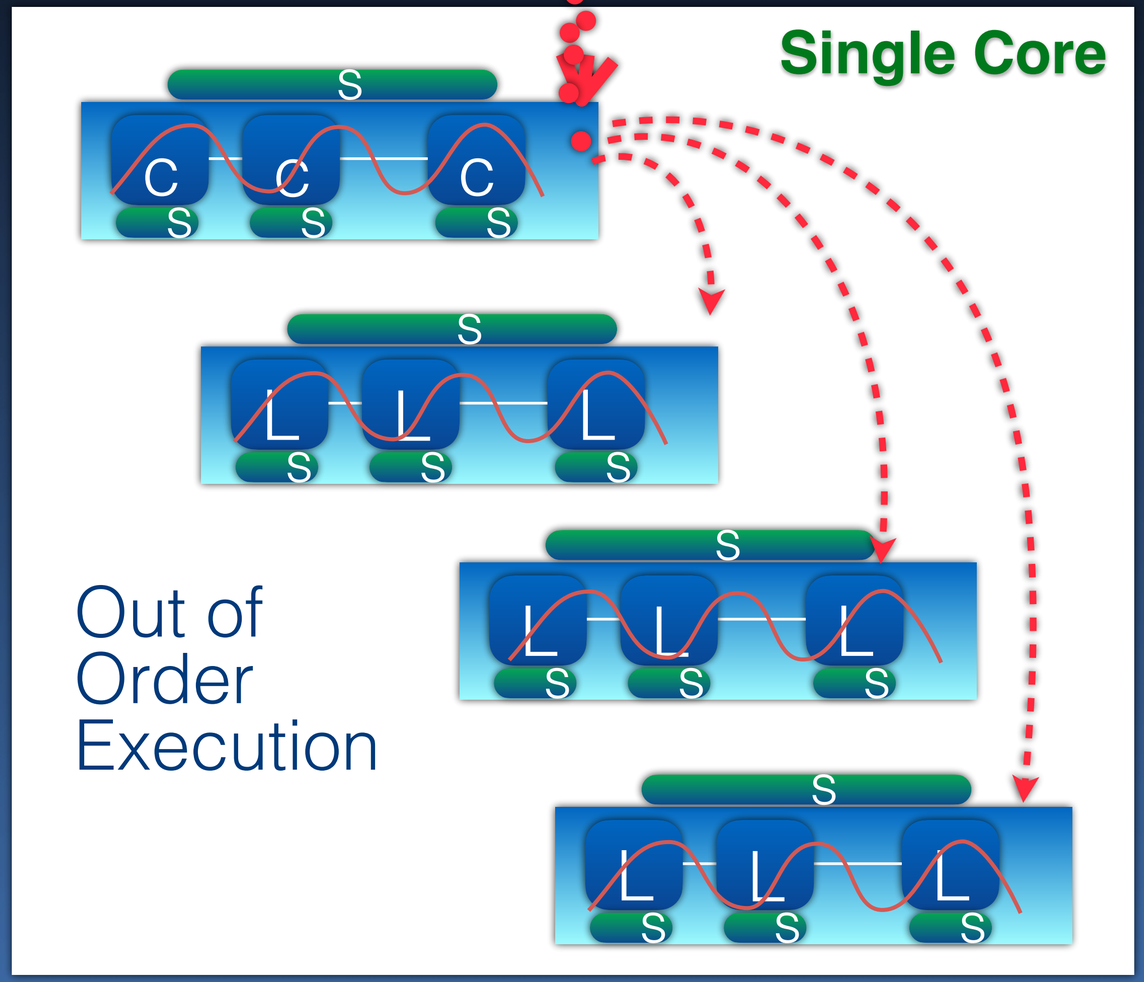
Z = R1+R2

State

Subodh Kumar

NOT Sequential

Instructions



volatile int x;

Access x

.

.

R1 = x

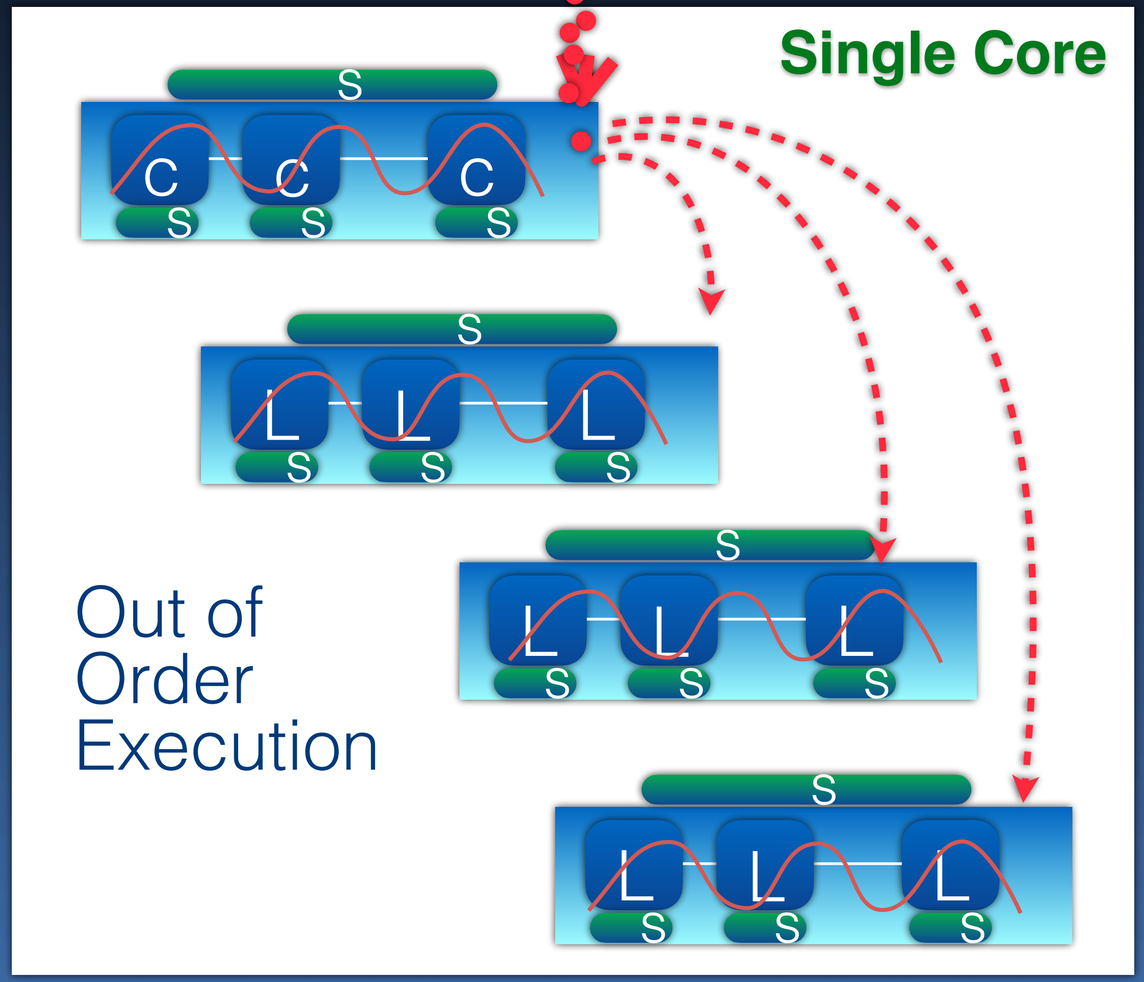
R2 = y

Z = R1+R2

State

NOT Sequential

Instructions



State

Mem Mgmt Unit

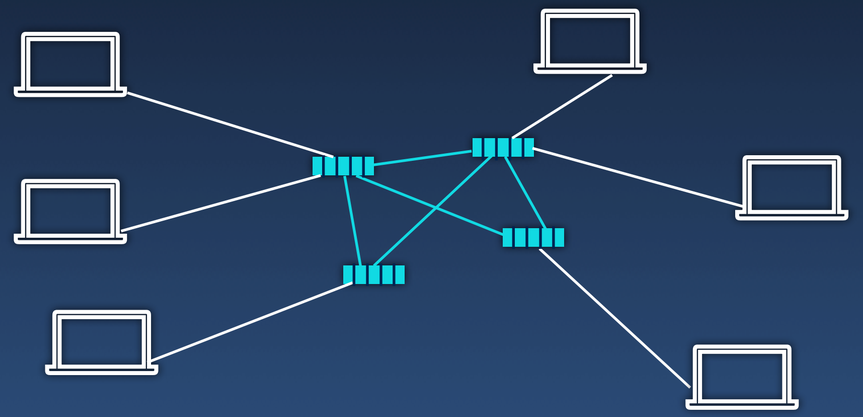
Network Controllers

DMA Engines

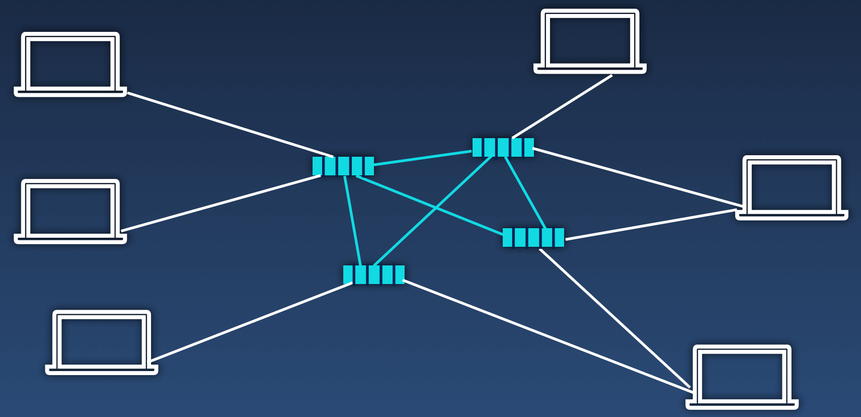
IO controller

Subodh Kumar

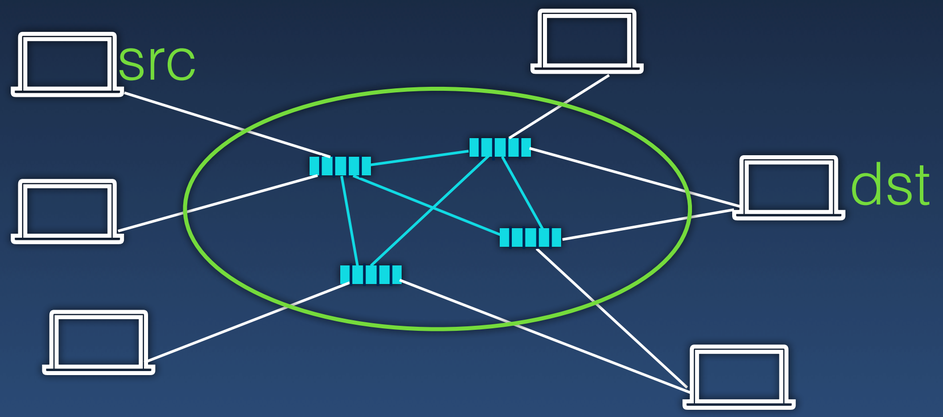
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



