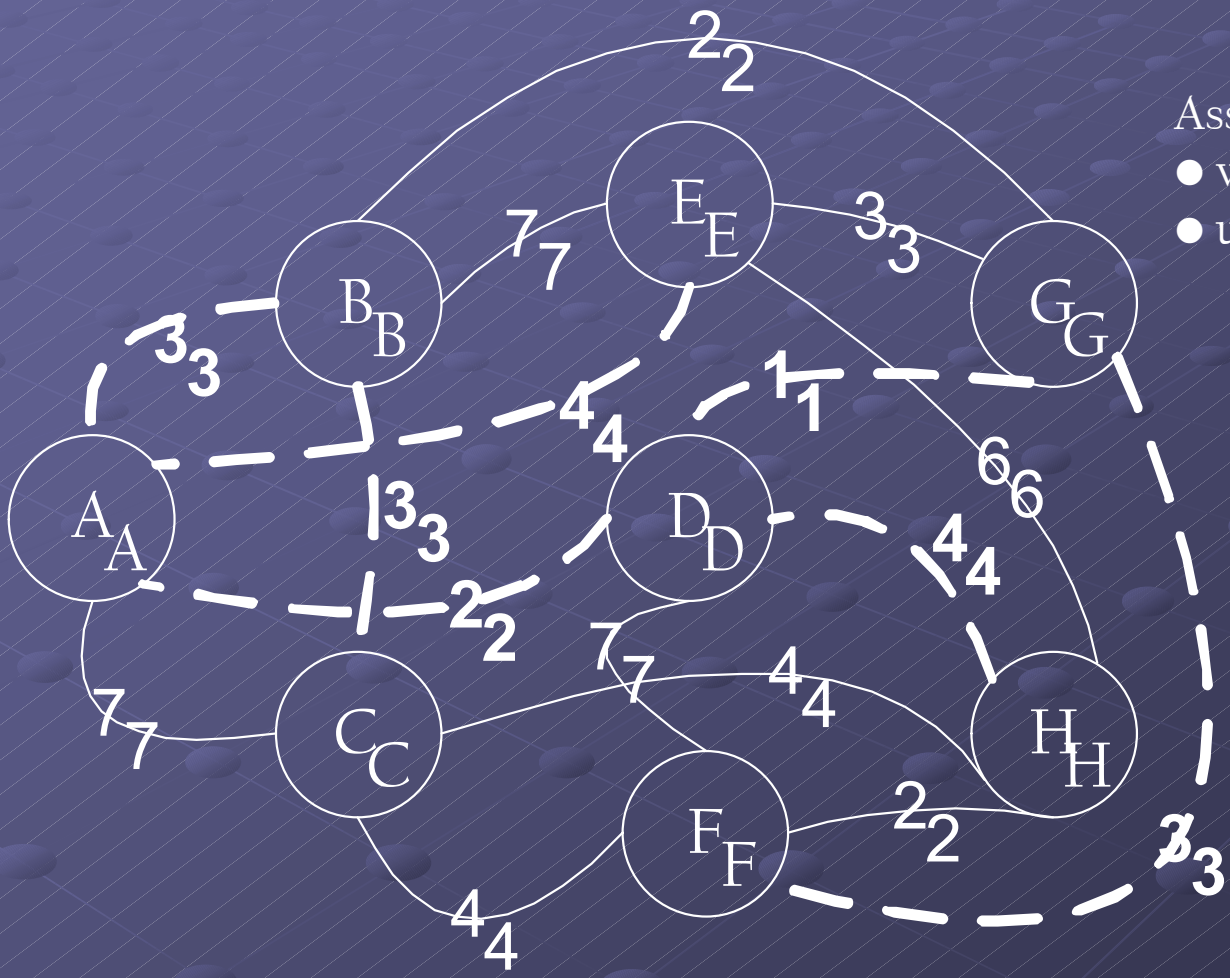
SPED

shortest path evaluation device

Blake Borgeson, Jacob Rios, Moffett Stephen, Angel Sun

Shortest Path Problem

- Given a network of nodes, find the shortest distance from A to all other nodes



Assumptions:

- weights range from 1 to 7
- up to 4 connections per node

Dijkstra's Algorithm

```
T = {A}
for all vertices
        if v is adjacent to A
                D(v) = w(A,v)
        else    D(v) = ∞

Find u not in T such that D(u) is a minimum
Add u to T
for v not in T and v adjacent to u
        D(v) = min[D(v), D(u) + w(u,v)]
```

