# A 40×40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

# Luke R. Everson, Sachin S. Sapatnekar, Chris H. Kim University of Minnesota, Minneapolis, MN

© 2019 IEEE **International Solid-State Circuits Conference** 



# - Vertex Details - Edge Details Applications Conclusion

© 2019 IEEE **International Solid-State Circuits Conference** 

# Outline

- Background: Path Planning Algorithms Time-Based A\* ASIC
  - Time-Based Primer
  - Top Level Design
- 65nm Test Chip Results





# • Formal definition:

© 2019 IEEE **International Solid-State Circuits Conference** 



- Graph: set of "objects" and their "connections"
  - $-G = (V, E), V = \{v_1, v_2, ..., v_n\}, E = \{e_1, e_2, ..., e_m\}$
  - V: set of vertices (nodes), E: set of edges (links, arcs)
  - Directed graph:  $e_k = (v_i, v_i)$
  - Undirected graph:  $e_k = \{v_i, v_i\}$
  - Weighted graph: w:  $E \rightarrow R$ , w(e<sub>k</sub>) is the "weight" of e<sub>k</sub>.

## **Network Analysis**



## [Wikipedia.org]

© 2019 IEEE International Solid-State Circuits Conference

# Why Graphs? **Routing/Path Planning**

## [yuchsia.github.io]

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control



## Self-Driving Cars



## [techrepublic.com]

4 of 62

# **Graph Algorithms- Breadth First Search** • Explores search area equally in all directions Adds new neighbors to queue and visits in order added Simple to implement in software



![](_page_4_Figure_9.jpeg)

![](_page_4_Picture_10.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • **Visited** = {**S**} • Queue = {A,D}

© 2019 IEEE International Solid-State Circuits Conference

![](_page_5_Picture_3.jpeg)

![](_page_5_Figure_4.jpeg)

![](_page_5_Picture_5.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A} • **Queue = {D,B}**

© 2019 IEEE International Solid-State Circuits Conference

![](_page_6_Picture_3.jpeg)

![](_page_6_Picture_4.jpeg)

![](_page_6_Picture_5.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A,D} • **Queue = {B,E}**

© 2019 IEEE International Solid-State Circuits Conference

![](_page_7_Picture_3.jpeg)

![](_page_7_Picture_4.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A,D,B} • **Queue = {E,C}**

© 2019 IEEE International Solid-State Circuits Conference

![](_page_8_Picture_3.jpeg)

![](_page_8_Picture_4.jpeg)

![](_page_8_Picture_5.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A,D,B,E} • **Queue = {C,T}**

© 2019 IEEE **International Solid-State Circuits Conference**  2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_9_Picture_3.jpeg)

![](_page_9_Picture_4.jpeg)

![](_page_9_Picture_5.jpeg)

10 of 62

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A,D,B,E,C} • Queue = $\{T\}$

© 2019 IEEE International Solid-State Circuits Conference

![](_page_10_Picture_3.jpeg)

![](_page_10_Picture_4.jpeg)

# **Graph Algorithms- Breadth First Search** Explore entire graph from Source (S) • Visited = {S,A,D,B,E,C,T} • **Queue** = {}

© 2019 IEEE **International Solid-State Circuits Conference** 

![](_page_11_Picture_3.jpeg)

![](_page_11_Picture_4.jpeg)

![](_page_11_Picture_5.jpeg)

• Similar to BFS when edges have weights neighbor first Keeps track of

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's Dijkstra's** BFS Uniform cost search

# Search "cheapest" distance FROM start

![](_page_12_Figure_7.jpeg)

![](_page_12_Figure_8.jpeg)

![](_page_12_Figure_9.jpeg)

# Find shortest path from Source (S) to Target (T) • Visited= $\{S_n\}$ • Cost=0• **PriorityQueue=** $\{A_{s_1}, D_{s_1}\}$

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's**

![](_page_13_Picture_5.jpeg)

## **Store Parent Node for** traceback

![](_page_13_Figure_8.jpeg)

# Find shortest path from Source (S) to Target (T) • Visited= $\{S_0, A_{S_1}\}$ • Cost=1 • **PriorityQueue=** $\{D_{S1}, B_{A2}\}$

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's**

![](_page_14_Figure_5.jpeg)

# Find shortest path from Source (S) to Target (T) • Visited= $\{S_0, A_{S_1}, D_{S_1}\}$ • Cost=1 • **PriorityQueue=**{ $B_{A2}, E_{D4}$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's**

![](_page_15_Picture_5.jpeg)