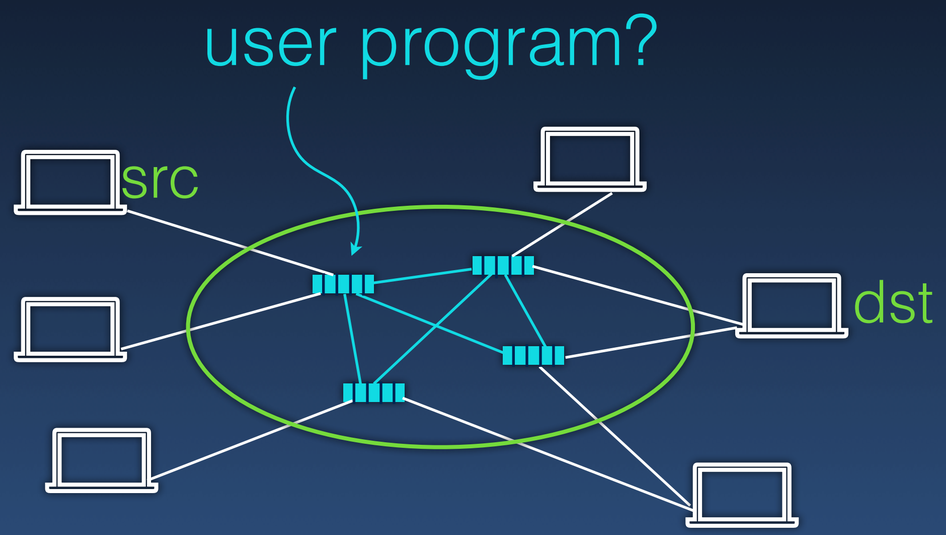
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



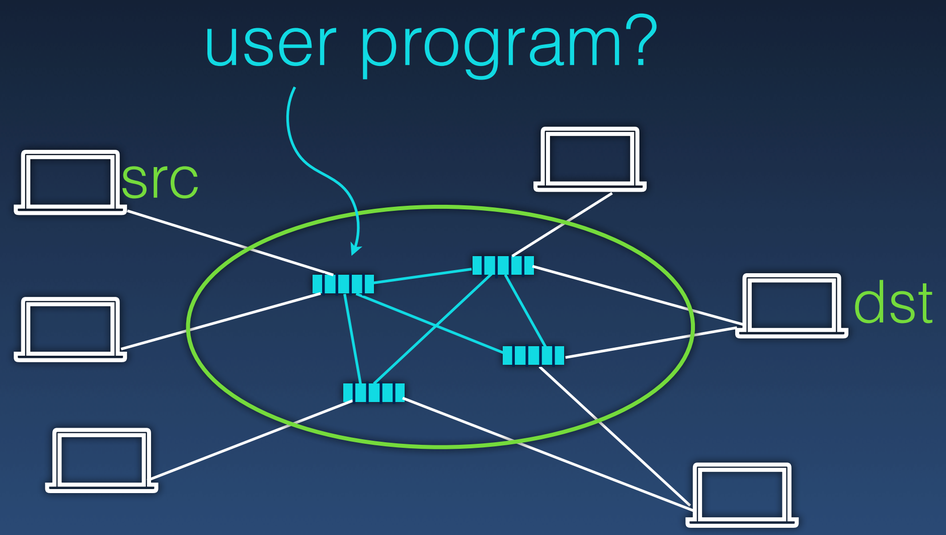
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



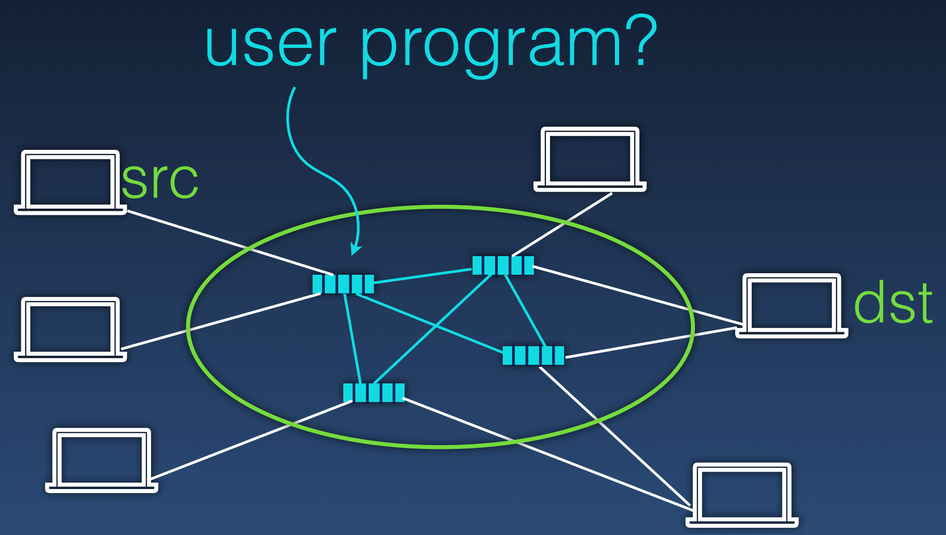
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



- Routing algorithm
  - ➡ Address, Low latency, High bandwidth
- Metrics
  - ➡ Number of links required
  - ➡ Number of ports on a node
  - ➡ Distance between nodes
  - ➡ Redundancy in routes

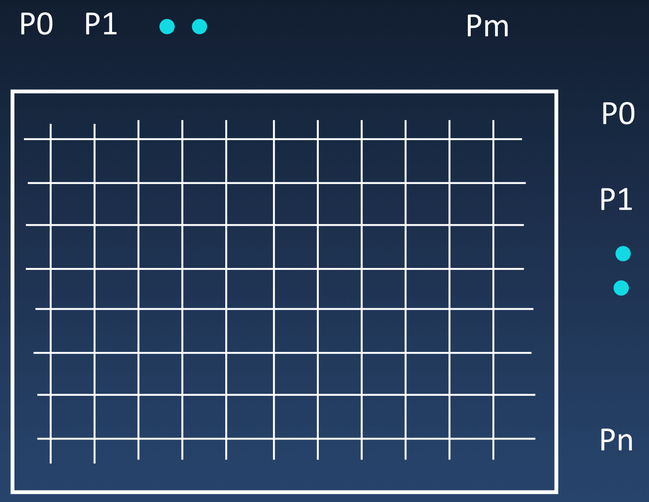


- Routing algorithm
  - ➡ Address, Low latency, High bandwidth
- Metrics
  - ➡ Number of links required
  - ➡ Number of ports on a node
  - ➡ Distance between nodes
  - ➡ Redundancy in routes



- Diameter: Longest path
- Bisection width: Min #links failures to bi-partition the nodes
- Blocking: If independent pairs can communicate at each step