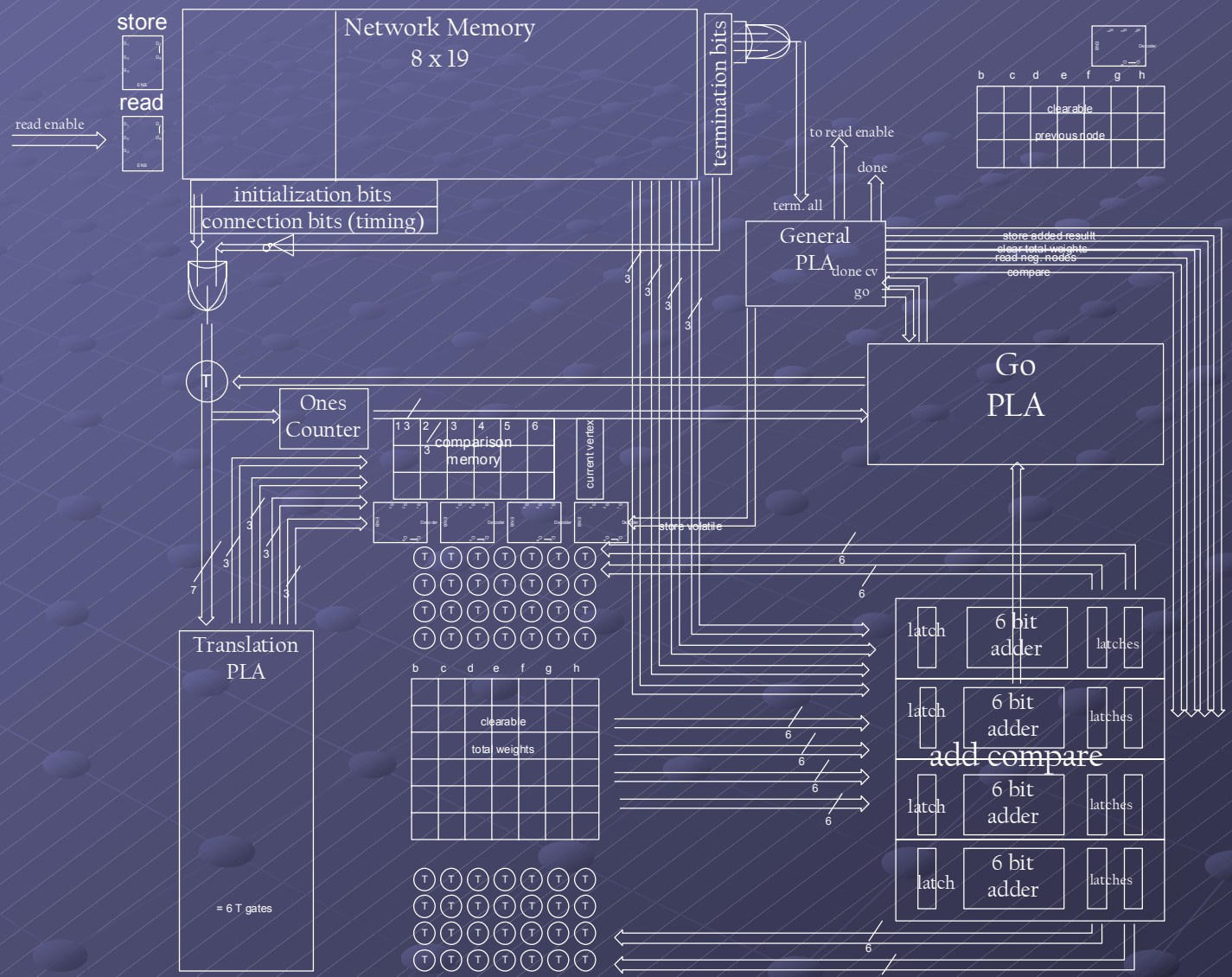
T = termination set

D(v) = current shortest distance to v

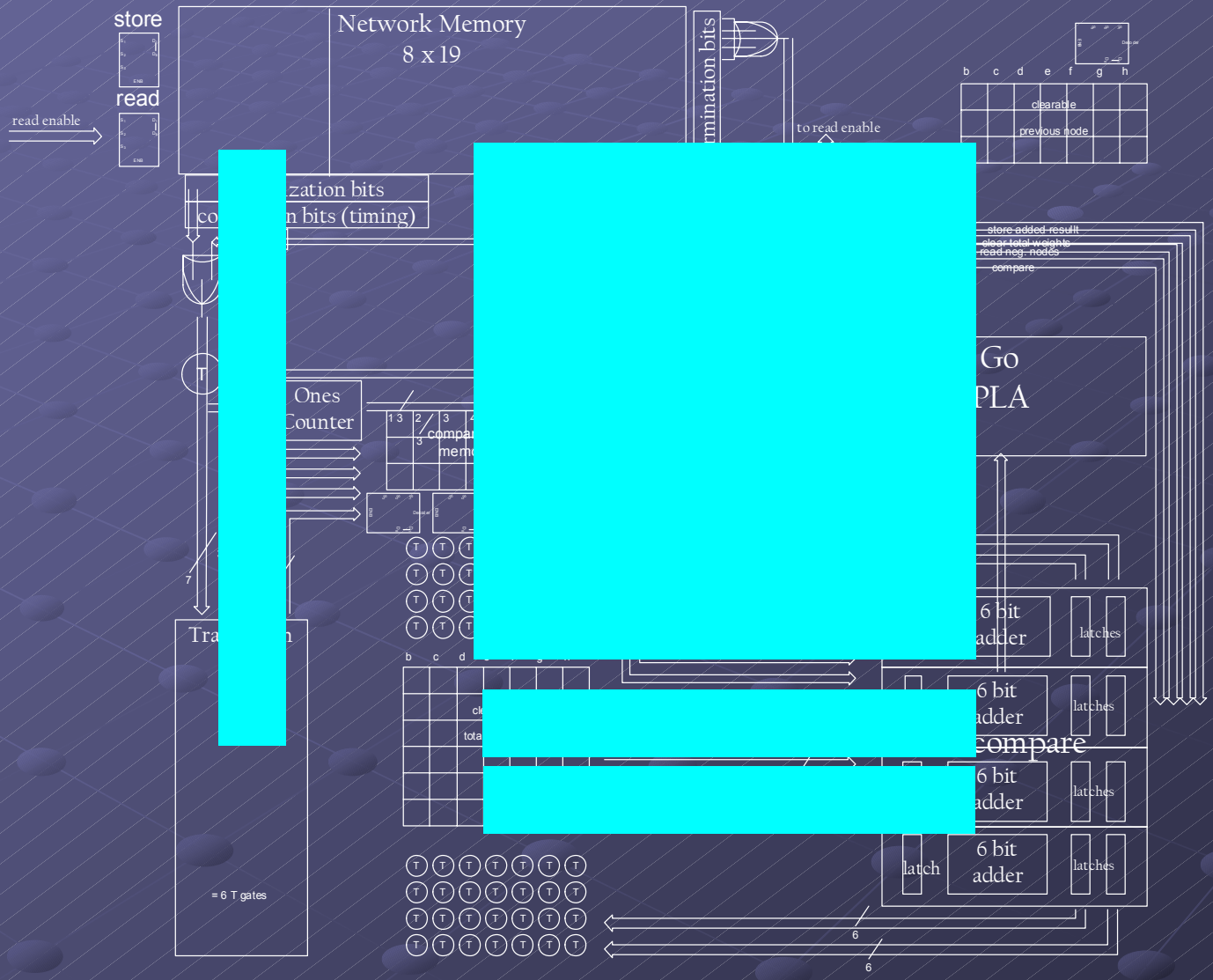
w(i,j) = weight of edge connecting i and j

