Embedding-Based Placement of Processing element Networks on FPGAs for Physical Model Simulation

Bailey Miller*, Frank Vahid*, Tony Givargis**
Intern at SpaceX

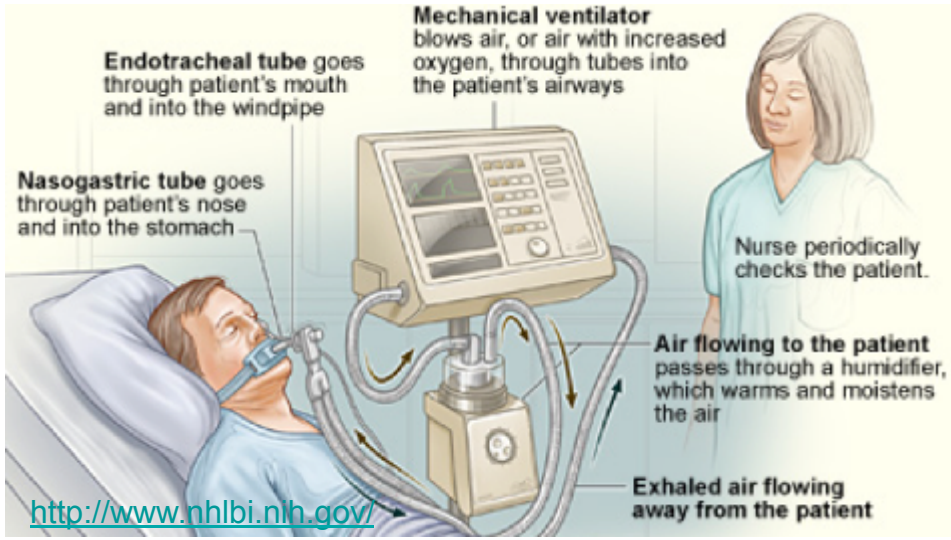
*Univ. of California, Riverside
**Univ. of California, Irvine

Supported by the NSF (CNS1016792, CPS1136146), and
Semiconductor Research Corporation (GRC 2143.001)

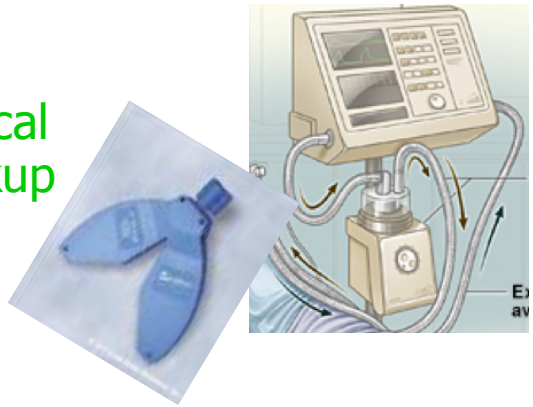

SPACEX

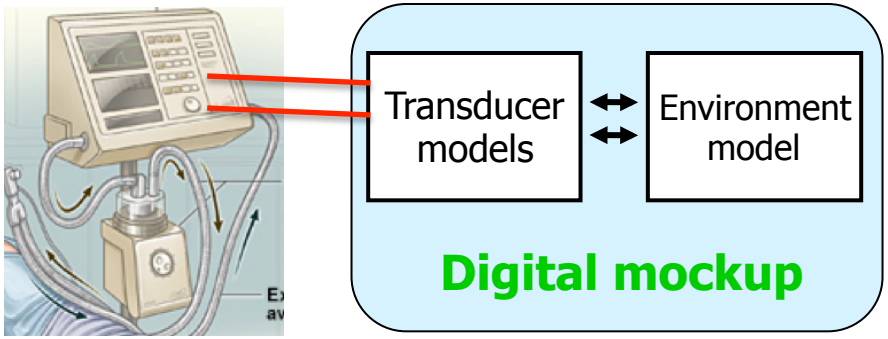
Models of physical world that run in real-time

Test cyber-physical systems



Physical mockup





Issue: Real-time achieved via inaccuracy

$$P_{tP} = \begin{cases} P_0 + A\left(1 - e^{-\frac{t}{tauR}}\right), & t = 0 \\ P_0 + A\left(1 - e^{-\frac{t}{tauF}}\right), & t = 0 \end{cases}$$

$$P_{tP} = P_0 + A\left(1 - \cos(2\ PI\ ft)\right)$$

$$uQ_1 = if(Q_1 > 0)\ 1\ else\ 0$$
$$uQ_2 = if(Q_2 > 0)\ 1\ else\ 0$$
$$uQ_3 = if(Q_3 > 0)\ 1\ else\ 0$$
$$umQ_1 = if(-Q_1 > 0)\ 1\ else\ 0$$
$$umQ_2 = if(-Q_2 > 0)\ 1\ else\ 0$$
$$umQ_3 = if(-Q_3 > 0)\ 1\ else\ 0$$