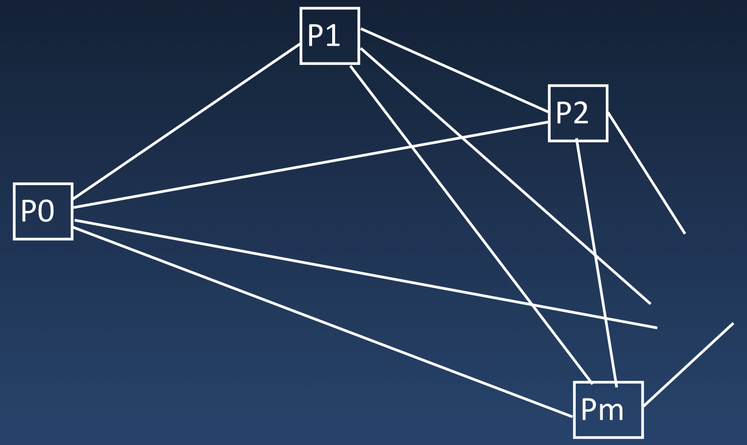
Basic Interconnects



m×*n* Crossbar

connects *m* inputs to *n* outputs

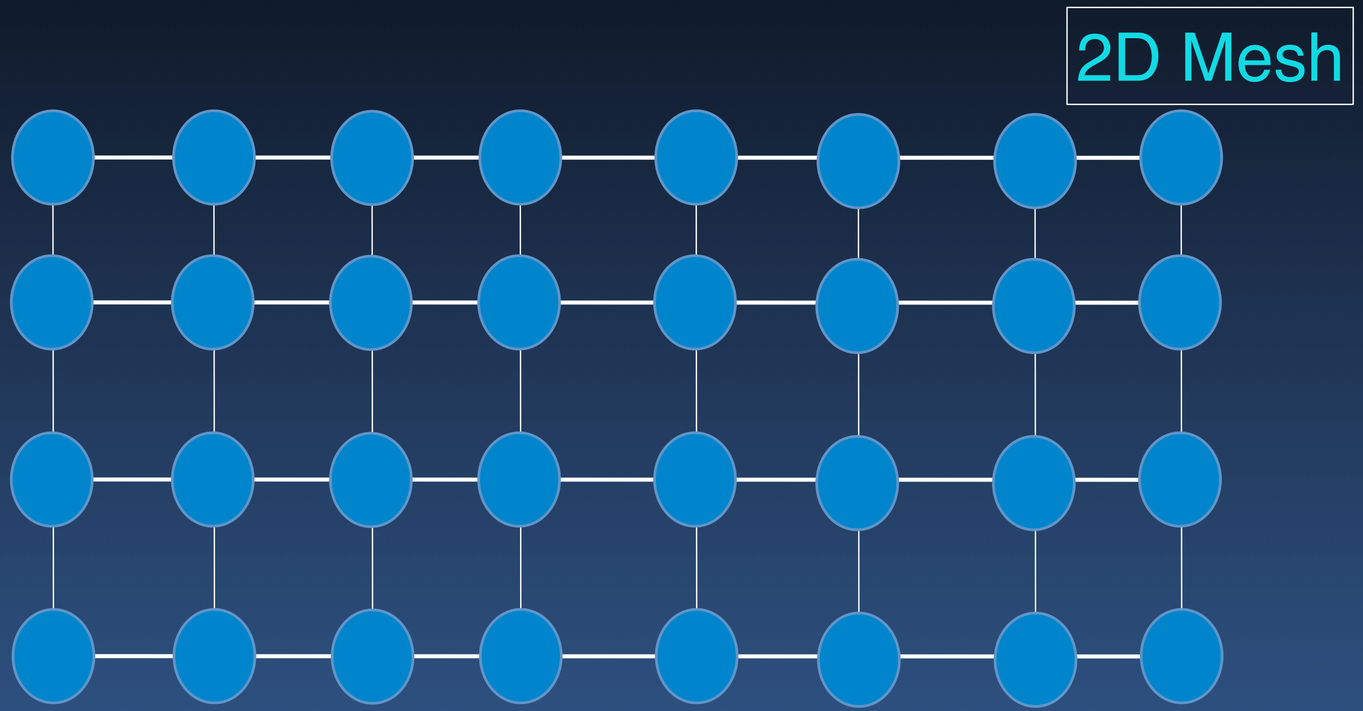
Cost scales well;
Performance does not



n node fully connected network

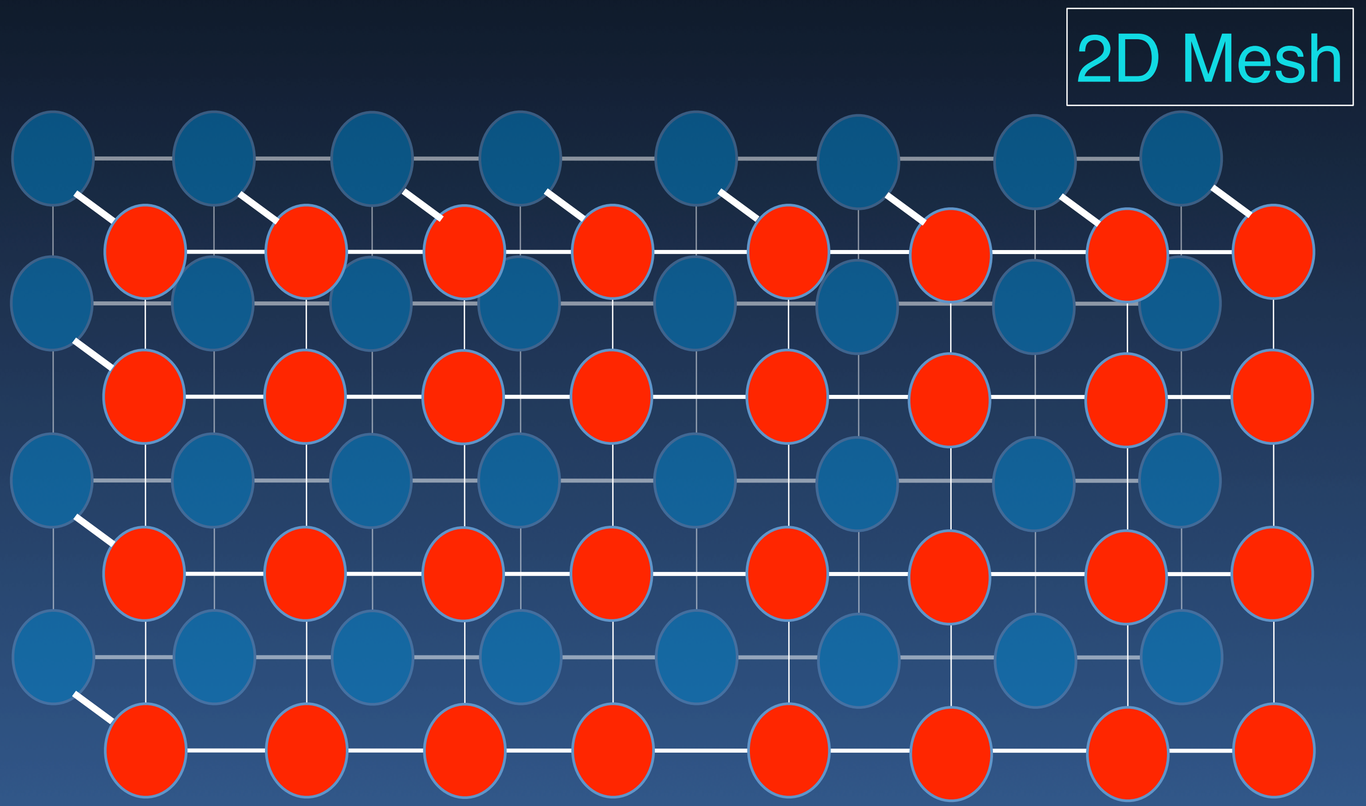
Single clock latency;
Link cost is quadratic, Layout complex

Mesh Network



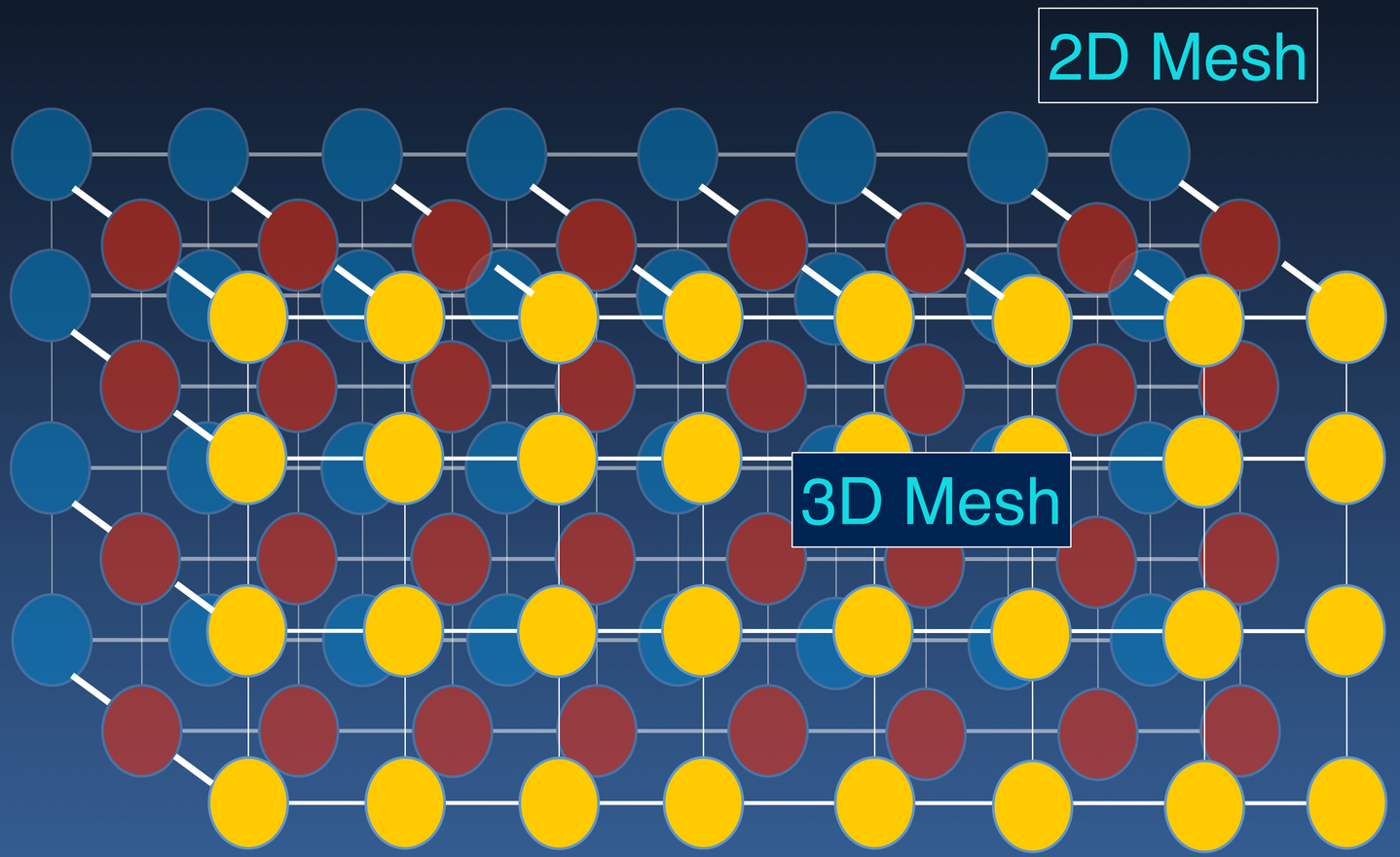
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Mesh Network



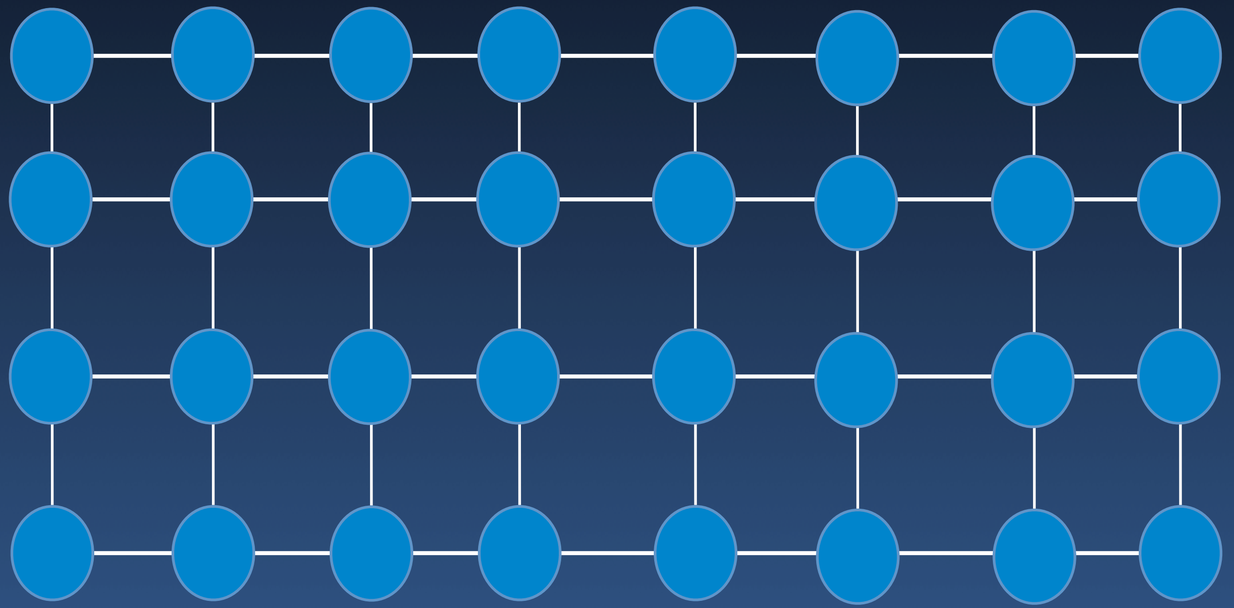
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Mesh Network



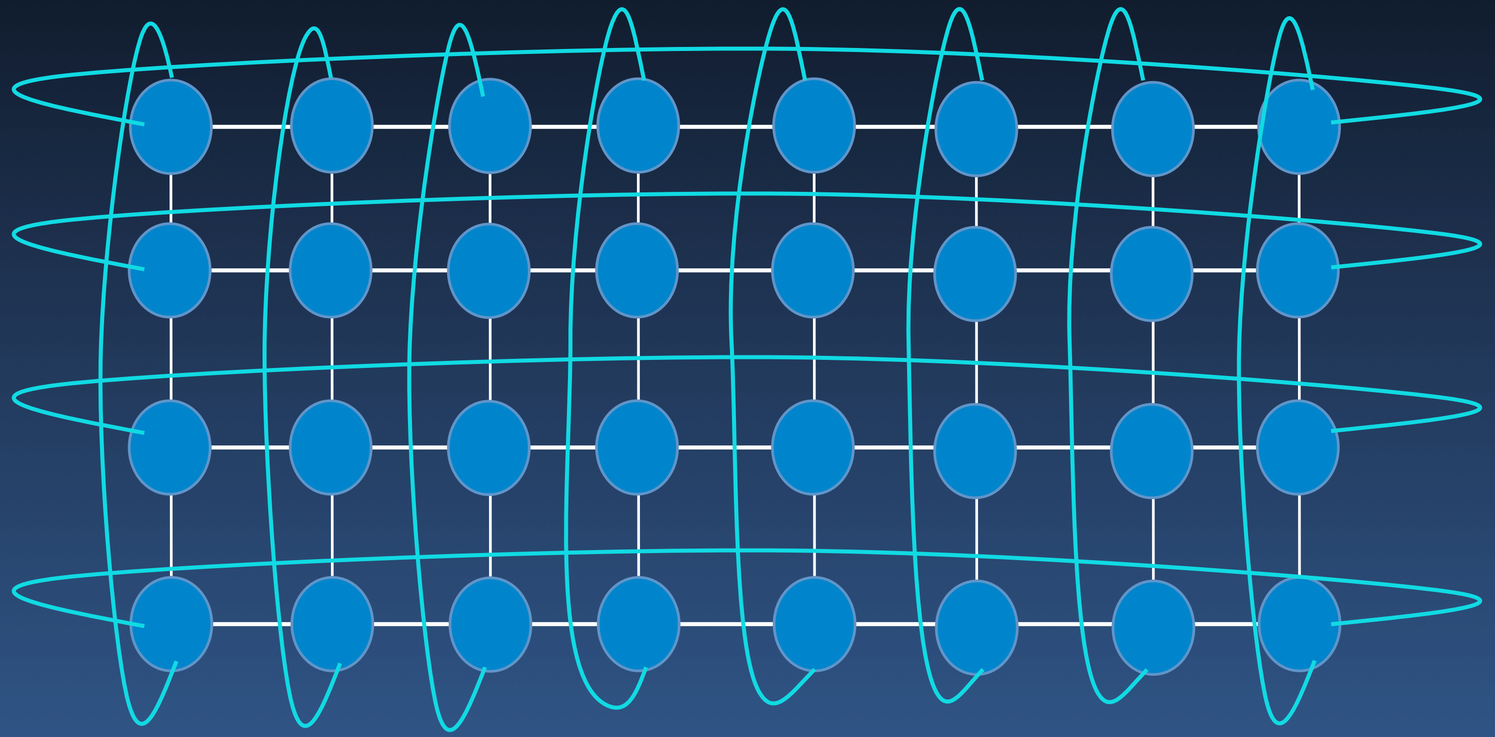
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network