* update D(v) if the path to v via u is shorter than the previous shortest path.

High Level Block Diagram



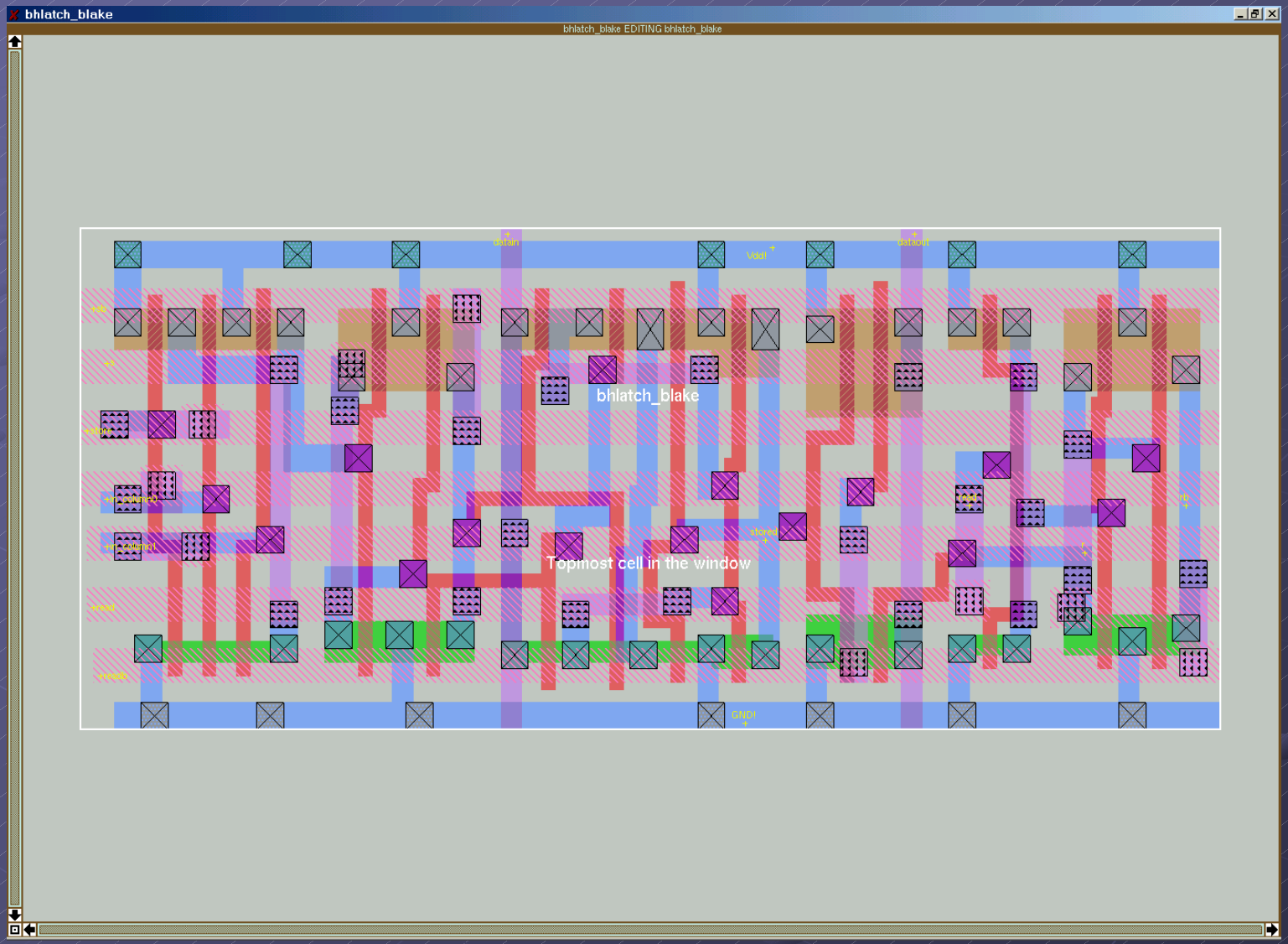
Simplified Data Flow



Network Memory

- 8 rows each consisting of 19 bits of storage
- Each row containing all information for one vertex
- First 7 bits storing connections
- Last 12 bits storing weights