# • Cost=2

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's** Find shortest path from • **PriorityQueue=**{ $C_{B3}$ , $E_{D4}$ }

Source (S) to Target (T) • Visited= $\{S_0, A_{S_1}, D_{S_1}, B_{A_2}\}$ 

![](_page_16_Picture_7.jpeg)

# Find shortest path from Source (S) to Target (T) • Visited= $\{S_0, A_{S1}, D_{S1}, B_{A2}, B_{A2}, B_{A2}, B_{A2}, B_{A3}, B_{A$ C<sub>B3</sub> • Cost=3• **PriorityQueue=**{ $T_{C4}, E_{D4}$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's**

![](_page_17_Picture_5.jpeg)

# Find shortest path from Source (S) to Target (T) • Visited= $\{S_0, A_{S1}, D_{S1}, B_{A2}, B_{A2}, B_{A2}, B_{A2}, B_{A3}, B_{A$ **C**<sub>B3</sub>, **T**<sub>C4</sub>} • Cost=3• **PriorityQueue=** $\{E_{D_4}\}$

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- Dijkstra's**

![](_page_18_Picture_5.jpeg)

# destination • Cost(n) = F(n) + H(n)Actual Cost

© 2019 IEEE **International Solid-State Circuits Conference** 

# Graph Algorithms- A\*

- A\* guides the search towards the

  - F(n) = actual distance from source
    - H(n) = heuristic that predicts the distance to target
  - H(n) = 0 for Dijkstra's
- Provides optimal path if H(n) <=</li>

![](_page_19_Figure_10.jpeg)

![](_page_20_Figure_0.jpeg)

© 2019 IEEE International Solid-State Circuits Conference

![](_page_20_Figure_2.jpeg)

# • Let H(n) = ManhattanDistance • $H(n) = abs(x_i-x_T)+abs(y_i-y_T)$

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

21 of 62

Distance • Visited= $\{S_n\}$ • Cost/F(n)=0

© 2019 IEEE **International Solid-State Circuits Conference** 

# Graph Algorithms- A\* Find shortest path from Source (S) to Target (T)

- Let H(n) = Manhattan
- **PriorityQueue=** $\{A_{S_1}\}$

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_21_Figure_7.jpeg)

to Target (T) Distance • Visited= $\{S_n\}$ • Cost/F(n)=0

© 2019 IEEE International Solid-State Circuits Conference

# Graph Algorithms- A\* Find path from Source (S) • Let H(n) = Manhattan • **PriorityQueue=** $\{A_{s_1}^5\}$ C(A) = F(A) + H(A) = 1 + 4

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_22_Figure_5.jpeg)

Distance • Visited= $\{S_n\}$ • Cost/F(n)=0

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- A\*** Find shortest path from Source (S) to Target (T) • Let H(n) = Manhattan

# • **PriorityQueue=** $\{A_{S1}^{5}, D_{S1}^{5}\}$

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_23_Figure_6.jpeg)

 Find shortest path from Source (S) to Target (T) Let H(n) = Manhattan Distance • Visited= $\{S_0, A_{S_1}^5\}$ • Cost/F(n)=1• **PriorityQueue=**{ $B_{\Delta 2}^{5}$ ,  $D_{S1}^{5}$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- A\***

![](_page_24_Figure_5.jpeg)

 Find shortest path from Source (S) to Target (T) • Let H(n) = Manhattan Distance • Visited= $\{S_0, A_{S_1}, D_{S_1}, D_{S_1}, D_{S_1}, D_{S_1}\}$ • Cost/F(n)=1• **PriorityQueue=**{ $B_{A2}^{5}$ ,  $E_{D4}^{6}$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# Graph Algorithms- A\*

![](_page_25_Figure_4.jpeg)

 Find shortest path from Source (S) to Target (T) • Let H(n) = Manhattan Distance • Visited= $\{S_0, A_{S_1}^5, D_{S_1}^5, B_{A_2}^5\}$ • Cost/F(n)=2• **PriorityQueue=**{ $C_{B3}^{5}$ ,  $E_{D4}^{6}$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- A\***

![](_page_26_Figure_4.jpeg)

 Find shortest path from Source (S) to Target (T) • Let H(n) = Manhattan Distance • Visited= $\{S_0, A_{S_1}^5, D_{S_1}^5, B_{A_2}^5, B_{A$  $C_{B3}^{5}$ • Cost/F(n)=3• **PriorityQueue=**{ $T_{C4}^4$ ,  $E_{D4}^6$ }

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- A\***

![](_page_27_Figure_5.jpeg)

