A Shortest Path Finding Time-based Accelerator Core with Built-in Gravity Control and Buffer Zone for Smooth 3D Navigation

Luke R. Everson, *Student Member, IEEE,* Jeehwan Song, *Student Member, IEEE,* Sachin S. Sapatnekar, *Fellow, IEEE,* and Chris H. Kim, *Fellow, IEEE*

***Abstract*— A mixed-signal time-based 65nm application specific integrated circuit is developed for solving shortest-path problems in 3D. Previous path planning ASICs have been restricted to 2D maps due to computational complexity or physical architecture limitations. Our time-based, asynchronous, one-shot architecture has been coupled with a novel dual axis interleaving strategy to solve the multi-dimensional shortest path problem in a simple, energy efficient manner. Additional features include circuit-based solutions for obstacle blockage avoidance and gravity. The efficacy of the proposed ASIC is evaluated on a drone navigation application, 3D Voronoi diagrams, and a physical optics experiment. The chip is twice as energy efficient as prior 2D work while containing 5x more vertices and 7.5x additional edge connections.**

***Index Terms*— A* algorithm, graphs, graph computing, single-source shortest path, time-domain computing, time-to-digital converter**

Drone navigation, underwater vehicle navigation, and unmanned ground vehicles, require a path planning algorithm that can handle a 3D map. Unfortunately, previous path planning ASICs have been restricted to 2D maps [1,3,5]. 3D navigation is so computationally expensive that drone navigation applications often ignore the third dimension as seen in Fig. 1 (left) [3]. In this paper, a 3D path planning time-domain ASIC featuring a scalable architecture, gravity, and obstacle buffering is presented along with three applications. This time based architecture leverages compute-in-memory where the spatial location of the memory positions has meaning and the computation occurs *in-situ.* Similar to [2], each element in the planarized array is connected to 6 adjacent elements, and a wavefront is propagated through the planarized array. If the first input to the vertex is the fastest way the wavefront arrives, by induction, tracing this back through the previous vertex will find the shortest path to the starting point.

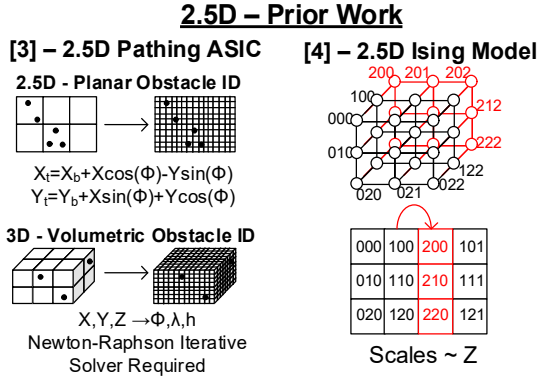

Fig. 1: Prior work of path planning chips targeted at 3D, but falls short in that they do not fully compute the third dimensions due to computational limitations [3] or limited connectivity [4].

Manuscript received on October 23, 2019 and revised on January 19 and April 2, 2020. This research was supported in part by the National Science Foundation under award number CCF-1763761.

L.R. Everson, J. Song, S.S. Sapatnekar, and C.H. Kim are with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, Minnesota, 55455, USA (email: chriskim@umn.edu).

One of the primary challenges associated with developing this concept was projecting a 3D volume onto a 2D plane. This limitation is seen in Fig. 1 (right) [4] where a 3D network of "spin" elements used to solve combinatorial optimization problems was implemented in a 2D chip. The design maps a two plane "3D" map into the 2D plane by nesting, or interleaving, the z-axis with x-axis. They manage to implement this only for a single bit realizing a 2.5D structure. This approach of mapping a 3D cube to a 2D plane does not work in general for a large 3D array.

Fig. 2 illustrates the solution leveraged by the proposed accelerator core. Near the top is the 3D volume representation of the cube. Instead of simply interleaving the z-axis in one

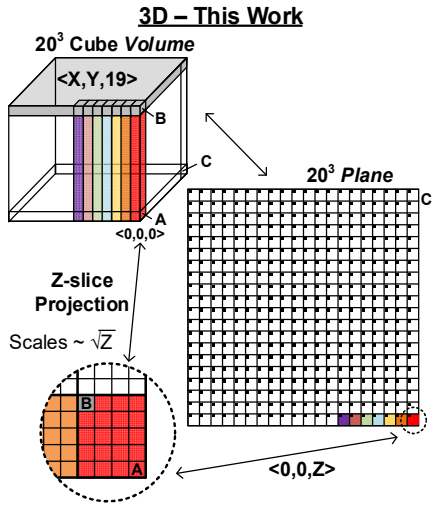

Fig. 2: The proposed mapping between the 2D planar CMOS ASIC and the 3D application space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSSC.2020.2995051, IEEE Solid-State Circuits Letters