Network Memory – Buffer/Latch



Total Weight Memory

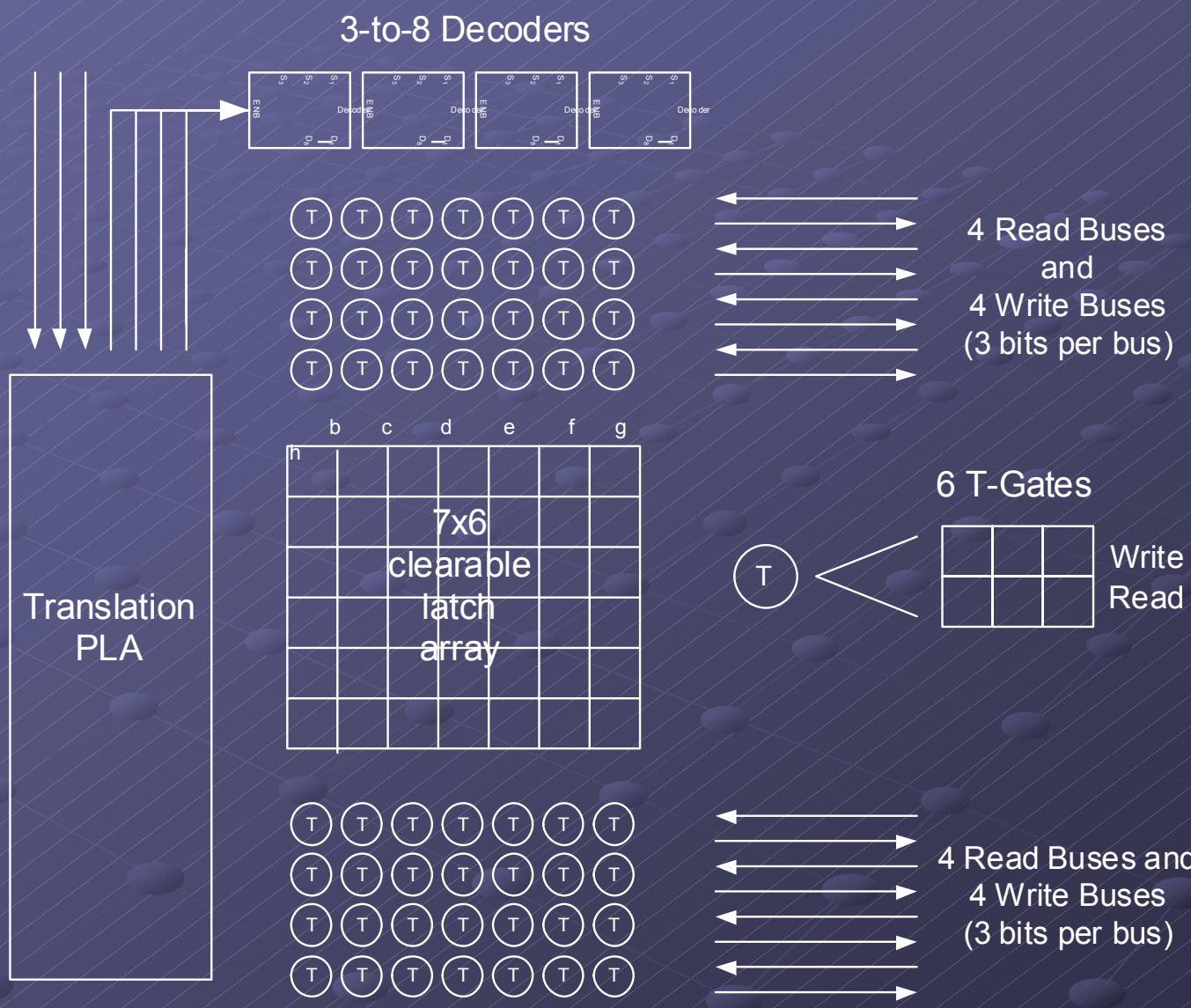
Function : Stores the current shortest distance D(v) for each vertex.

Problem : Cell takes in four 6-bit buses simultaneously and must be able to read and store to any combination of the seven columns.

Solution :

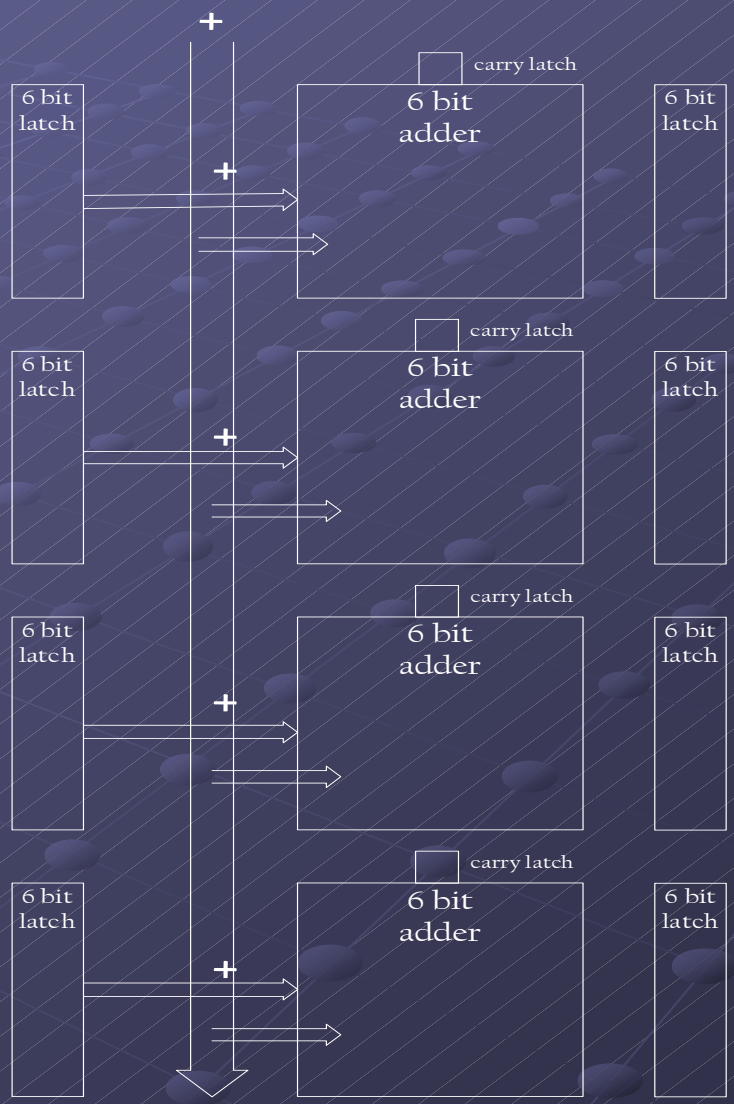
- Array of T-gates
- Four 3 to 8 decoders
- Translation PLA