 Find shortest path from Source (S) to Target (T) • Let H(n) = Manhattan Distance • Visited= $\{S_0, A_{S_1}, D_{S_1}, B_{A_2}, B_{$  $C_{B3}^{5}, T_{C4}^{4}$ • Cost/F(n)=3• **PriorityQueue=** $\{E_{D_4}^6\}$ 

© 2019 IEEE **International Solid-State Circuits Conference** 

# **Graph Algorithms- A\***

![](_page_28_Figure_4.jpeg)

![](_page_29_Picture_0.jpeg)

- 4-neighbor grid • Directed edges

  - on/off control
  - edge weight
  - runtime

# **Graph Structure- This Work**

# Each direction binary 4 bit digital control Programmable at

![](_page_30_Picture_10.jpeg)

# Outline • Background: Path Planning Algorithms Time-Based A\* ASIC **– Time-Based Primer** - Top Level Design - Vertex Details - Edge Details 65nm Test Chip Results Applications Conclusion

© 2019 IEEE **International Solid-State Circuits Conference** 

![](_page_32_Figure_1.jpeg)

© 2019 IEEE **International Solid-State Circuits Conference** 

## **Digital Computing**

Advantages of digital arithmetic: – Binary representation – Less "buy-in" required – Existing IP for rapid SoC development – No calibration

## **Time-based Computing**

![](_page_32_Figure_8.jpeg)

## **Advantages of time-based circuits: Compact area** Low power consumption

High precision tunability

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_32_Picture_12.jpeg)

33 of 62

![](_page_33_Figure_0.jpeg)

# 40×40 A\* ASIC

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

# • Time $\propto$ Distance Each vertex stores first input **Directly readout** shortest path Multiple start points

![](_page_34_Figure_0.jpeg)

# 40×40 A\* ASIC- Vertex

![](_page_34_Figure_3.jpeg)

![](_page_35_Figure_0.jpeg)

# 40×40 A\* ASIC- Vertex

![](_page_36_Figure_0.jpeg)

# 40×40 A\* ASIC - Edge

![](_page_36_Figure_4.jpeg)

![](_page_37_Figure_0.jpeg)

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_37_Figure_4.jpeg)

# Outline Background: Path Planning Algorithms Time-Based A\* ASIC 65nm Test Chip Results Applications • Conclusion

© 2019 IEEE **International Solid-State Circuits Conference** 

# Test Setup Custom Test PCB

![](_page_39_Picture_1.jpeg)

![](_page_39_Picture_4.jpeg)

![](_page_40_Figure_0.jpeg)

# **Edge Delay Linearity**

## Edge Code

STEP

DNI

[T<sub>STEP</sub>]

Z

![](_page_40_Figure_6.jpeg)

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

Architecture	This Work	<b>FPGA</b> [4]	μProcessor	CPU [5]			
Product	ASIC	Xilinx Virtex	<b>ARM Cortex-M0</b>	Intel Xeon E5630	NVIC		
Technology	65nm	20nm	<b>40nm</b>	<b>32nm</b>			
Voltage	<b>1.2V</b>		<b>1.1V</b>	0.7-1.35V			
Peak Power	26.4mW	24.22W	127µW	20W/core			
Throughput [MTEPS]	559	731	5.34*10 <sup>-4</sup>	0.83			
Energy per Node	0.328pJ*	33nJ	89.1nJ	24.1µJ			
Normalized Energy	<b>1</b> x	<b>10</b> <sup>5</sup>	2.7x10 <sup>5</sup>	<b>1.19x10<sup>6</sup></b>			
<sup>55%</sup> from SRAM Program (does not include cache access energy) Energy/Node=Unit Delay*Unit Power							

[4]S. Zhou, et al, "High-Throughput and Energy-Efficient Graph Processing on FPGA," IEEE International Symp. on Field-Programmable Custom Computing Machines, pp. 103-110, 2016. [5] Y. Zhou and J. Zeng, "Massively Parallel A\* Search on a GPU," Conference on Artificial Intelligence *(AAAI)*, pp. 1248-1254, 2015.