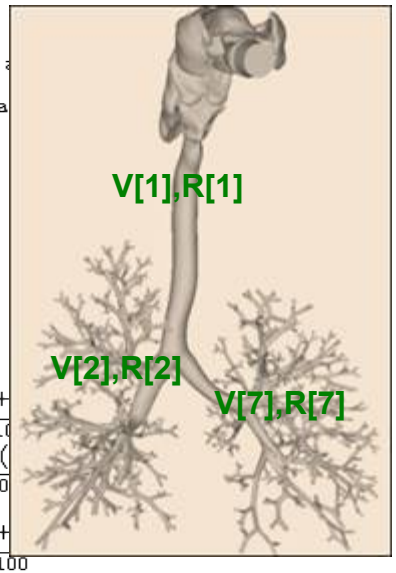
$$Cao = c00$$
$$\frac{d}{dt}c00 = \frac{Q_1\left(uQ_1\ Cin - umQ_1\ c01 + \right.}{VD\ /\ 10}$$
$$\frac{d}{dt}c01 = \frac{Q_1\left(uQ_1\ c00 - umQ_1\ c02 + (\right.}{VD\ /\ 10}$$
$$\cdots$$
$$\frac{d}{dt}c98 = \frac{Q_1\left(uQ_1\ C97 - umQ_1\ c99 + \right.}{VD\ /\ 100}$$
$$\frac{d}{dt}c99 = \frac{Q_1\left(uQ_1\ c98 - umQ_1\ C_m + (-uQ_1 + umQ_1)c99\right)}{VD\ /\ 100}$$

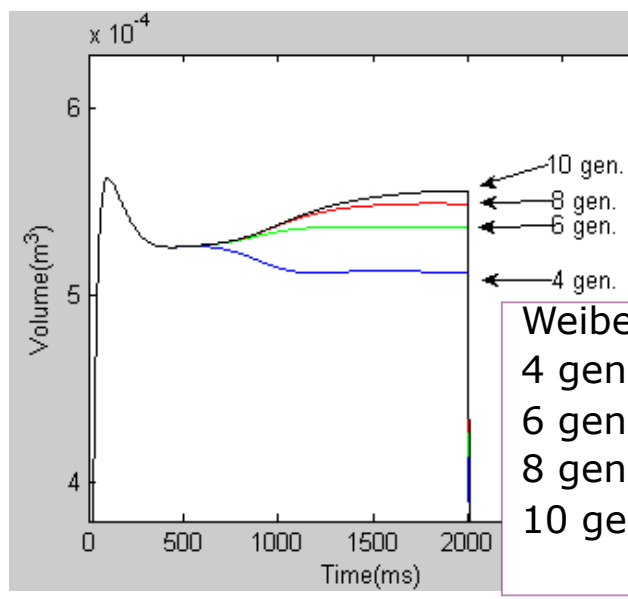
$$C_m = \frac{uQ_1\ Q_1\ c99 - umQ_2\ Q_2\ C2 - umQ_3\ Q_3\ C3}{uQ_1\ Q_1 - umQ_2\ Q_2 - umQ_3\ Q_3}$$

$$\frac{d}{dt}V_2 = V_2\left(-R_1\frac{d}{dt}V_3 - h_2\ e^{\frac{V_2}{epj_2}} + P_{tP}\right)(V_2\ R_1 + b_2)^{-1}$$
$$\frac{d}{dt}V_3 = V_3\left(-R_1\frac{d}{dt}V_2 - h_3\ e^{\frac{V_3}{epj_3}} + P_{tP}\right)(V_3\ R_1 + b_3)^{-1}$$

$$V_2\frac{d}{dt}C_2 = Q_2\,(C_m - C_2)uQ_2$$
$$V_3\frac{d}{dt}C_3 = Q_3\,(C_m - C_3)uQ_3$$