Total Weight Memory



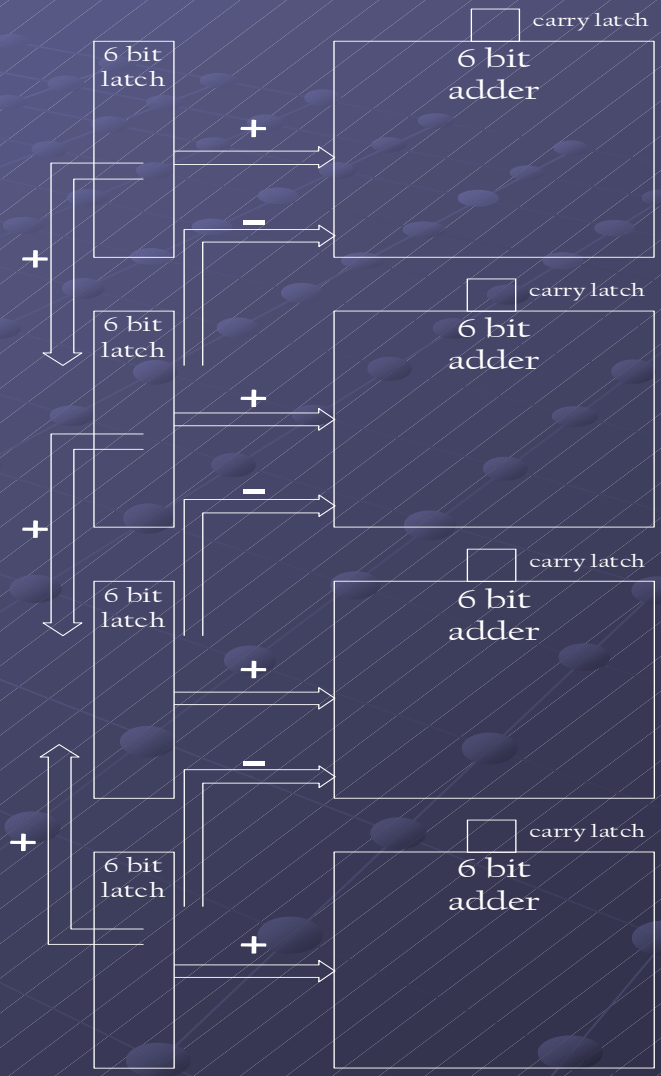
Add/Compare

- Chip's main logic unit
- Under General PLA's control

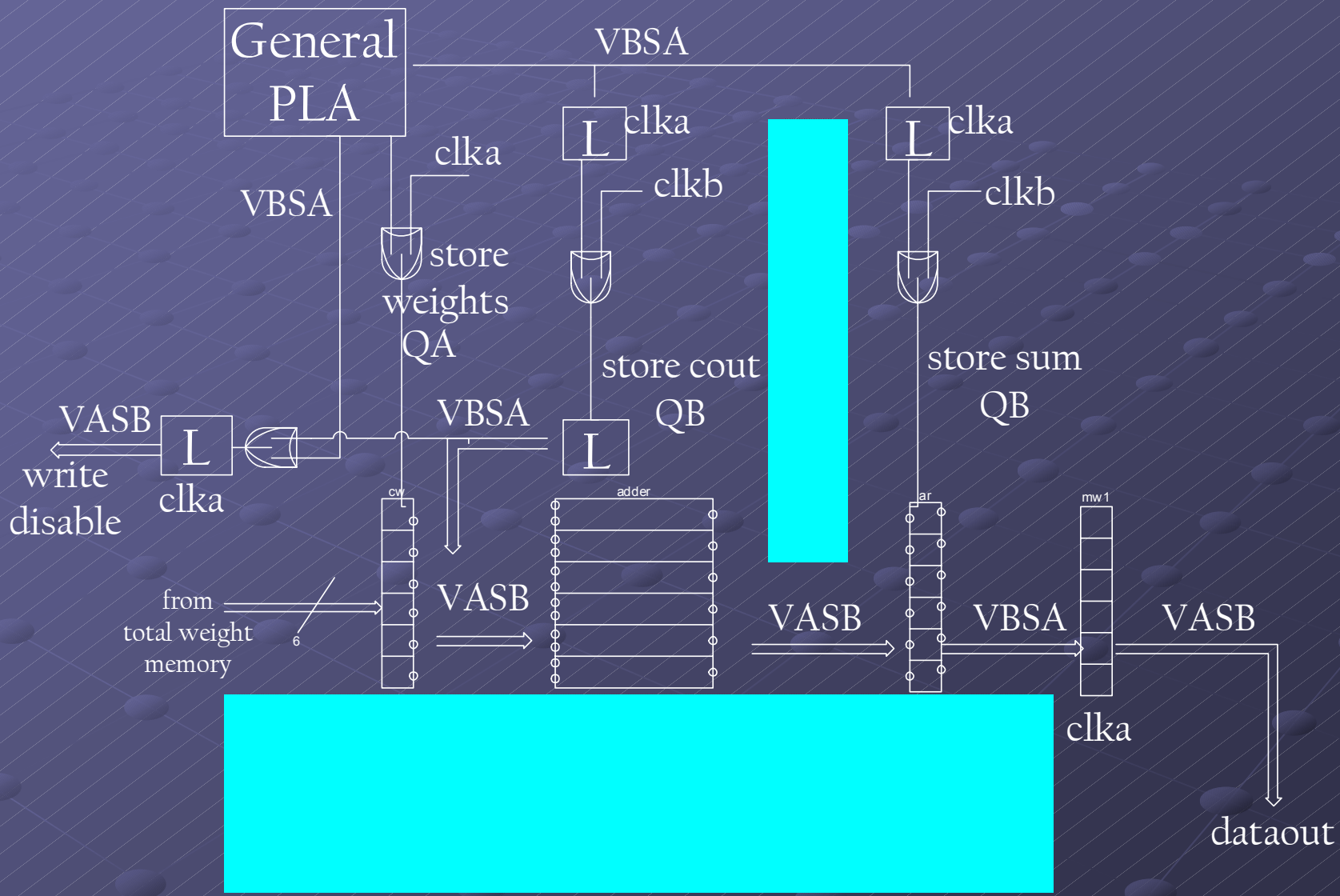


Determining Next Vertex

- 2 to 6-way comparison
- Takes advantage of rarity of having many active nodes
- Run by Go PLA