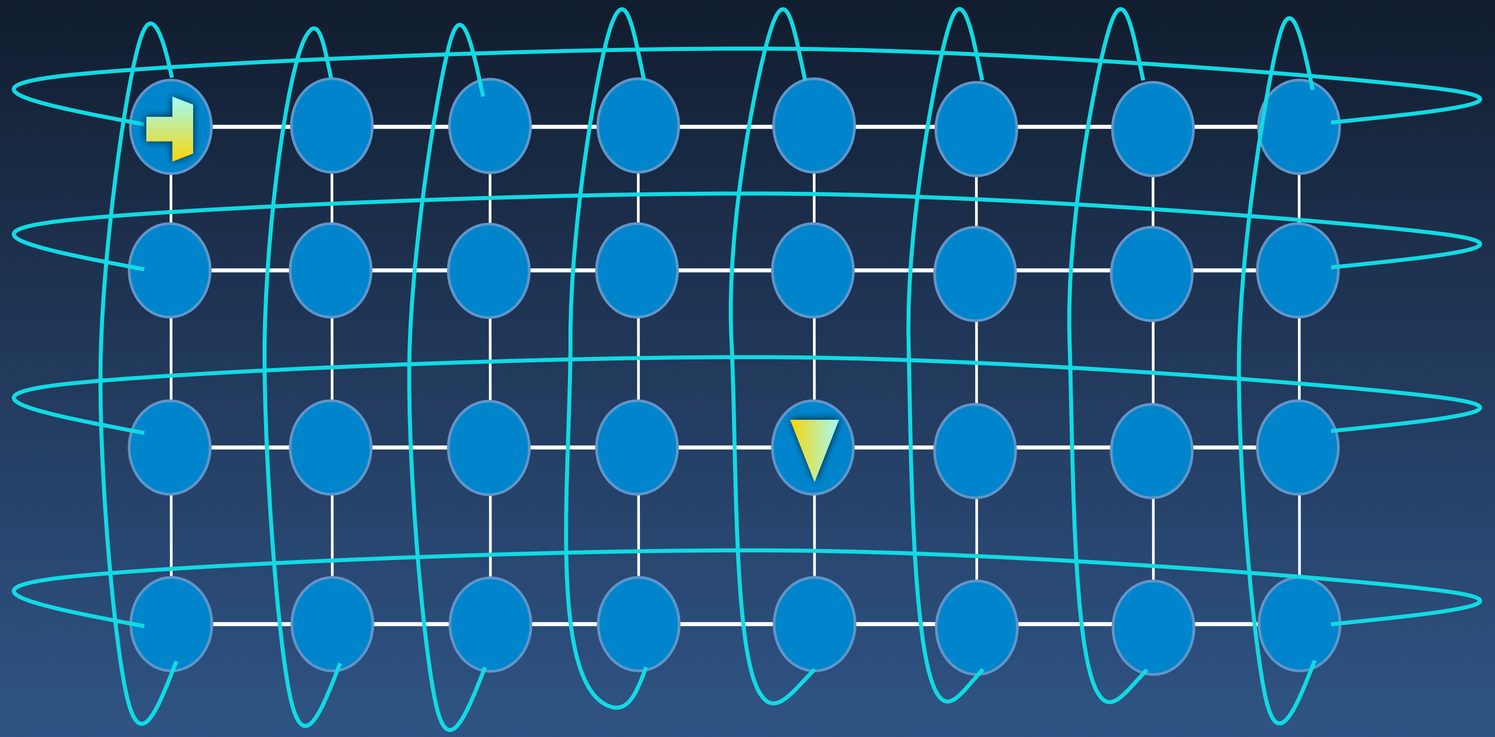
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network



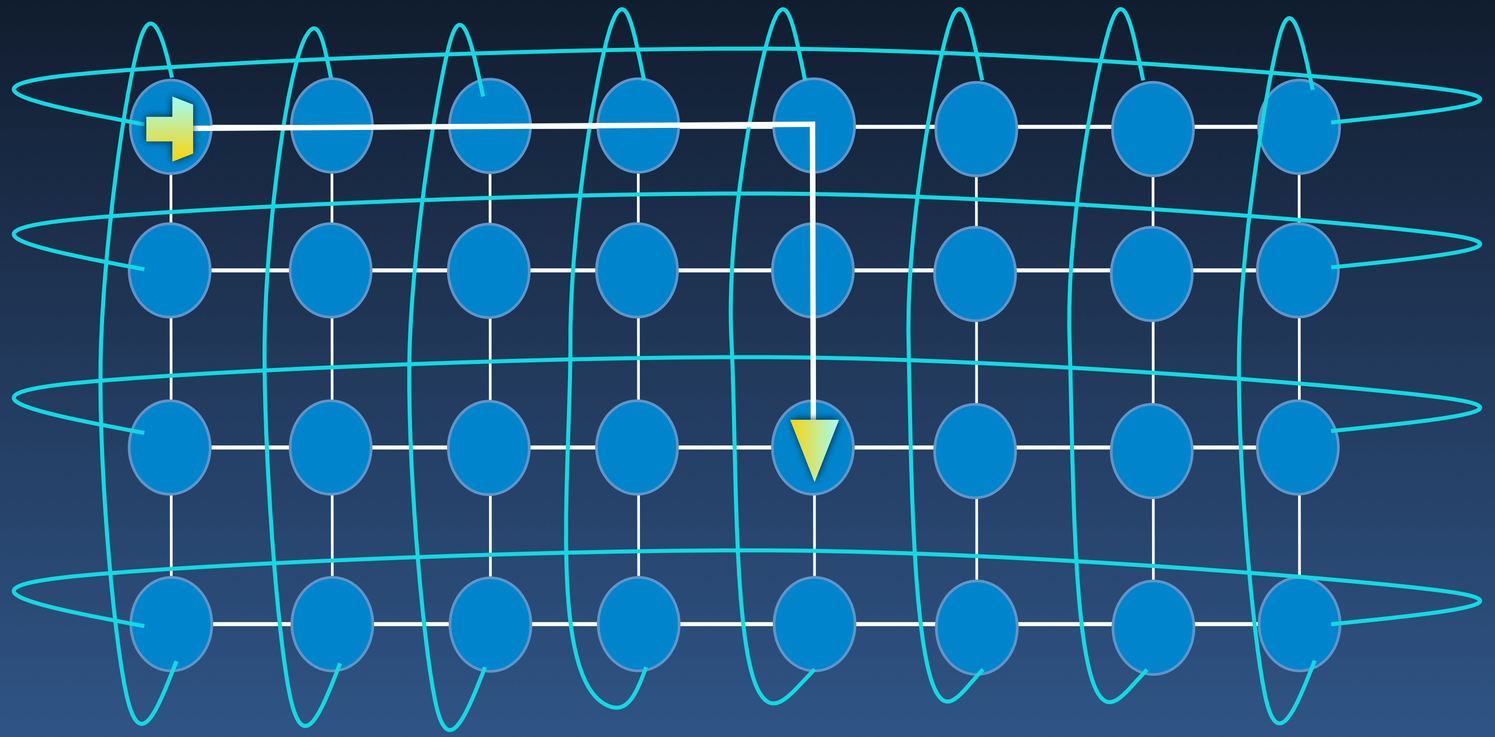
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network



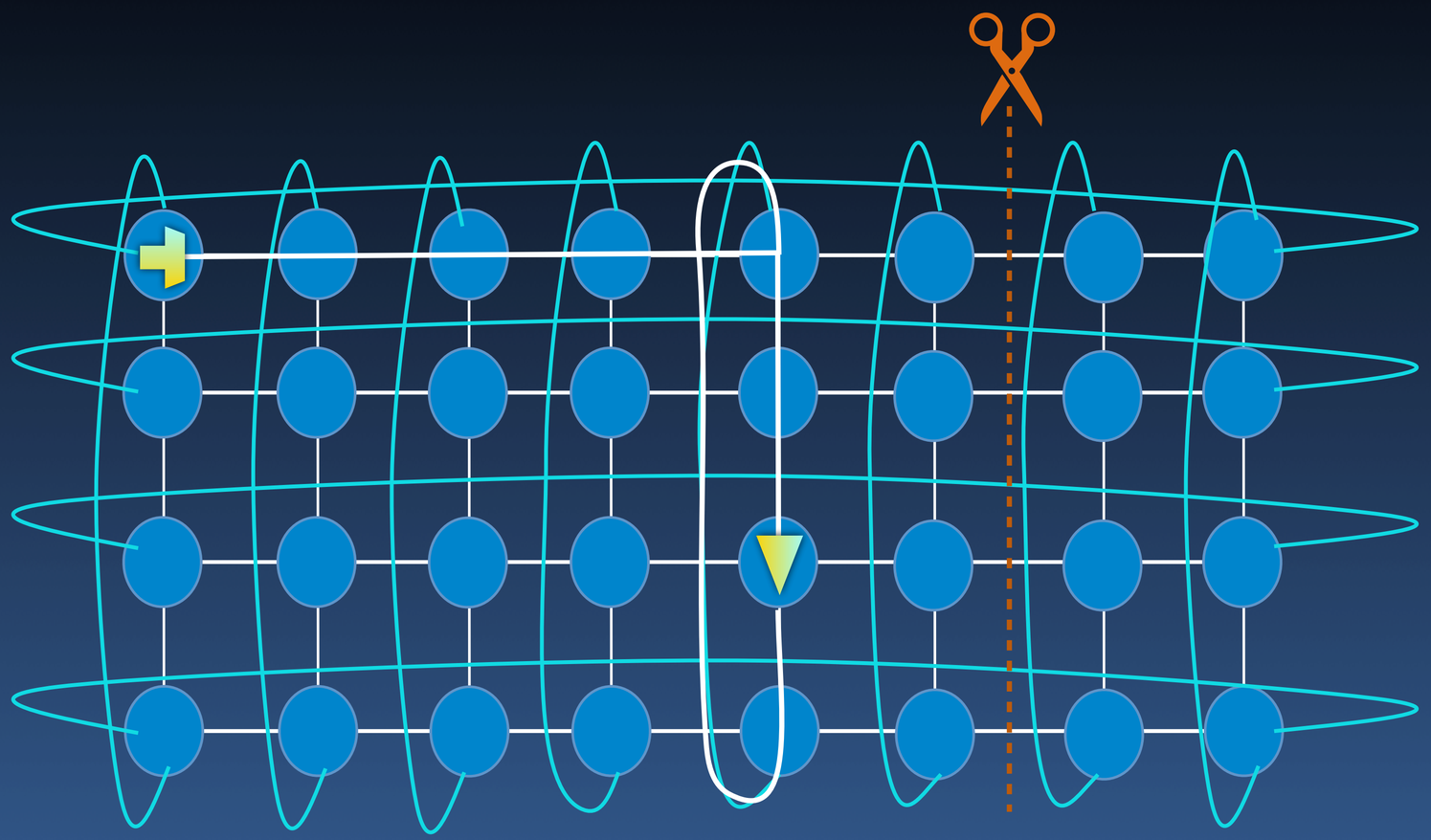
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network