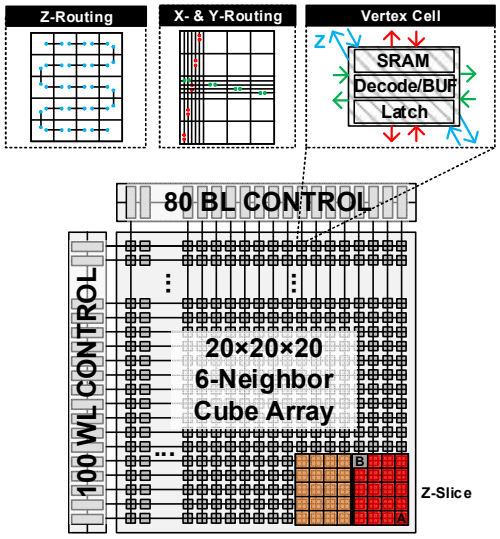

Fig. 3: Block diagram of the 3D path planning ASIC. The routing in each dimension is shown in the top boxes. The highlighted groups of cells in the bottom right of the array correspond to a Z-slice.

direction, it is distributed in both the x- and y-axis. This reduces the routing overhead by $\sqrt{z}$ compared to z. This is seen in the colored Z-pillars on the front right side of the cube. This vertical pillar fits into the global grid in the flattened array. Each of the Z-slice locations corresponds to the <X,Y> location in the array. The points A, B, and C are identified in the three representations with their appropriate locations marked to aid in understanding the translation between the different structures.

The top level block diagram is shown in Fig. 3. The core of the chip is the 6-neighbor 20×20×20 array. On the peripherals, the wordline and bitline control blocks enable access to the SRAM cells as in a standard array. The routing methodology shown in Fig. 3 (top callouts) connects cells in this efficient

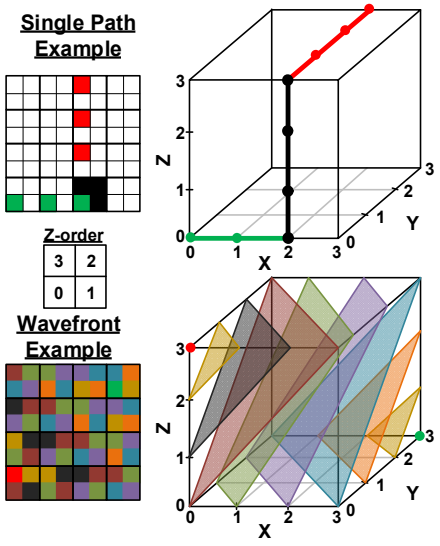

Fig. 4: Single path example shows the mapping of the traversed path in the volume to the recorded path in the core. The different directions of travel are shown in different colors. In the wavefront example, each colored plane shows the relationship between time steps and the position from the start node. This is also translated back to the planar storage location in the core.

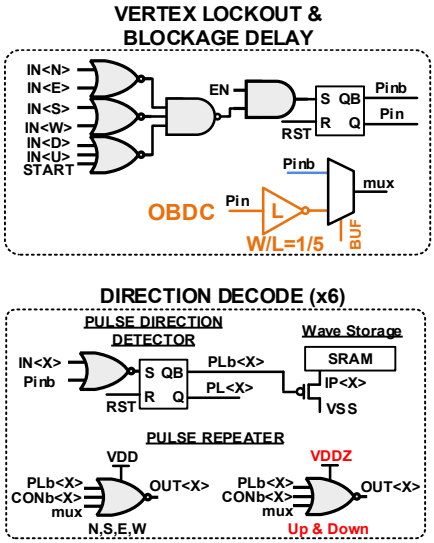

Fig. 5: Circuit schematics of the vertex lockout, blockage delay, and direction decoder. SRAM direct memory access programs the bit in the array corresponding to the pulse input direction.

layout. Each vertex has incoming and outgoing connections in all six directions shown here as a single line. The x-axis routes are staggered four wide and the y-axis is staggered five wide. This difference is due to the unit cell aspect ratio in the layout of a 6T SRAM cell. This makes the routing structure dense, regular, and automation-friendly because the coordinate location of the cell dictates the location of the tapping vias to the global routes. A simple example of the routing structure for a 4x4x4 cube is shown in Fig. 4 (upper). Each z-slice represents a 2x2 group in the overall example array. Fig. 4 (lower) describes the position in the volume to the ordering the array. Additionally, each triangular plane represents equidistant points from the start node.