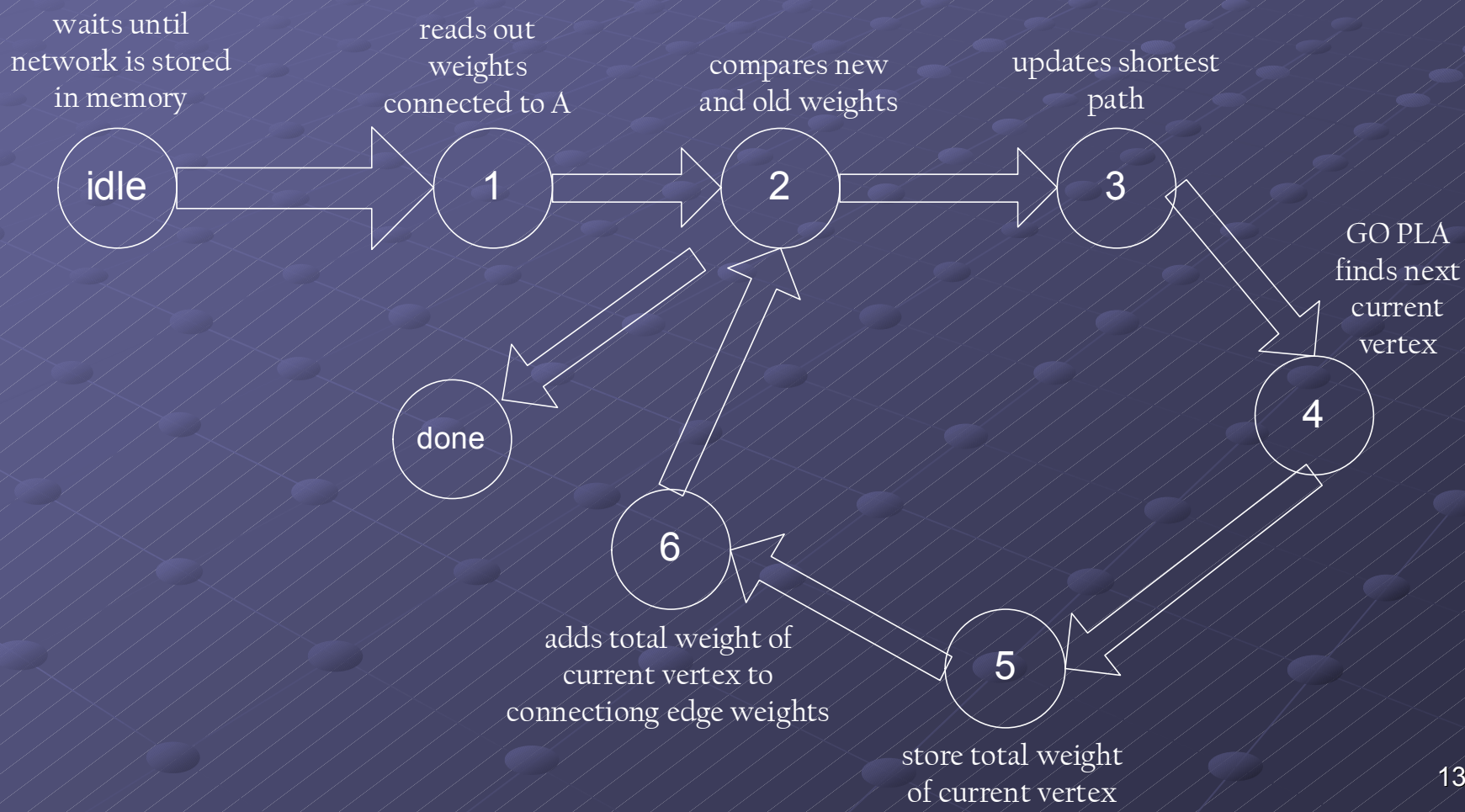


Two Phase Timing



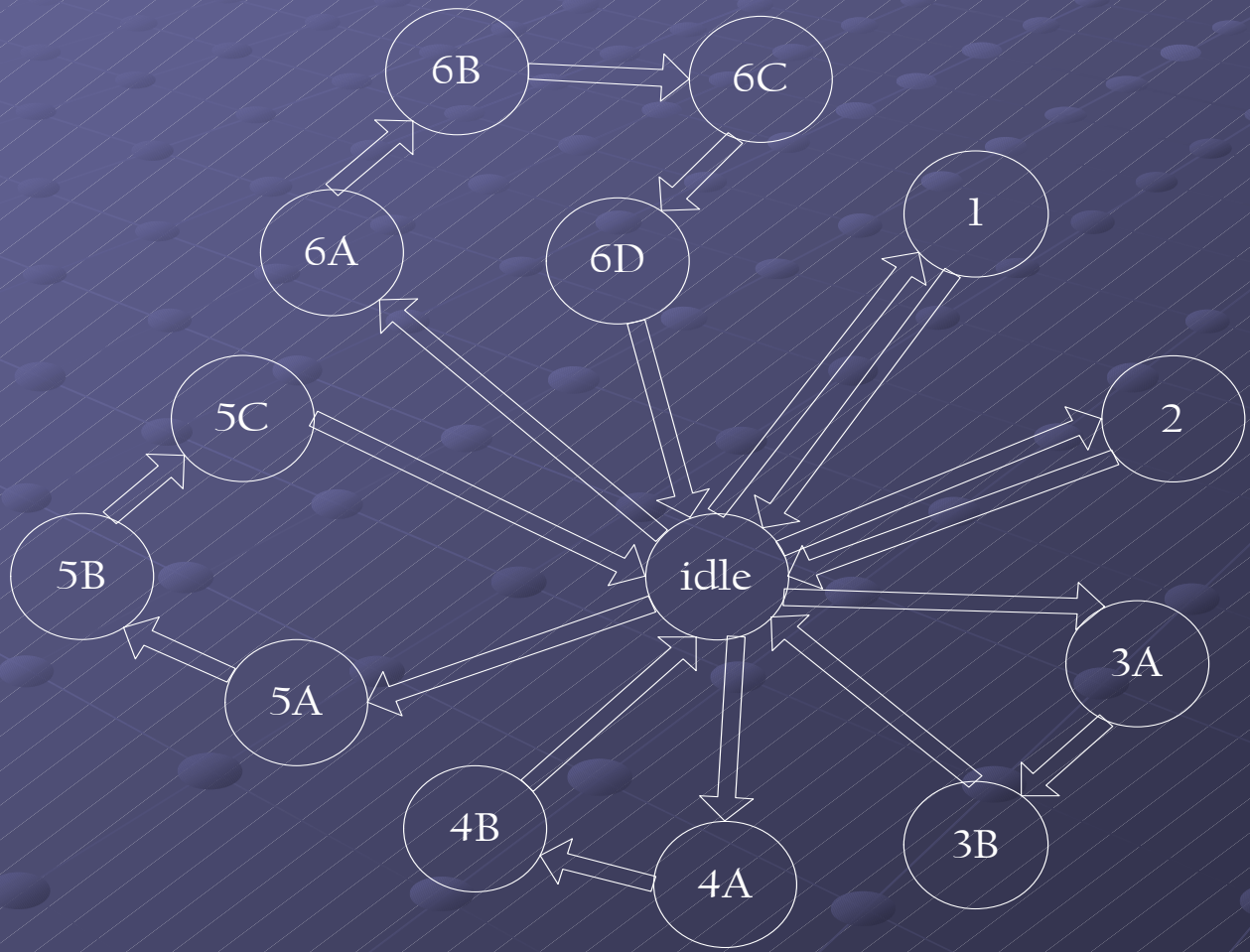
General PLA

- Compares weights of newly discovered paths to current path