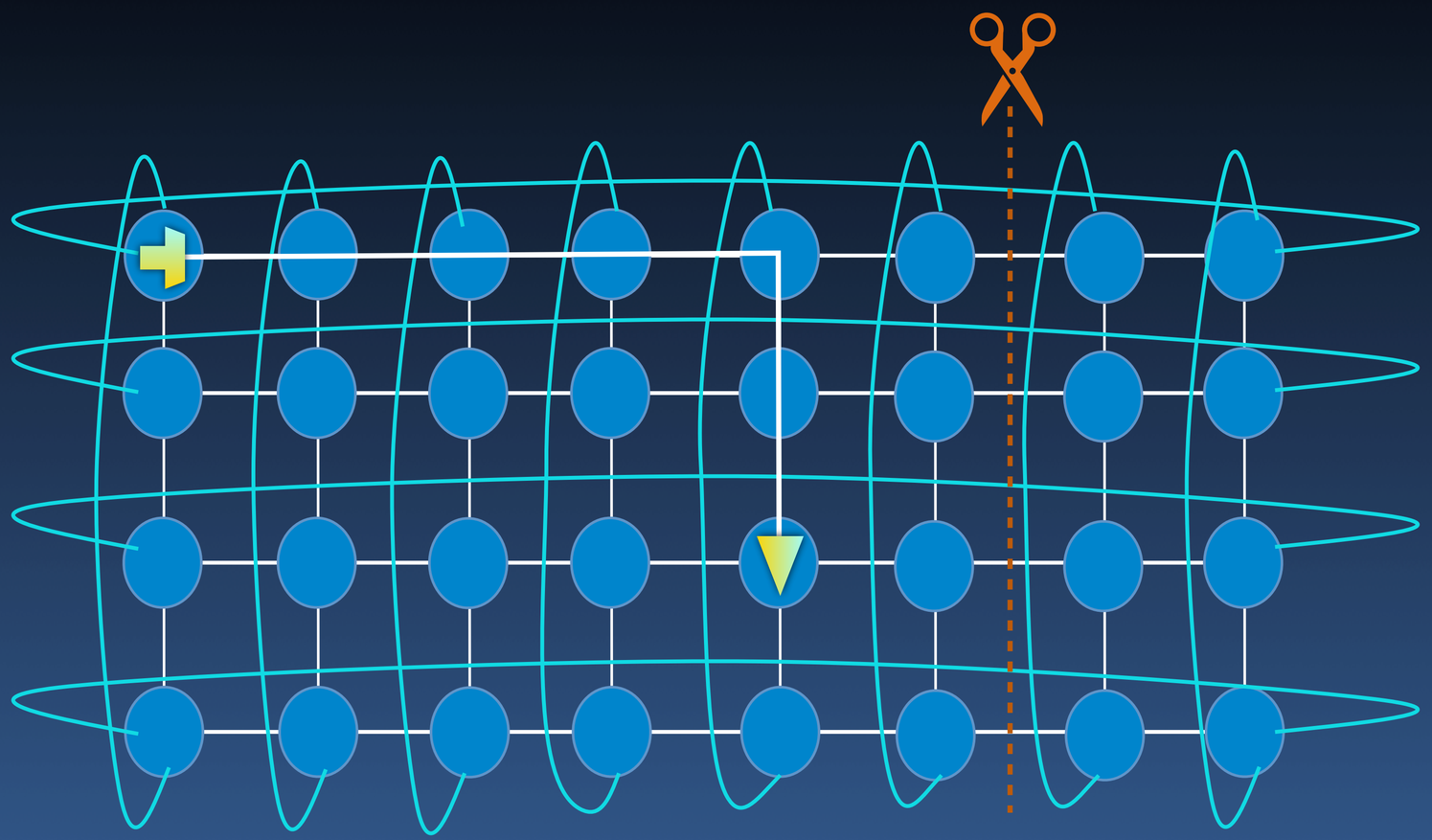
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network