Fig. 5 shows the details of the circuit schematic of the vertex cell. The key functionality of the vertex is to detect if a pulse arrives, decode the direction of the first pulse, latch this information in-situ, and propagate the pulse to the connected neighbors. The output of the lockout merging latch is fed into the decoders and the obstacle buffer delay circuit (OBDC). Decoder outputs program SRAMs in each vertex used to readout the shortest path. OBDC generates a longer delay, analogous to a buffer zone, around the obstacles. OBDC is enabled within some range of the boundary of the obstacle to penalize the risky routes. This single bit is stored locally in each vertex and is programmable, just like the rest of the array, by the user at runtime.

The decoder network is tasked with determining which direction carried the first pulse. Each of the six directions has a copy of this circuit with the input connected to the corresponding direction. Additionally, the decoder output, *PLb<x>,* is fed into the pulse repeater block. A unique feature of pulse repeater circuit is that the z-axis (Up and Down) NOR gate VDDZ is separate from the core VDD. Recall from Fig. 3 the distance of the z-axis routing was a single unit, whereas the x-axis and y-axis were four and five units respectively. Having the ability to reduce the z-axis VDDZ compared to the core VDD gives the ability to match the short RC delay to the longer RC. This can also be modulated to penalize travel in

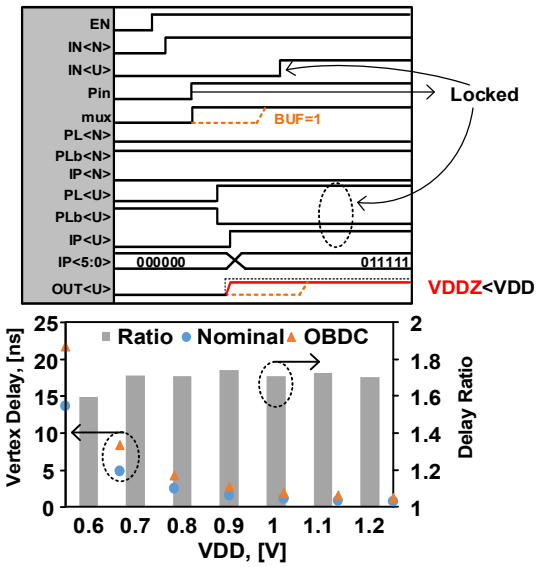

Fig. 6: (Top) Timing diagram of activated vertex from North direction. Lower chart shows nominal vertex delay and delay with blockage control enabled across supply voltages.

the Z-axis, which is analogous to "gravity" along the Z-dimension. The impact of gravity in circuit performance can be seen in Fig. 6 (lower). The delay increases uniformly by 1.7x across a wide range of supply voltages the making the OBDC a solution that is robust to PVT variations.

With-in die PVT variations are minimized in the time based architecture. Since interactions only happen between close neighbors spatially distributed variations, such as implant doping and temperature differences, have a matched effect. Long delay chains will have larger variation than shorter chains, but grows proportional to the square root of the number of delay units compared to linearly. Impact from

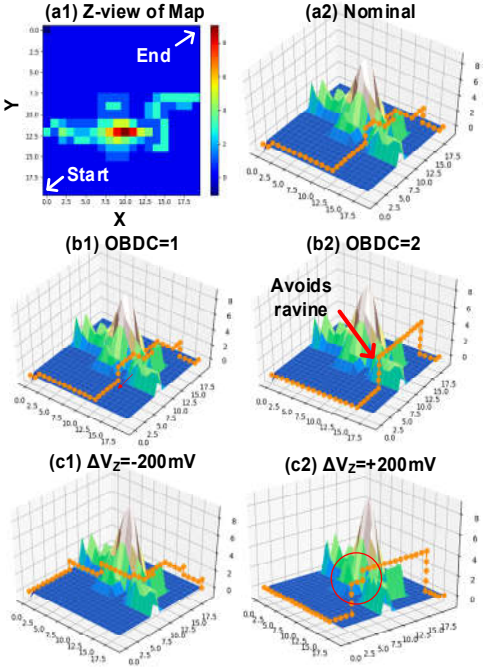

Fig. 7: 40x40 graph ASIC chip for solving single-source shortest path problems based on 2-dimensional wavefront expansion.