$$Q_2 = \frac{d}{dt}V_2$$
$$Q_3 = \frac{d}{dt}V_3$$
$$Q_1 = Q_2 + Q_3$$



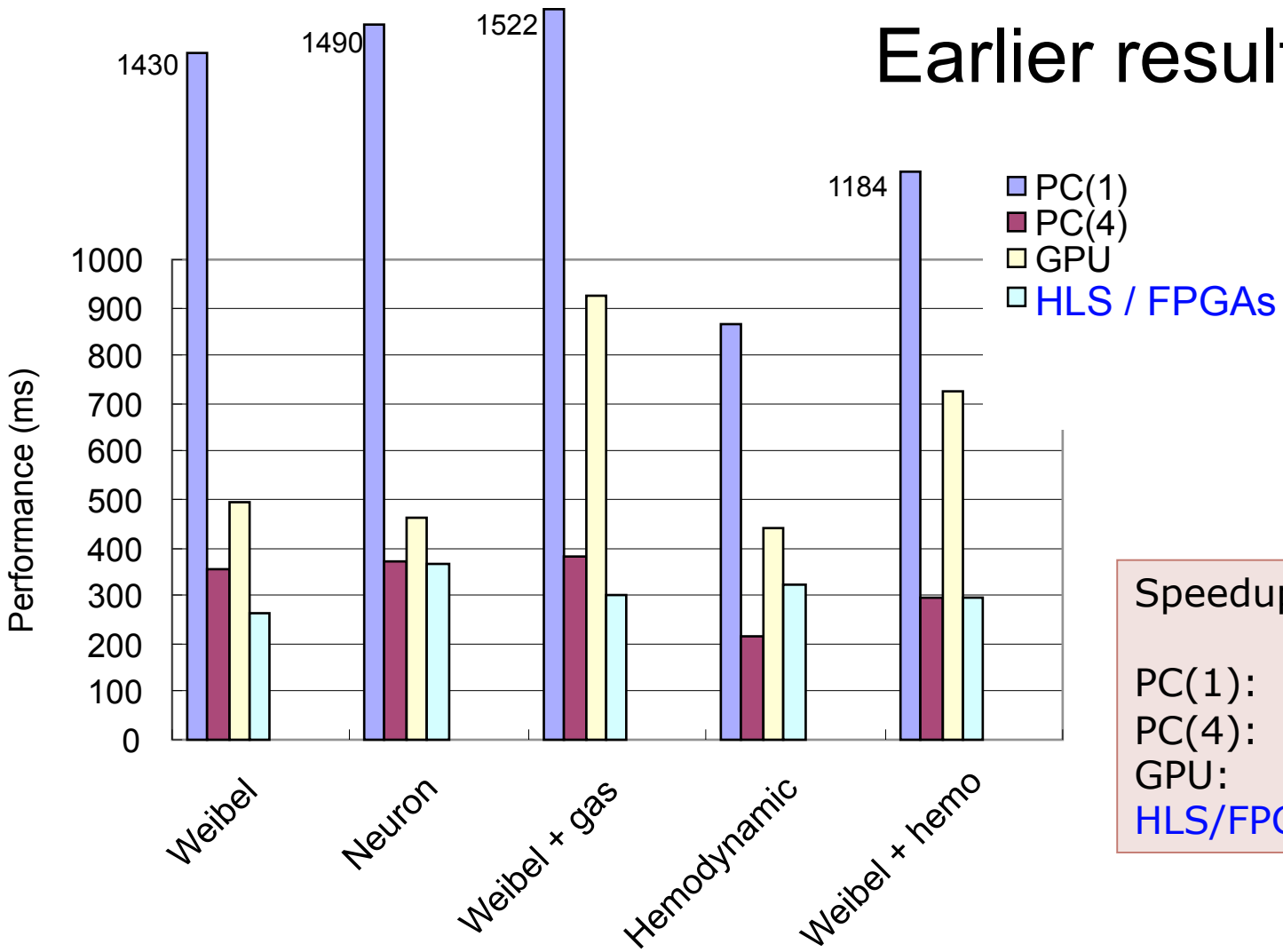


"2-3 minutes to simulate one breath accurately"

Weibel lung complexity

4 gen:	32 ODEs
6 gen:	128 ODEs
8 gen:	512 ODEs
10 gen:	2048 ODEs

Earlier results



Speedup vs real-time

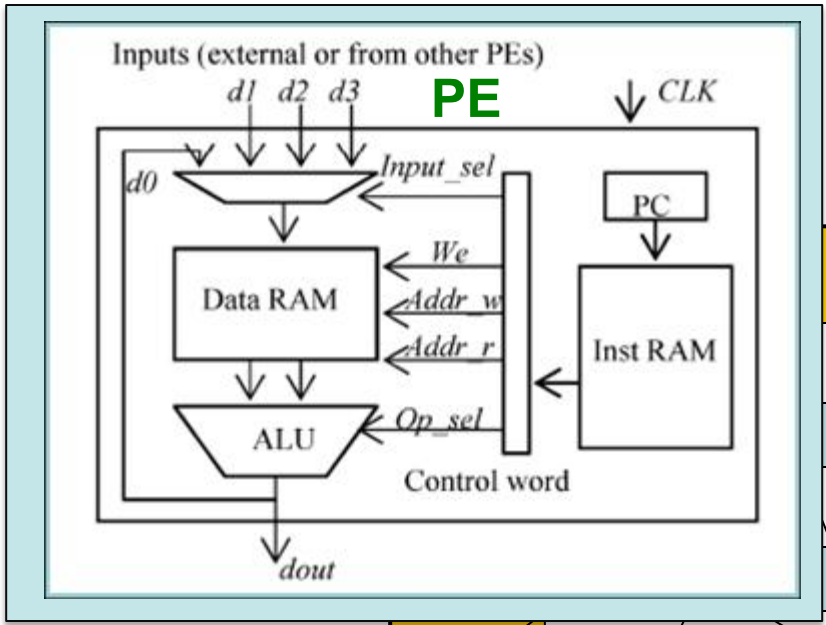
PC(1):	0.8x
PC(4):	3.1x
GPU:	1.6x
HLS/FPGA:	3.2x