No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Torus Network

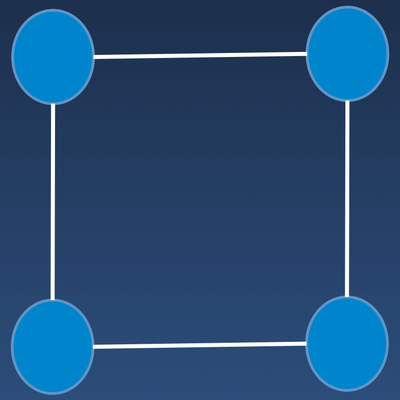


No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Hypercube

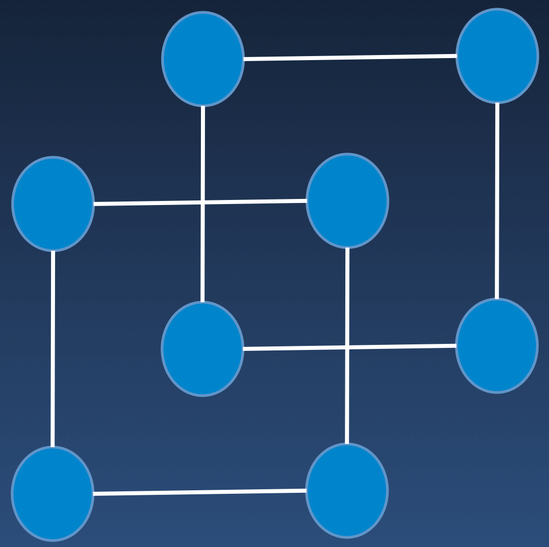
Subodh Kumar

Hypercube

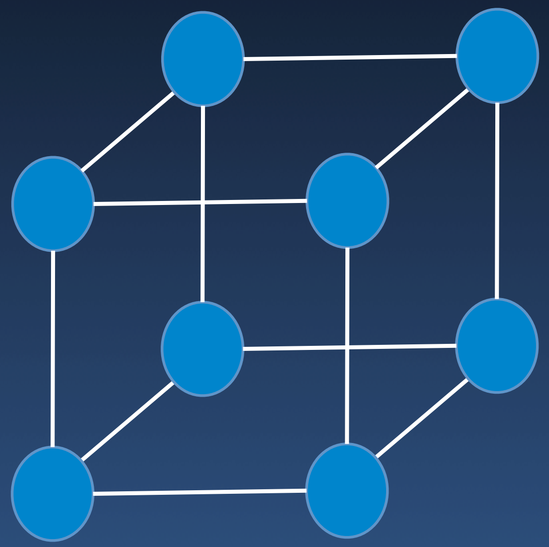


Subodh Kumar

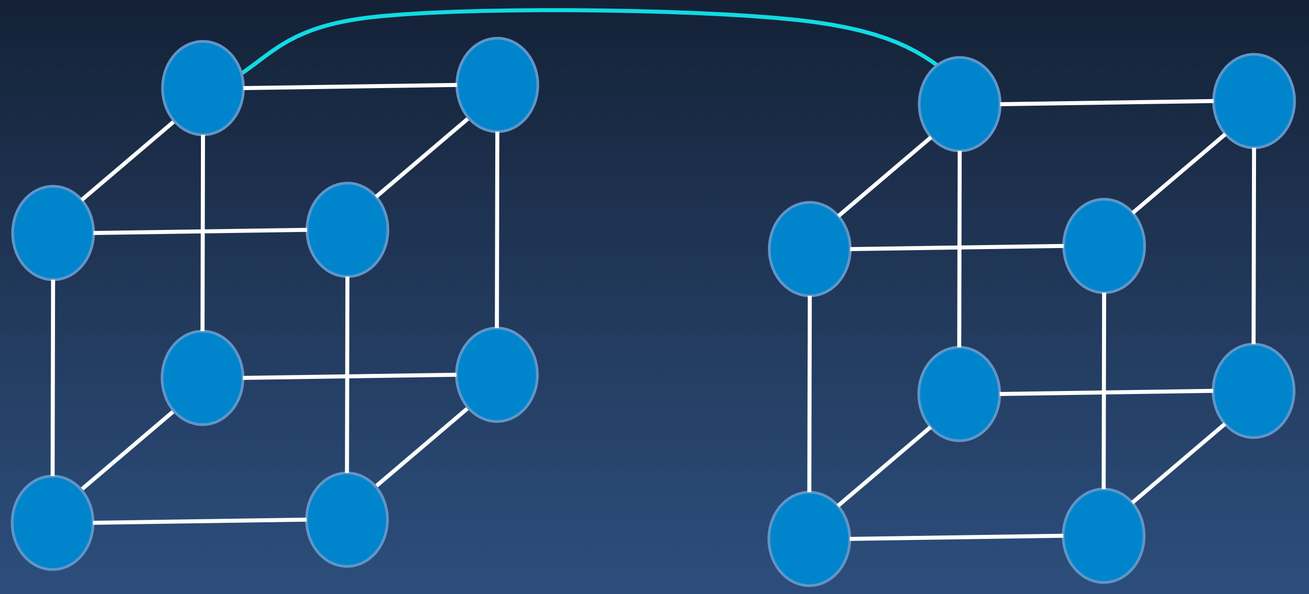
Hypercube



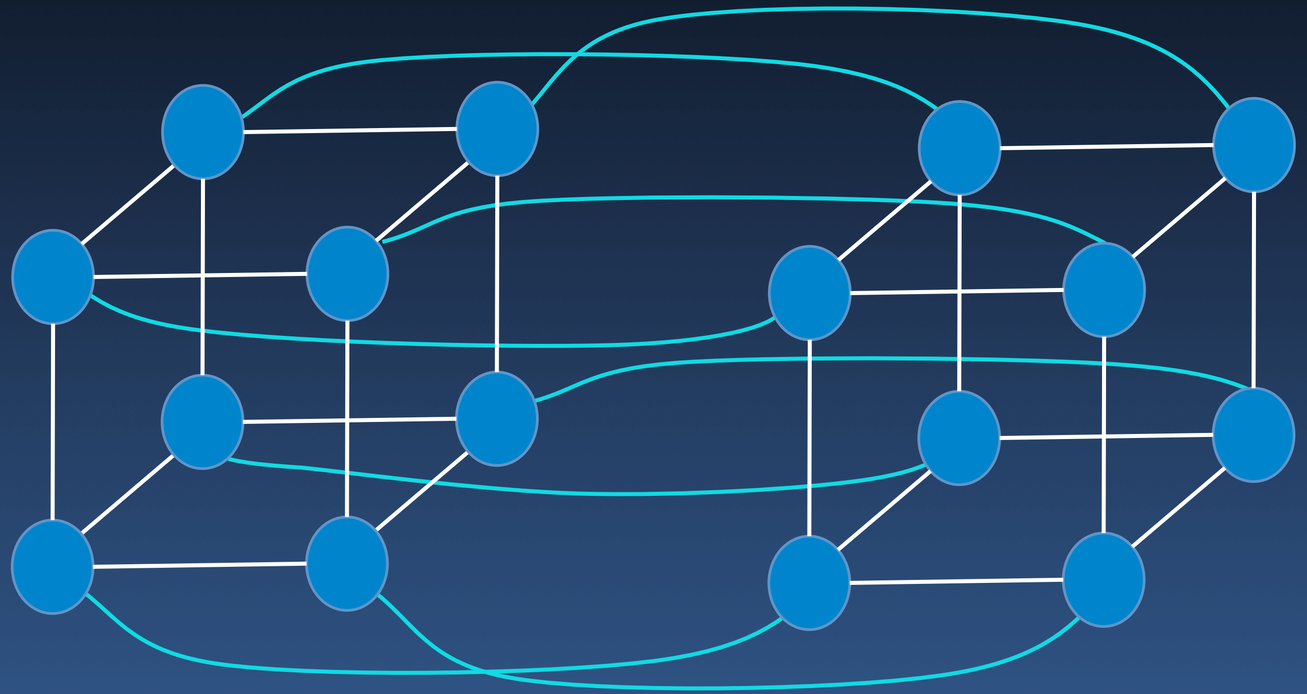
Hypercube



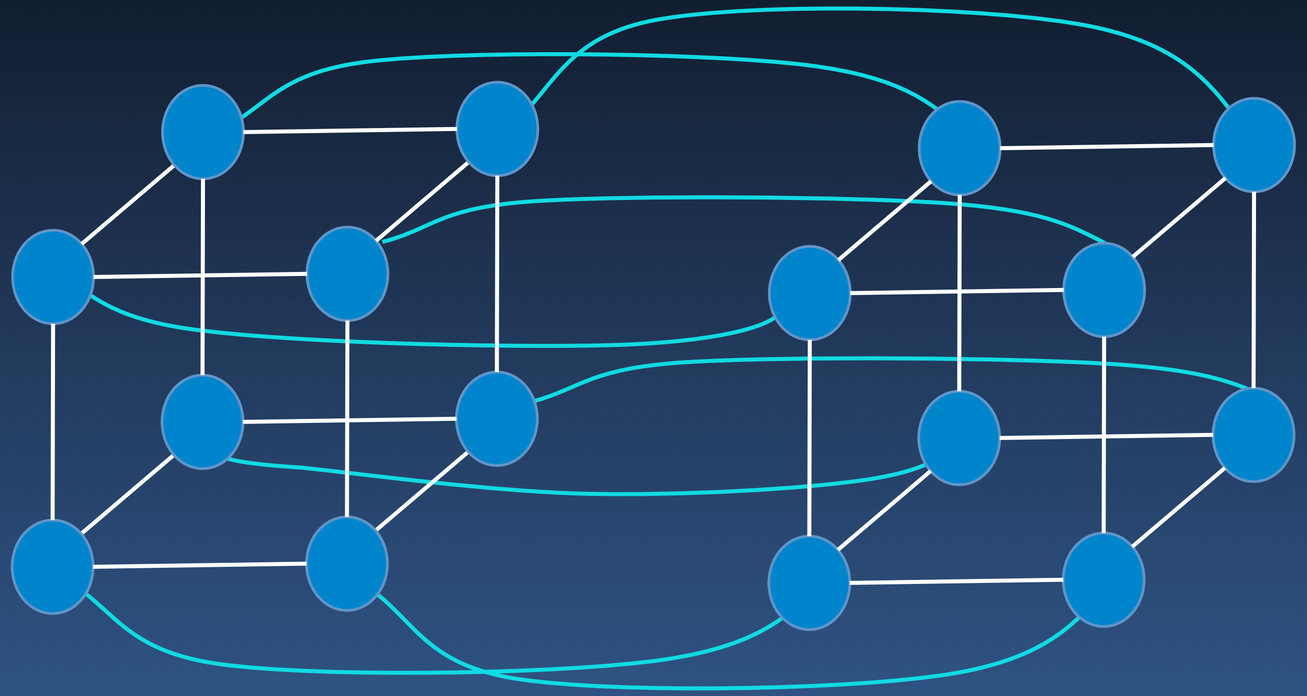
Hypercube



Hypercube

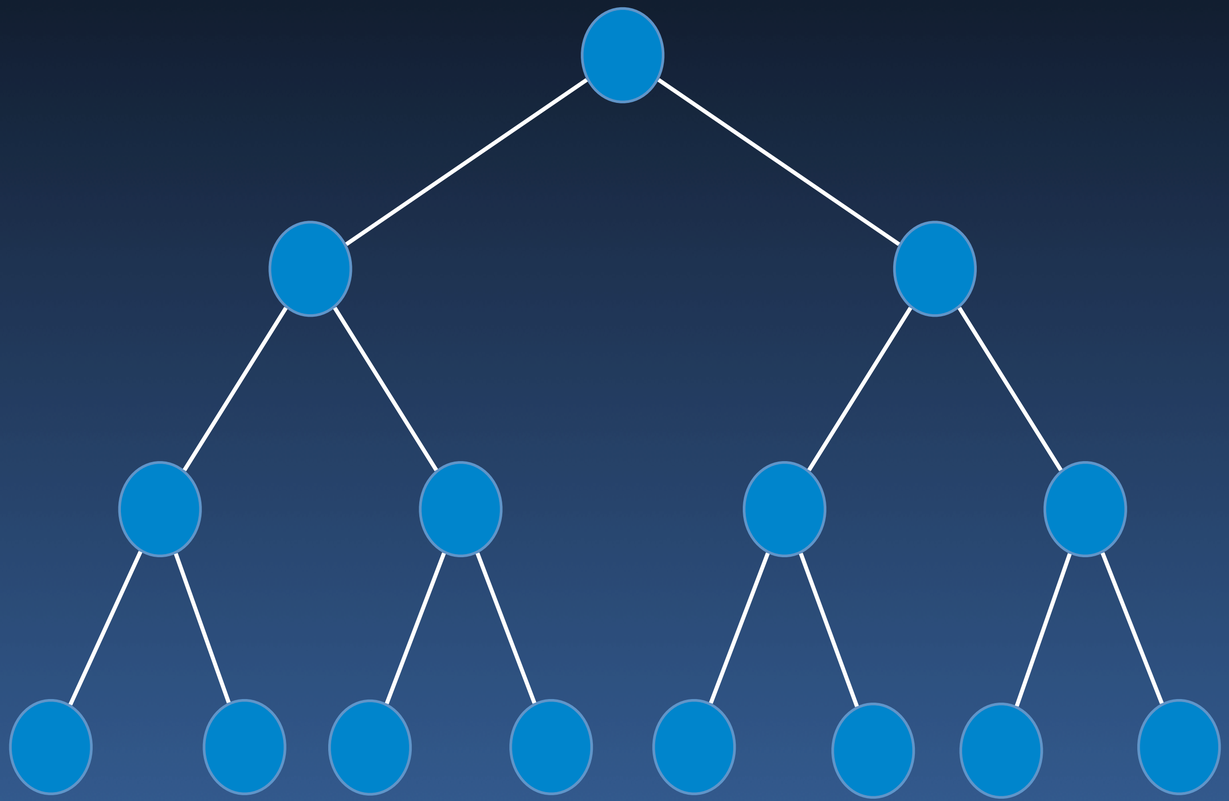