Table I. Performance Comparison

Architecture	3D Time-Based [This Work]	2D Time-Based ISSCC'19 [1]	Digital ASIC ISSCC'19 [3]
Technology	65nmLP	65nmLP	22nm
Voltage	1.2V	1.2V	0.6-1V
Peak Power	536.8mW	26.4mW	37.5mW
Throughput [MTEPS]	1383.1	558.7	0.225
Energy per Node	0.162pJ	0.328pJ	165nJ
Efficiency Factor	1×	0.5×	10^{-5}×

random variations will cause errors in the accuracy of the path, especially for smaller maps with fewer delay stages, but longer paths will be less susceptible as the deviation will be averaged out.

Fig. 7 shows an example of the 3D navigation application for autonomous drones. The goal of the application is to find the shortest path from (0,0,0) to (19,19,0). Blockages are programmed based on the map (Fig. 7a1) arbitrarily determined by the application. In this example, there is a blockage that spans the entire horizontal axis near y=10. Along the diagonal, or shortest path from start to finish, the highest point in the obstacle range forces the shortest path off the diagonal through a ravine (Fig. 7a2). In Fig. 7b, the OBDC is enabled and the buffer range is set to 1 unit (Fig. 7b1). The shortest path still follows the risky ravine, but avoids the lower parts of the blockage. With the OBDC set to 2 units (Fig. 7b2) the shortest path avoids the ravine completely. With stronger gravity (Fig. 7c1, ΔVDDz=-200mV), the shortest path takes incremental z-axis increases, compared to Fig. 7b1, as vertical steps are more expensive than planar steps. Low gravity, or incentivized Z routes (ΔVDDz=+200mV), in Fig. 7c2 show the shortest path traveling in the Z-axis preemptively and entirely avoiding the ravine. The OBDC and gravity features enable tradeoffs to optimize risk and optimal routes based on application requirements. Further study is needed to determine methods for increasing the array size, or grouping many cells into one to increase the directional resolution [1].

Fig. 8 shows how the Voronoi algorithm (VA) segments a volume into regions that are closest to the various seed nodes. VAs have applications in k-nearest neighbors in machine learning and mesh generation in CFDs. Fig. 8 shows how this core can determine the partitions from the VA. The circles represent start nodes and the arrows represent the latched pulse direction at intermediate nodes. The arrow boundary is

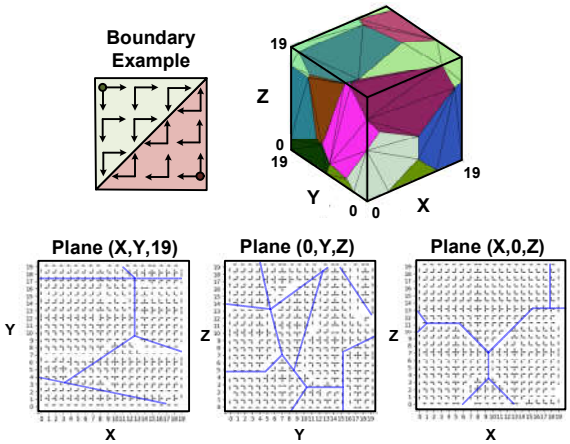

Fig. 8: 40x40 graph ASIC chip for solving single-source shortest path problems based on 2-dimensional wavefront expansion.

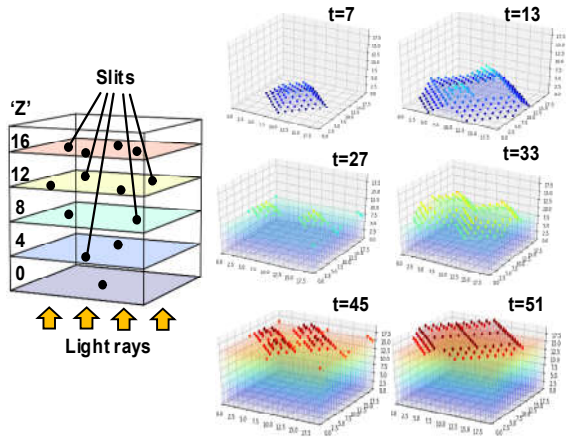

Fig. 9: Example of the physical optics experiment recreated from [5]. Time steps of the output have been included to show the wave propagation through the different z-levels.

equidistant from the start nodes, which is the expected result in VA. At the beginning of the computation there are only the start points available. The core is enabled, and wavefronts propagate from all start nodes and the plane is segmented by the wavefront collisions. This is visualized by the simulated Voronoi diagram (VD) that looks like shards of glass. Three planes slice the core at <X,Y,19>, <0,Y,Z>, and <X,0,Z>. Each of the corresponding planes has been obtained from the fabricated test chip, annotated where the wavefronts met.

