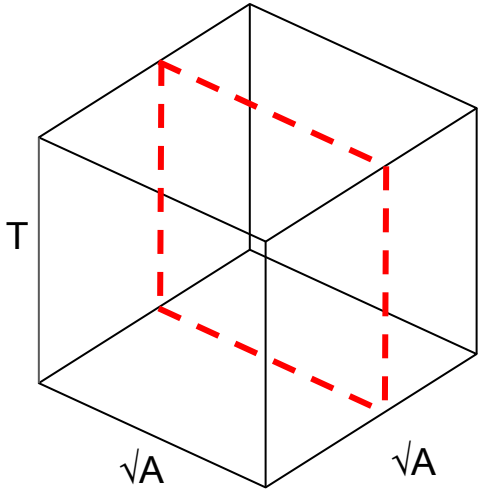
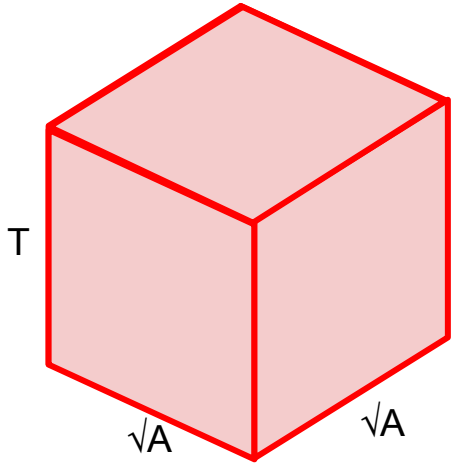
# Updates to VLSI theory

# Refresher on VLSI bounds

We have a few bounds, mostly from the 1980s, for implementation on chips:




1. All nodes must be mapped to some unique location and time: $AT \geq \Omega(N)$
2. Bisections of the volume induce balanced cuts: $A \geq \Omega(k)$ (*) and $\sqrt{(A)}T \geq \Omega(k)$ (therefore $AT^2 \geq \Omega(k^2)$)
3. (in some cases) it must be possible to communicate across the chip: $T \geq \sqrt{(A)}$

Combining (1) and (3): $T^3 \geq \Omega(N)$

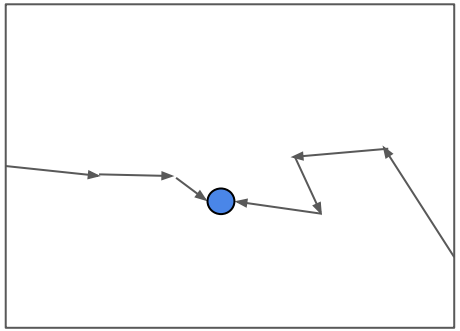
*: including memory-only area

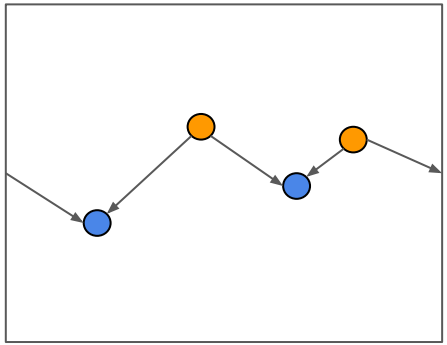
# First question: When do we have T ≥ √(A)?

Suppose A is "minimum bounding box area," i.e. there are values on all 4 borders of the chip

- Single-output computation: some path has length at least √(A)/2
- Generalization: A computation graph with "path diameter" d has some path with length at least √(A)/d

So for computations with path diameter d, we have T ≥ √(A)/d (assume inputs are not replicated off-chip)





# A bound on path diameter

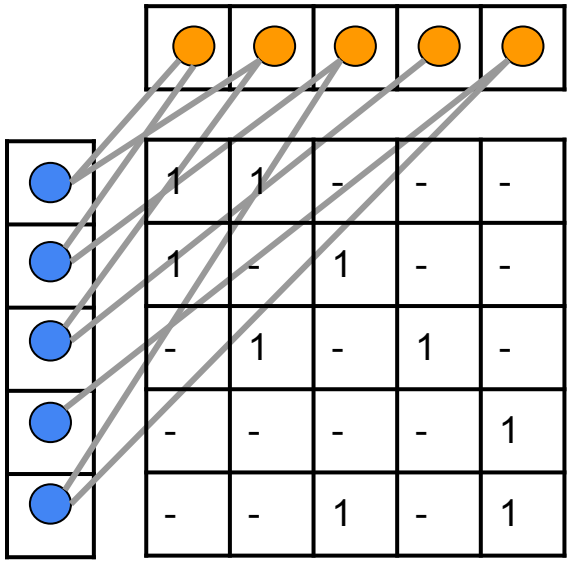
- Dense matmul has path diameter 6:

$$v_1 \to C_{ij} \leftarrow A_{i0} \to C_{in} \leftarrow B_{0n} \to C_{mn} \leftarrow v_2$$

- Combined with AT^2 ≥ Ω(n^4), this gives us T^4 ≥ Ω(n^4) for T ≥ Ω(n)
- Bound achievable even with all I/O on perimeter and n^3-style computation

# Another bound on path diameter

- Sparse matrix - dense vector multiplication (SpMV) has path diameter determined by the input matrix
  - This is actually the same communication structure as a single iteration of Bellman-Ford
- Equal to the diameter of the bipartite graph defined by the matrix
  - Has something to do with the diameter of the input graph
- This gives bounds which hold even if you know the graph far in advance and can do fancy layout:
  - $T \geq \Omega(\sqrt[3]{(n/d^2)})$
  - $T \geq \Omega(\sqrt{(b/d)})$ where b is the minimum bisection of the graph



# Notes on path-diameter-based bounds

- Any bound based on "there must exist a path of length…" is only a latency bound: time elapsed between first input and last output
  - Nothing to say the other paths didn't finish much earlier or start much later
  - So if we have to do k operations in a row, we can't just multiply the bound by k; they might be overlapped, even in the same area
  - The T in $\sqrt{(A)}T$ is throughput time, though
- Is the matmul bound useful?
  - We already kind of knew it
- Is the SpMV bound useful (say, for GNNs)?
  - "Maybe" -Alok Tripathy (paraphrased)
- Other thoughts?
  - We'd really like to have bounds on things other than latency – I have more on this

# Bounds on Total Communication

Suppose we could show something like:

For any layout of the computation on a chip, at least k values must be communicated between the left and right thirds of the chip

Since the distance these values must cross is $\sqrt{(A)}/3$, we have that the "total communication distance" is at least $k\sqrt{(A)}/3$

This is a lower bound on "total work" and thus energy

Energy $\geq \Omega(k\sqrt{(A)})$

$\sqrt{(A)}T \geq \Omega(k)$ as usual, so $ET \geq \Omega(k^2)$

k=n/3 for load-balanced, oblivious SpMV