# **Comparison Table**

# MTEPS = Million Traversed Edges Per Second

![](_page_41_Figure_8.jpeg)

# Outline Background: Path Planning Algorithms Time-Based A\* ASIC 65nm Test Chip Results Applications Conclusion

© 2019 IEEE International Solid-State Circuits Conference

• Applications: [Wikipedia]

© 2019 IEEE **International Solid-State Circuits Conference** 

# Voronoi Diagrams

۲

•

۲

- Segmentation of plane such that distance to seeds is maximized

  - kNN classification
  - Biological structures (bone and cells)
  - Computational fluid dynamics meshes
  - Autonomous vehicles

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

۲

![](_page_44_Figure_0.jpeg)

# **Collision Avoidance**

![](_page_44_Figure_4.jpeg)

![](_page_45_Figure_0.jpeg)

# **Collision Avoidance**

![](_page_46_Figure_0.jpeg)

# **Collision Avoidance**

© 2019 IEEE International Solid-State Circuits Conference

# Path Planning - Dijkstra's

5				
0	1	2	3	
1	2	3	4	
2	3	4	5	
3	4	5	6	
4	5	6	7	
5	6	7	8	

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_47_Figure_5.jpeg)

# All diagonal paths are equal

![](_page_47_Picture_7.jpeg)

© 2019 IEEE International Solid-State Circuits Conference

# Path Planning - Dijkstra's

![](_page_48_Figure_2.jpeg)

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

# Paths above blockage dominate paths below

49 of 62

.8 

© 2019 IEEE International Solid-State Circuits Conference

ath Planning - 1					
	TVV S 2		VTVV 5.4	VTV 6.8	
	0		2	3	
			6.8		
			3		
	11	9.4	8		9.
	6	5	4		
	12	10.6			10.
	7	6			
	13.2	11.6	12.2	11.4	10.
	8	7	8	9	
	14.2	12.4	12	11.4	1
	9	8	9	10	

![](_page_49_Picture_5.jpeg)

# Path Planning Finding shortest path w/o gradient 0.9V**0.9V** 0.7

![](_page_50_Figure_1.jpeg)

![](_page_50_Figure_4.jpeg)

![](_page_51_Figure_0.jpeg)

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

# Finding shortest path w/ gradient 0.7

![](_page_51_Picture_4.jpeg)

							L_ L_ L_ L_ L_	
		LLLLL			LLLL			LLI
5			La La La La La	La La La La La	La La La La La 1	La La La La La I	- La L <mark>a La</mark> La	L. L. 1_
-								L- L- 1
		LLLLL						LLI
10				to to to to to	to to to to to 1	La La La La La t	- La La La La	t_ t_ 1_
		to to to to to to	La La La La La	to to to to to	to to to to to	La La La La La L	- La La La La	t. t. 1
							_ L_ L_ L_ L_	
				La La La La La 1 1 1 1 1		La La La La La L		
15								
IJ	1 -1 -1 -1 -1 -1		La La La La La	to to to to to	La La La La La 1	La La La La La L	- t- t- t- t-	L L 1
		- La La La La La	La La La La La		to to to to to !	La La La La La L		L- L- 1
								L_ L_ 1
20						L. L. L. L. L. L.		
20								
		LLLLL						LLI
	للے لے لے لے ا			L. L. L. L. L.	L. L. L. L. L. !	L. L. L. L. L. I		L L 1
		to to to to to	Le Le Le Le Le	Le Le Le Le Le	to to to to to	La La La La La L	- to to to to	L- L- 1
25								
				La La La La La 1 1 1 1 1	1 + 1 + 1 + 1	La La La La La L	- L- L- L- L- 1 1 1 1	
								LLI
	1							L L 1
30			to to to to to	L. L. L. L. L.	to to to to to	Le Le Le Le Le I		L. L. I_
~~								L L !
25		LLLLL			4444			L L L
55		44444		4444	4 4 4 4 4			++1
		FFFFFF	FFFFF	FFFFF	FFFFF	rrrrr		FFI
	1 -1 -1 -1 -1 -1 -1	rrrrr						F F 1
10	1				rrrrr.			F F 1
40								
	5	10	15	20	25	30	35	40

# **Multicore Evaluation (1/5)** Global Map

![](_page_52_Figure_4.jpeg)

![](_page_52_Figure_7.jpeg)

![](_page_52_Picture_8.jpeg)

![](_page_53_Figure_0.jpeg)

## **Multicore Evaluation (2/5)** Global Map 15

![](_page_53_Figure_3.jpeg)

![](_page_53_Figure_6.jpeg)

![](_page_53_Picture_7.jpeg)