Fig. 9 shows a recreation of the optics slit experiment from [6]. Waves propagate through a medium via Euclidean geometries, but in this chip the same phenomenon is observed in Manhattan geometry. The left diagram shows the barriers at the different Z-levels, and the black dots correspond to the single opening slits in the barrier. Barriers can be programmed by enabling *CONb<x>* in Fig. 5. The single start point is at the middle of the Z=0 plane. Samples from successive time points are shown to reconstruct the 3D wavefront. For example, at T=13 the wavefront has just cleared the Z=4 barrier at the two openings and the two pyramidal shapes correspond to the split wavefront.

Table I highlights the performance of the 3D time-based accelerator core by comparing it to previous path planning ASICs. The 3D version is twice as energy efficient as a 2Dtime-based ASIC [1], and is five orders of magnitude more energy efficient than a custom digital ASIC [3]. The energy does not include generating the gravity supply or readout phase. Because the answer is encoded in the spatial positioning, the array needs to be read once. Whereas in a

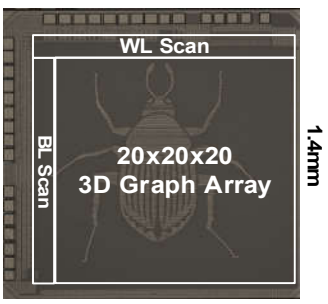

Fig. 10: Chip micrograph. Each vertex in the array can be accessed from the peripheral scan chain.

Table II. Chip Summary

Applications	3D shortest path Voronoi diagrams Scientific computation (optics)
Technology	65nm LP CMOS
Architecture	Time-based
# of Vertices	8,000
Vertex Area	156.8μm^2
# of Edges	48,000
Edge Resolution	1b + Vertical "Gravity"
Voltage	1.2V
Peak Power	1.07W (4800 cells)
Delay per Node	720ps @ [V_{DD}=1.2V]
Power per Node	224μW
Energy per Node	0.162pJ

conventional approach the memory would have to be read many more times to retrieve data for processing. Peak power in this chip corresponds to activating 4800 vertex cells at the same time. This scenario corresponds to starting a pulse at the center of eight sub-cubes and propagating outward at a uniform velocity. In [1] peak power corresponds to 156 cells on the outer edge which yields a similar energy expenditure/cell. The architecture of [3] relies on a conventional von Neumann architecture and thus peak power is not fair comparison. The energy efficiency metric tracks the energy consumed to service a vertex which provides a universal comparison between different architectures. The die microphotograph is shown in Fig. 10. The chip feature summary is compiled in Table II. The asynchronous vertex lockout architecture enables the chip to operate at this full frequency with minimal control signals. While the time-based architecture offers energy efficiency and throughput benefits for the core path finding operation, the accuracy limits of the path delays, along with the required full array readout and analog bias routing, must be thoroughly addressed before large scale adoption of this architecture is possible. Further study could be directed towards a hybrid architecture where the time-domain wavefront is discretized in time and implemented in distributed manner in digital logic.

References

[1] L. R. Everson, S. S. Sapatnekar and C. H. Kim, "A 40x40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2-Dimensional Gradient Control," in *International Solid-State Circuits Conference*, San Francisco, CA, 2019.

[2] M. Liu, L. R. Everson and C. H. Kim, "A Scalable Time-Based Iintegrate-and-fire Neuromorphic Core with Brain-Inspired Leak and Local Lateral Inhibition Capabilities, in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4.

[3] V. Honkote, et al., "2.4 A Distributed Autonomous and Collaborative Multi-Robot System Featuring a Low-Power Robot SoC in 22nm CMOS for Integrated Battery-Powered Minibots," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisoc, CA, USA, 2019, pp. 48-50.

[4] M. Yamaoka, et al, "24.3 20k-spin Ising chip for combinational optimization problem with CMOS annealing," in *2015 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2015, pp. 1-3.

[5] T. Takemoto et al, "2.6 A 2 ×30k-Spin Multichip Scalable Annealing Processor Based on a Processing-In-Memory Approach for Solving Large-Scale Combinatorial Optimization Problems," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019, pp. 52-54.

[6] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Transactions on Electronic Computers,* Vols. EC-10, no. 3, pp. 346-365, Sept. 1961.