- Updates with new path if shorter



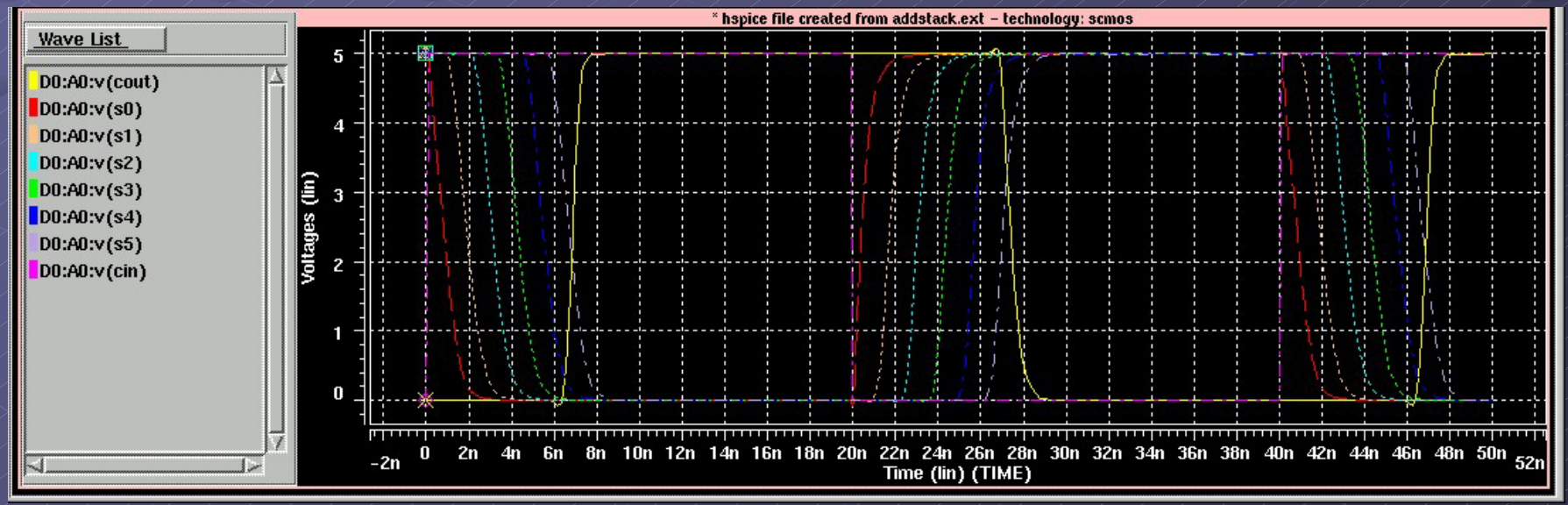
GO PLA

- Finds the next current vertex.
- Performs up to a 6 way comparison using the four 6-bit adders.