Hypercube

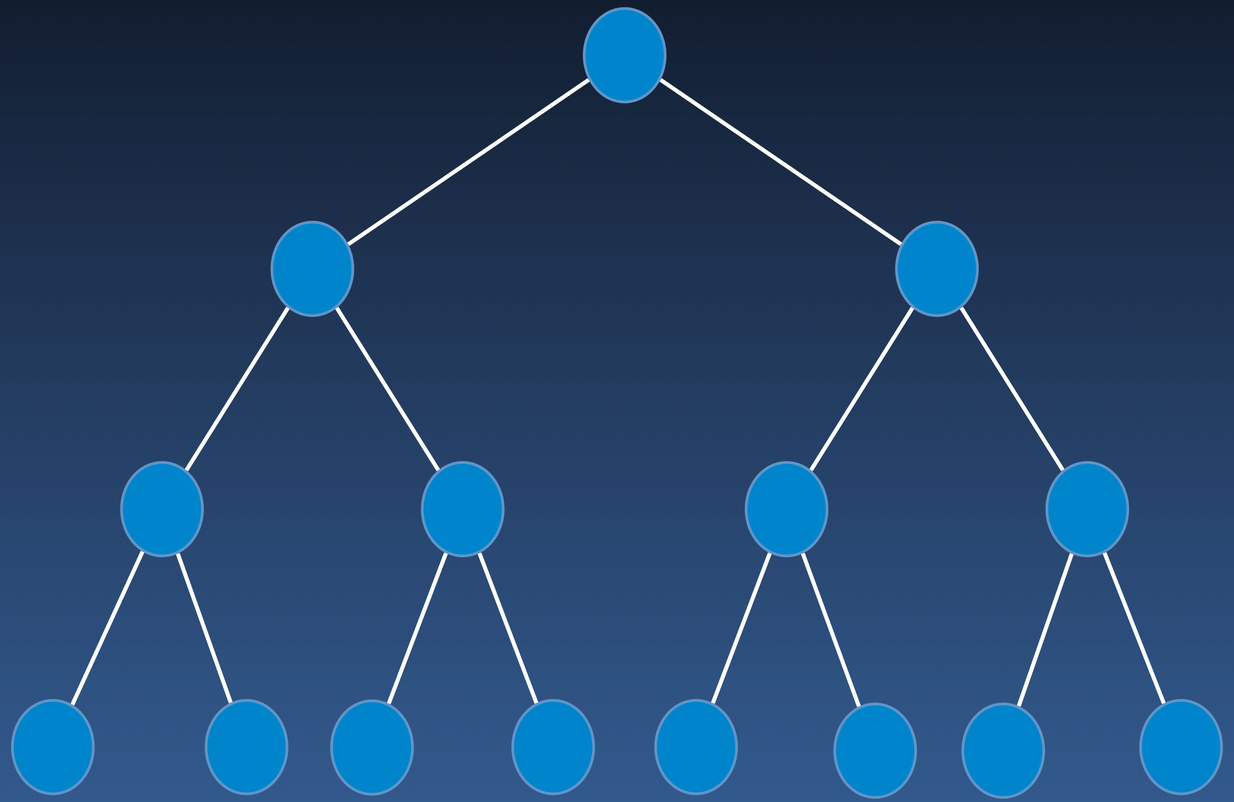


No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Tree Network

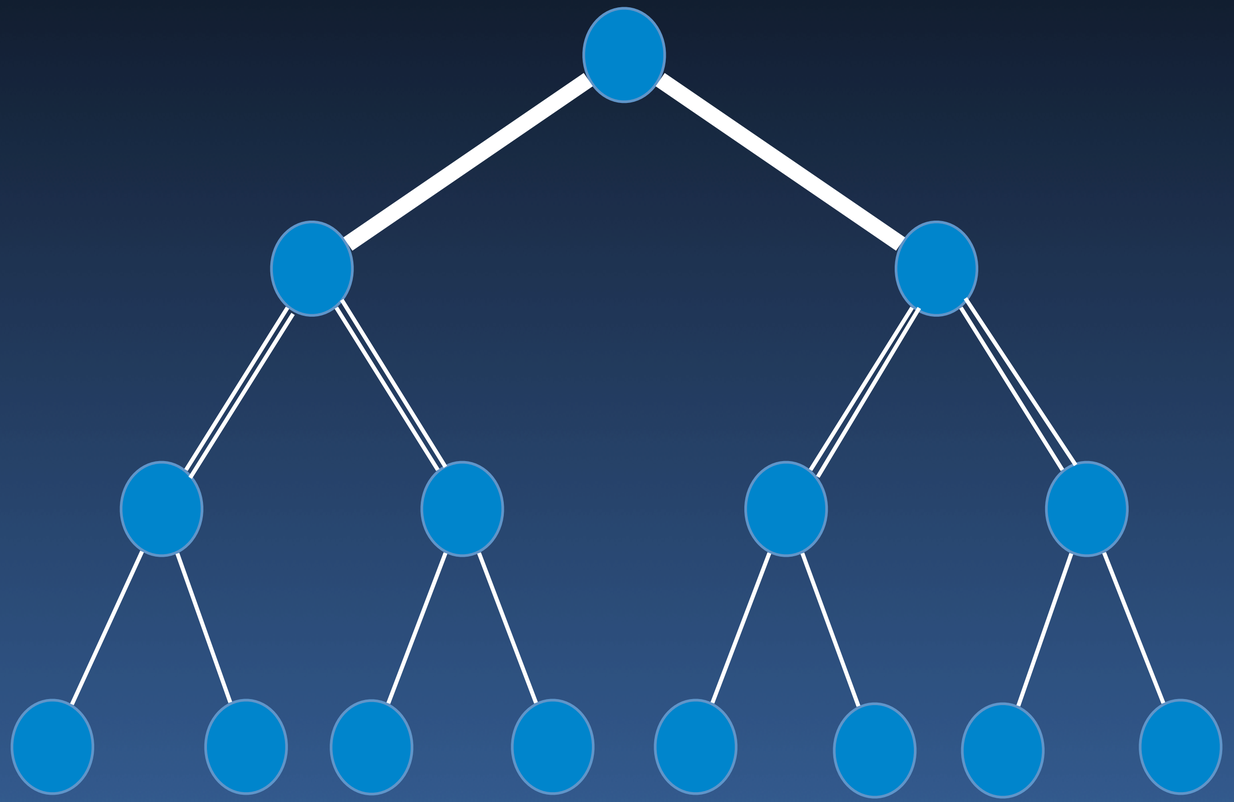


Tree Network



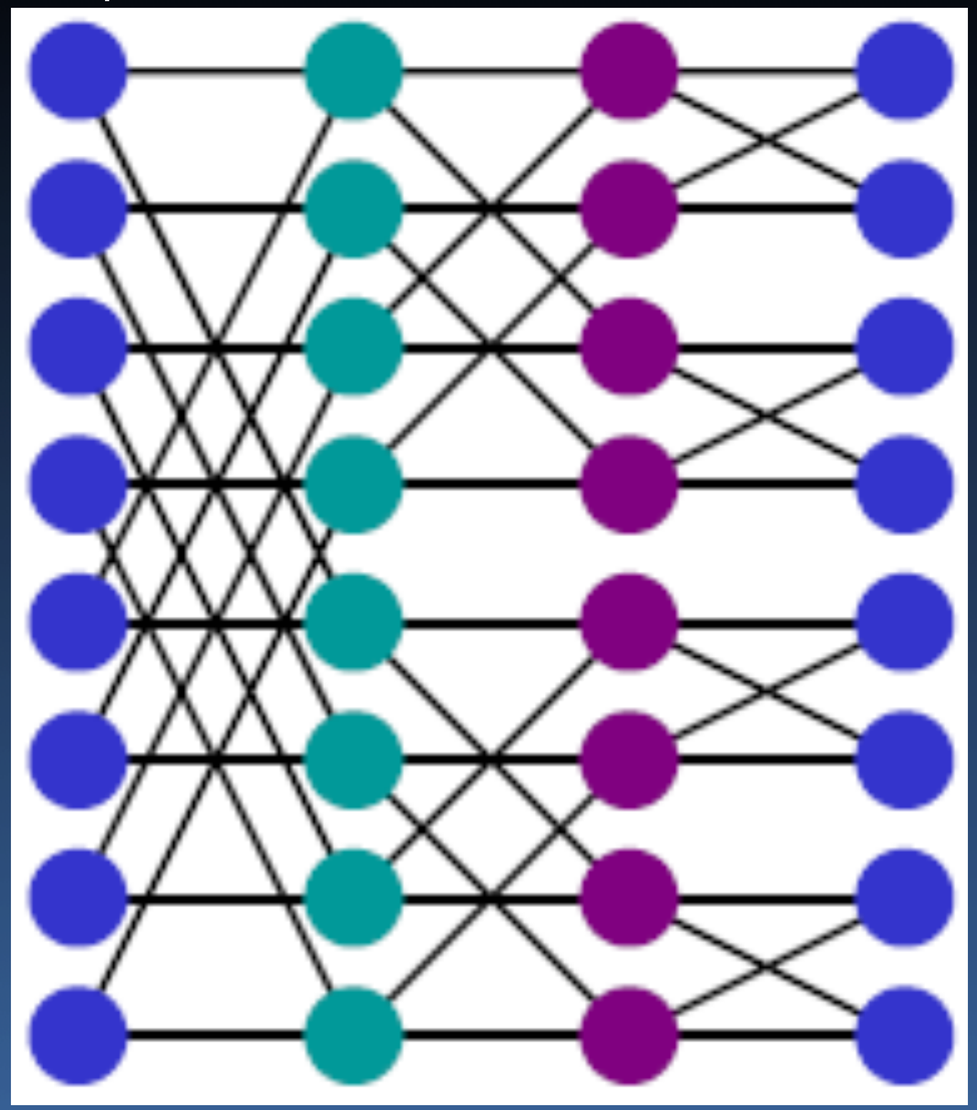
No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

Fat Tree Network



No. of links = ?
Diameter =?
Bisection width = ?
Blocking?

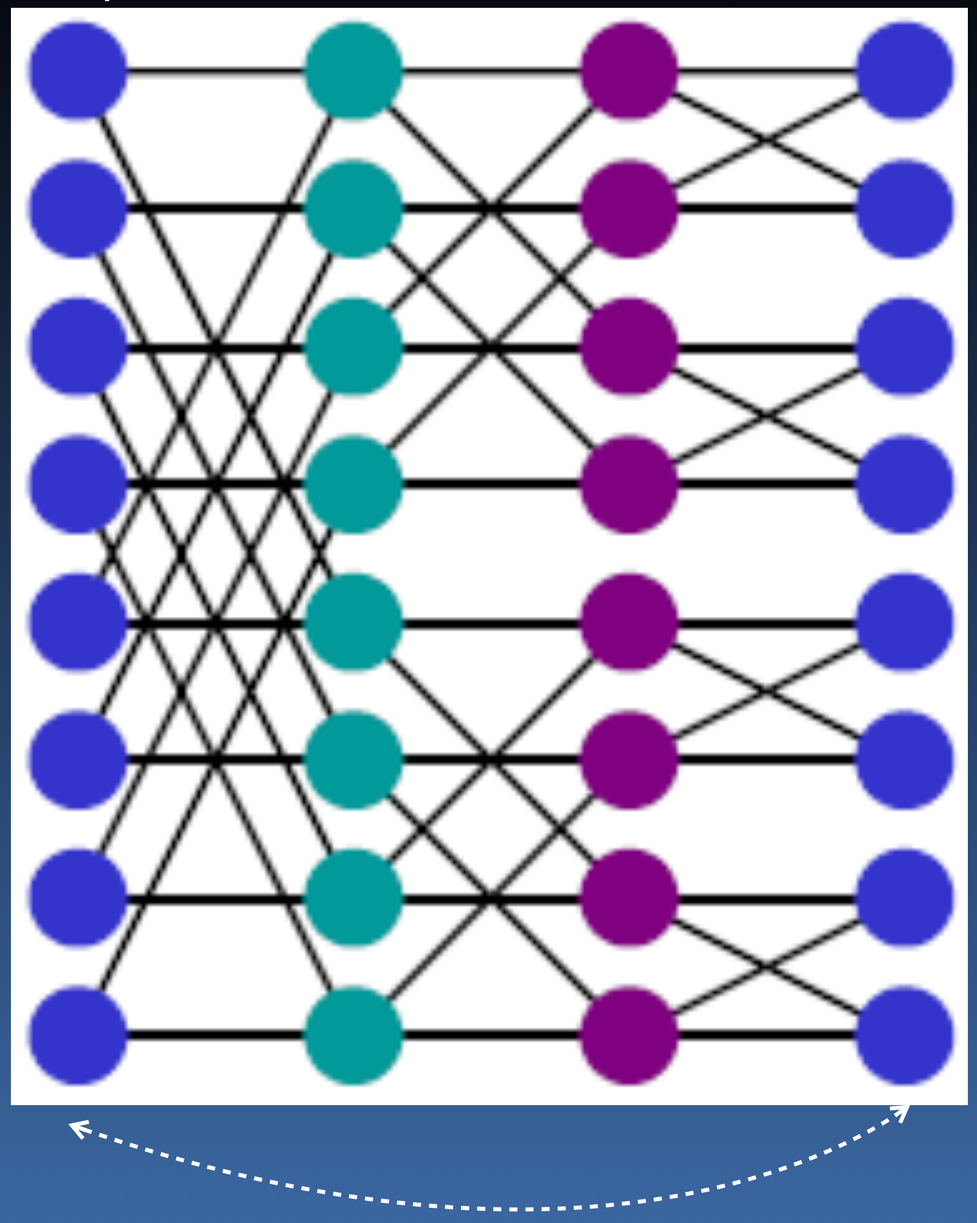
Look this up..



Butterfly

Multi-stage Network

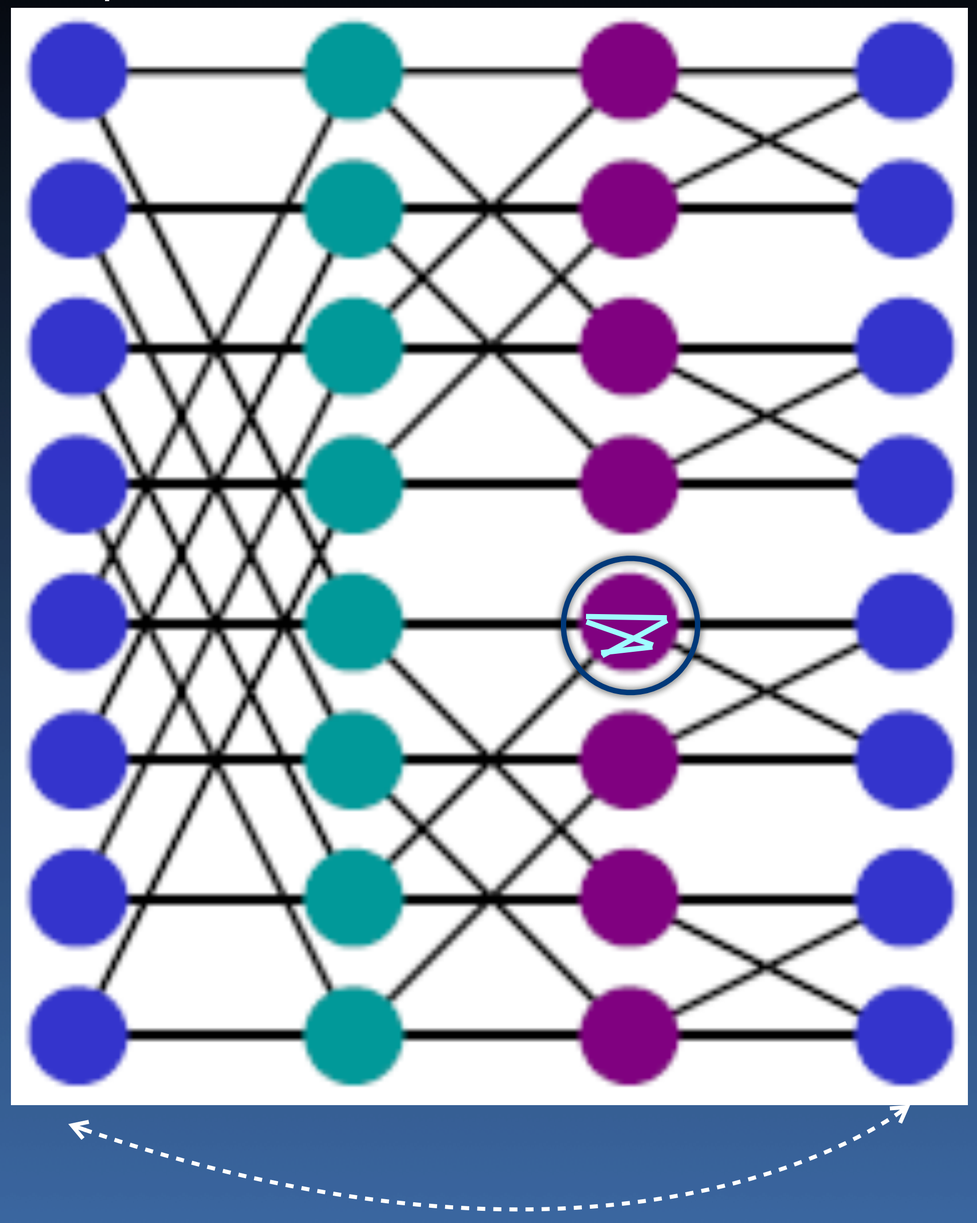
Look this up..