					·			-
								L 1
-			- L- L- L- L	- L. L. L. L. L.				
5			- L- L- L- L- L					L i
	1		- La La La La L		. La La La La La			L-1
				- L- L- L- L- L- 1 1 1 1 1				
10			LLLL		LLLL	LLLL	LLLL	L i_
10	1	- L. L. L. L. L. L.		- L. L. L. L. L.	. La La La La La		La La La La La	L 1
			- L. L. L. L. L. - 1. 1. 1. 1. 1	- L. L. L. L. L. - 1- 1- 1- 1- 1-	. L. L. L. L. L. L.			
								Li
15		- Le Le Le Le Le Le I	e Le Le Le Le L	- [- [- [- [- [-	. La La La La La	Le Le Le Le Le	La La La La La	L. !_
				- L. L. L. L. L.				
								L i
	1-1-1-1-1-1		- La La La La L	- La La La La La	. La La La La La			L_ 1
20								
								LI
		L to to to to to t	. L. L. L. L. L	- L. L. L. L. L.				L 1
05				- L- L- L- L- L-				
25								
		- Lo toolo to to to t	- t- t- t- t- t	- L. L. L. L. L.	. La La La La La	t. t. t. 📩		
			- L- L- L- L- L	- L- L- L- L- L-				
20								
30						L. L. L.		
			- L. L. L. L. L. - L. L. L. L. L	- L- L- L- L- L- - L- L- L- L- L-		La La La		
35						LLL		
~~								
				- L. L. L. L. L.				
								L 1
40		to to to for to to t		+ L+ f+ L+ L+ L+	. La fa La La La	La fa La La La	to for to to to	L- 1-
	5	10	15	20	25	30	35	40

© 2019 IEEE International Solid-State Circuits Conference

## Multicore Evaluation (3/5) Global Map 15

![](_page_54_Figure_4.jpeg)

0

![](_page_54_Picture_7.jpeg)

![](_page_54_Picture_8.jpeg)

![](_page_55_Figure_0.jpeg)

## © 2019 IEEE International Solid-State Circuits Conference

## **Multicore Evaluation (4/5)** Global Map 15

![](_page_55_Figure_5.jpeg)

![](_page_55_Picture_8.jpeg)

![](_page_55_Picture_9.jpeg)

![](_page_56_Figure_0.jpeg)

## **Multicore Evaluation (5/5)** Global Map 15

<b>#</b> = = = = = = = = = = = = = = = = = = =				FF!
15 20	25	30	35	40

![](_page_56_Figure_5.jpeg)

![](_page_56_Picture_7.jpeg)

![](_page_57_Figure_0.jpeg)

# **Optics Experiment**

		FFFF	
<u> </u>	י <mark>ד רררו</mark> ידיררי		

![](_page_57_Picture_5.jpeg)

F								
5								
10				· · · · · · · · · · · · · · · · · · ·	·	$ \begin{bmatrix}                                    $		
15	$ \begin{bmatrix} - & - & - & - & - & - & - & - & - & - &$							
20		$ \begin{array}{c} \cdot $						
25								
30						$\begin{array}{c} 1 \\ $		
35						$\begin{array}{c} 1 \\ $		
40	<u></u>	<u>יב ב ב</u> 10	<u></u>	20	<u>2222</u> 25	22 <mark>2</mark> 55 30	<u>35</u>	<mark>בן</mark> 40

# **Optics Experiment**

![](_page_58_Picture_4.jpeg)

# Outline Background: Path Planning Algorithms Time-Based A\* ASIC 65nm Test Chip Results Applications Conclusion

© 2019 IEEE **International Solid-State Circuits Conference** 

![](_page_60_Picture_0.jpeg)

![](_page_60_Picture_1.jpeg)

© 2019 IEEE International Solid-State Circuits Conference

# **Die Photo and Chip Summary**

![](_page_60_Figure_4.jpeg)

<b>Applications</b>	A* shorte st path, obstacle avoidance, scientific computation (optics)
Technology	65nm LP CMOS
Architecture	Time-based
<b># of Vertices</b>	1600
Unit Area	249µm <sup>2</sup>
# of Edges	6400
Edge Resolution	4b + Analog Gradient
Voltage	<b>1.2V</b>
Peak Power	26.4mW
Delay per Node	1.79ns @ [V <sub>B</sub> =.9V, V <sub>DD</sub> =1.2V]
Power per Node	183.1μW
Energy per Node	0.238pJ

- A\* heuristic implemented with analog bias gradient
- Vertex cells asynchronously evaluate and lockout
- Diverse range of applications
- Scalable to multicore for larger maps
- Orders of magnitude improvement in energy efficiency

© 2019 IEEE **International Solid-State Circuits Conference** 

# Conclusion

• 40×40 A\* ASIC in 65nmLP

This research was supported in part by the National Science Foundation under award number CCF-1763761.

2.5: A 40 × 40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

![](_page_61_Picture_16.jpeg)

62 of 62

![](_page_62_Picture_1.jpeg)