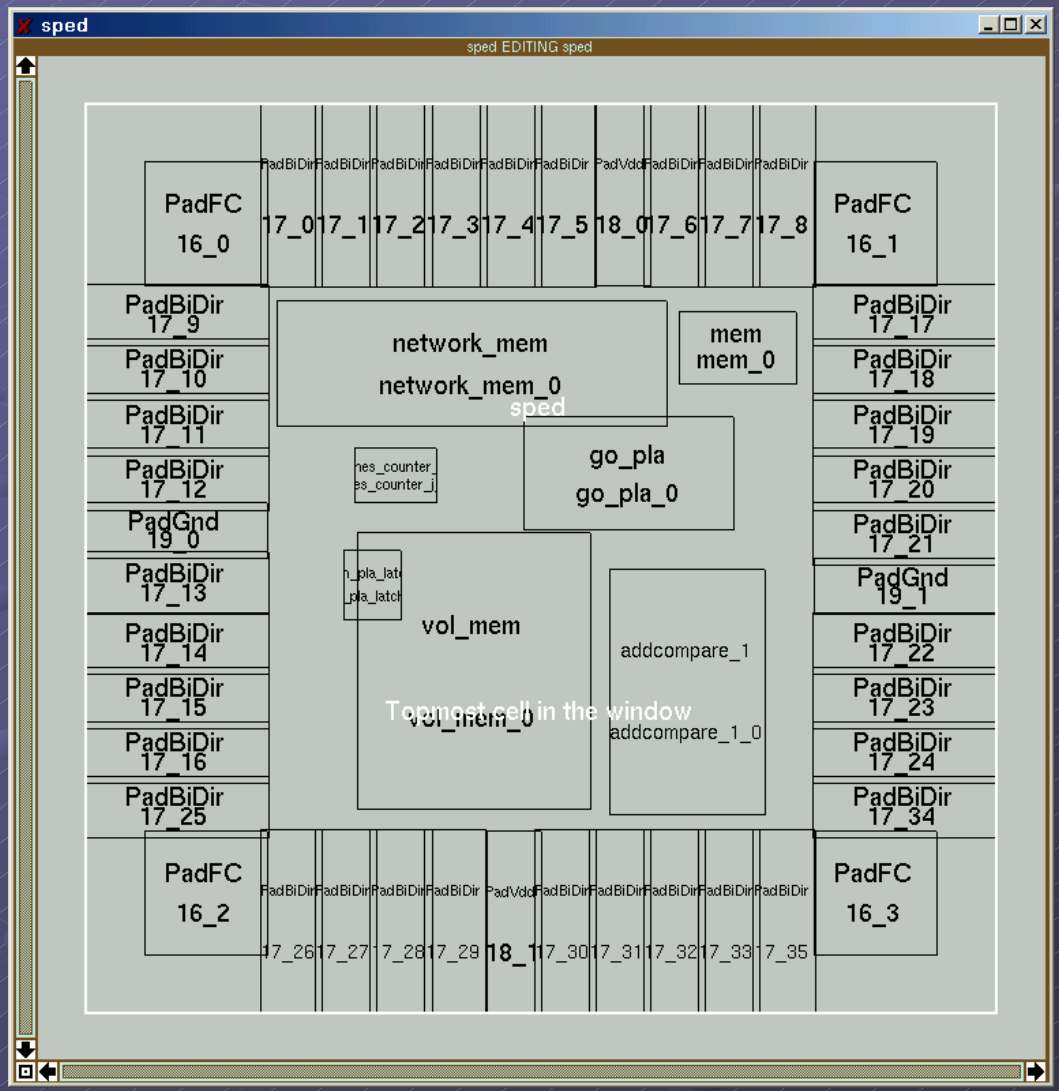


Critical Path

- 6 bit ripple carry adder in add compare unit
- Longest delay ~ 8.4 ns
- Estimated clock frequency ~ 30 MHz



Chip Layout


sped
sped EDITING sped
PadFC 16_0
PadBiDir 17_0
PadBiDir 17_1
PadBiDir 17_2
PadBiDir 17_3
PadBiDir 17_4
PadBiDir 17_5
PadVdd 18_0
PadBiDir 17_6
PadBiDir 17_7
PadBiDir 17_8
PadFC 16_1
PadBiDir 17_9
PadBiDir 17_10
PadBiDir 17_11
PadBiDir 17_12
PadGnd 19_0
PadBiDir 17_13
PadBiDir 17_14
PadBiDir 17_15
PadBiDir 17_16
PadBiDir 17_25
network_mem
network_mem_0
mem
mem_0
sped
go_pla
go_pla_0
vol_mem
addcompare_1
addcompare_1_0
Topmost cell in the window
PadBiDir 17_17
PadBiDir 17_18
PadBiDir 17_19
PadBiDir 17_20
PadBiDir 17_21
PadGnd 19_1
PadBiDir 17_22
PadBiDir 17_23
PadBiDir 17_24
PadBiDir 17_34
PadFC 16_2
PadBiDir 17_26
PadBiDir 17_27
PadBiDir 17_28
PadBiDir 17_29
PadVdd 18_1
PadBiDir 17_30
PadBiDir 17_31
PadBiDir 17_32
PadBiDir 17_33
PadBiDir 17_35
PadFC 16_3

Conclusion

- Design Challenges

Parallel operation –need to read out and operate on 4 sets of values

- Applicability

Computer Network – shortest path in which to transfer data

- Scalable

Easily handles increase in problem complexity