Butterfly

Multi-stage Network

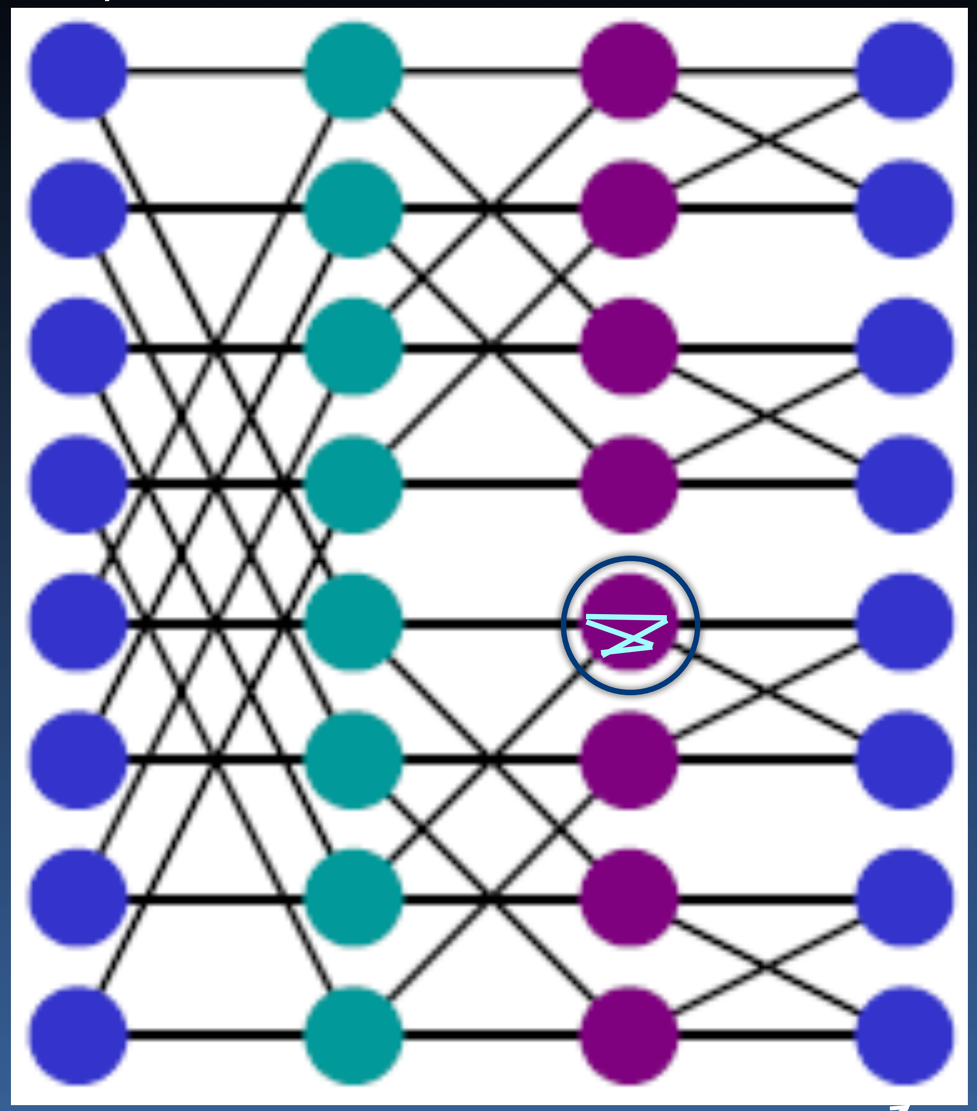
Look this up..



Butterfly

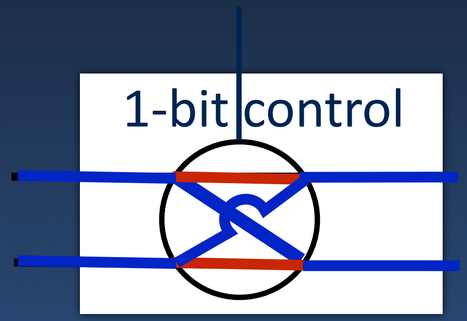
Multi-stage Network

Look this up..



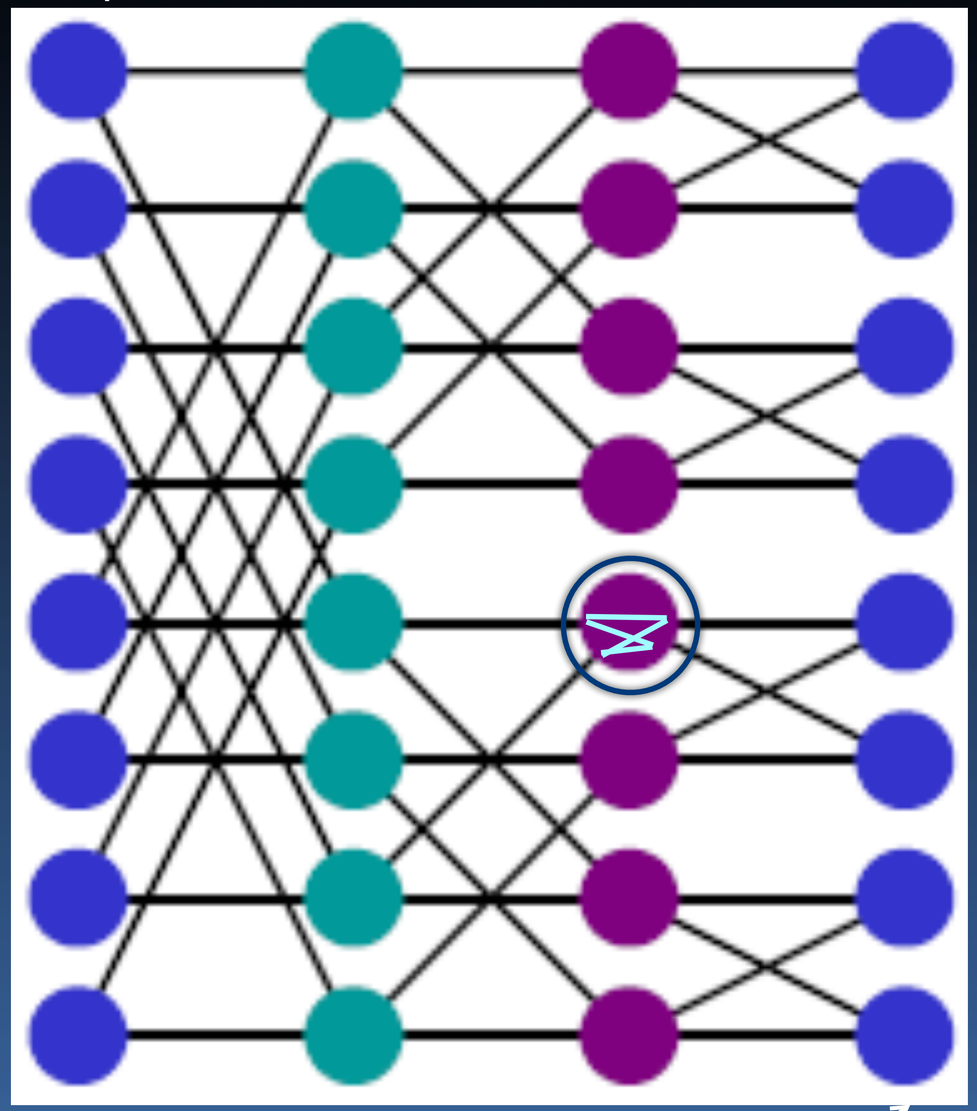
Butterfly

Multi-stage Network



Pass-through or
Crossover

Look this up..



Butterfly

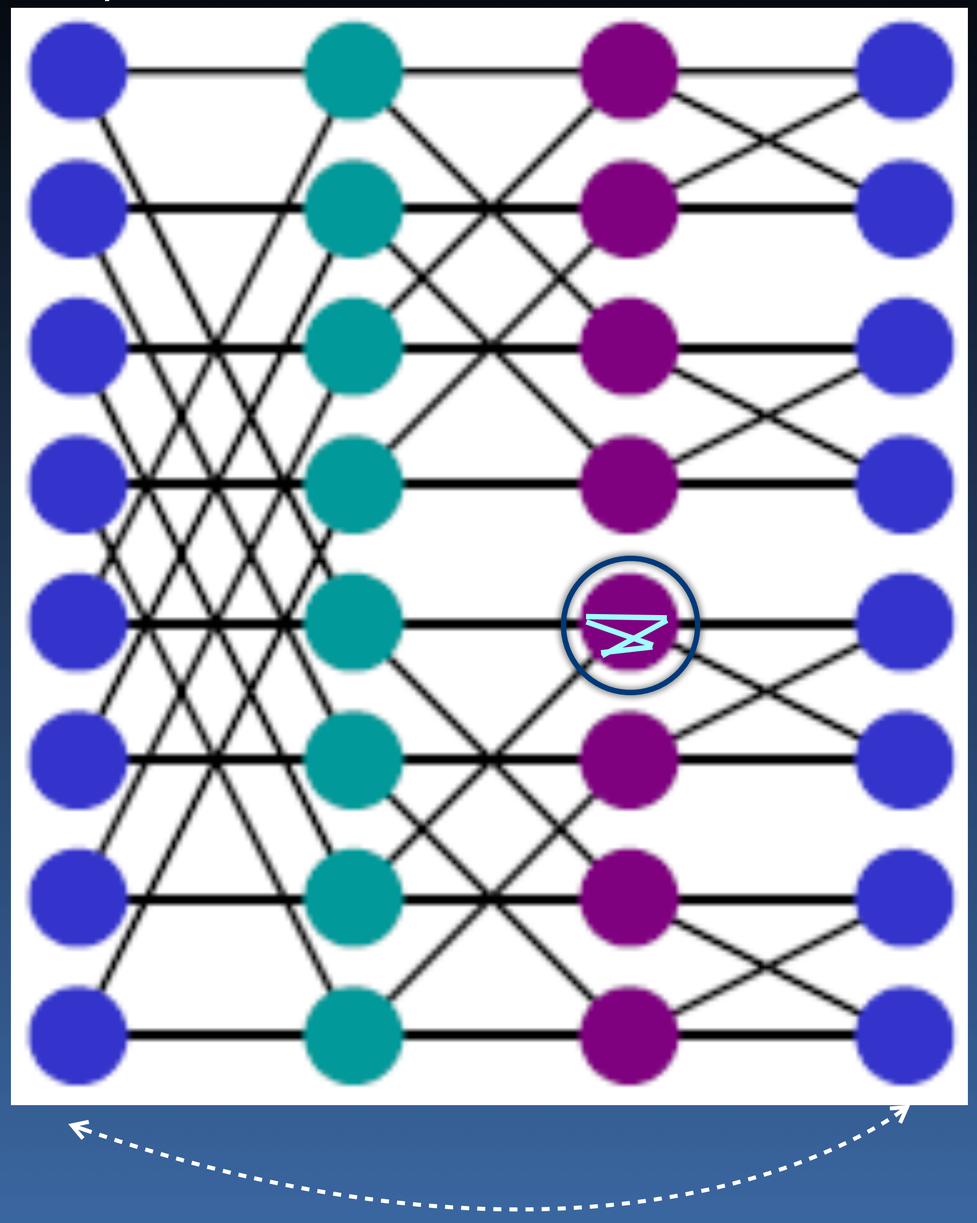
Multi-stage Network

Pass-through or
Crossover

Look this up..

Butterfly

Shuffle Exchange



Multi-stage Network

Pass-through or Crossover

Summary

- Instruction latency and overlap
- Core organization
- Inter-communication