Parallel computations +

- Neighbor communication

→ Seem like great match for FPGAs

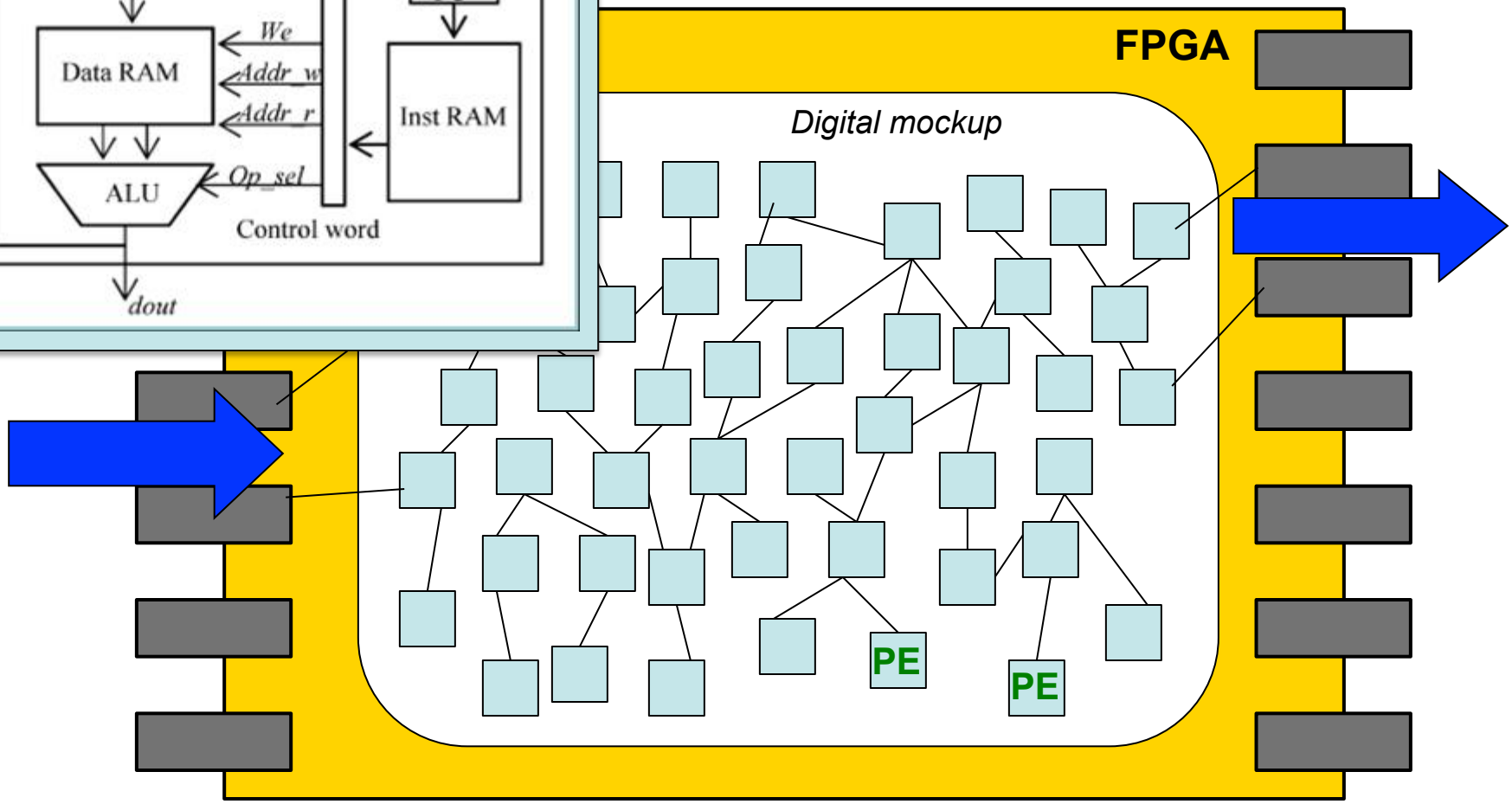
Network of synchronized PEs on FPGAs



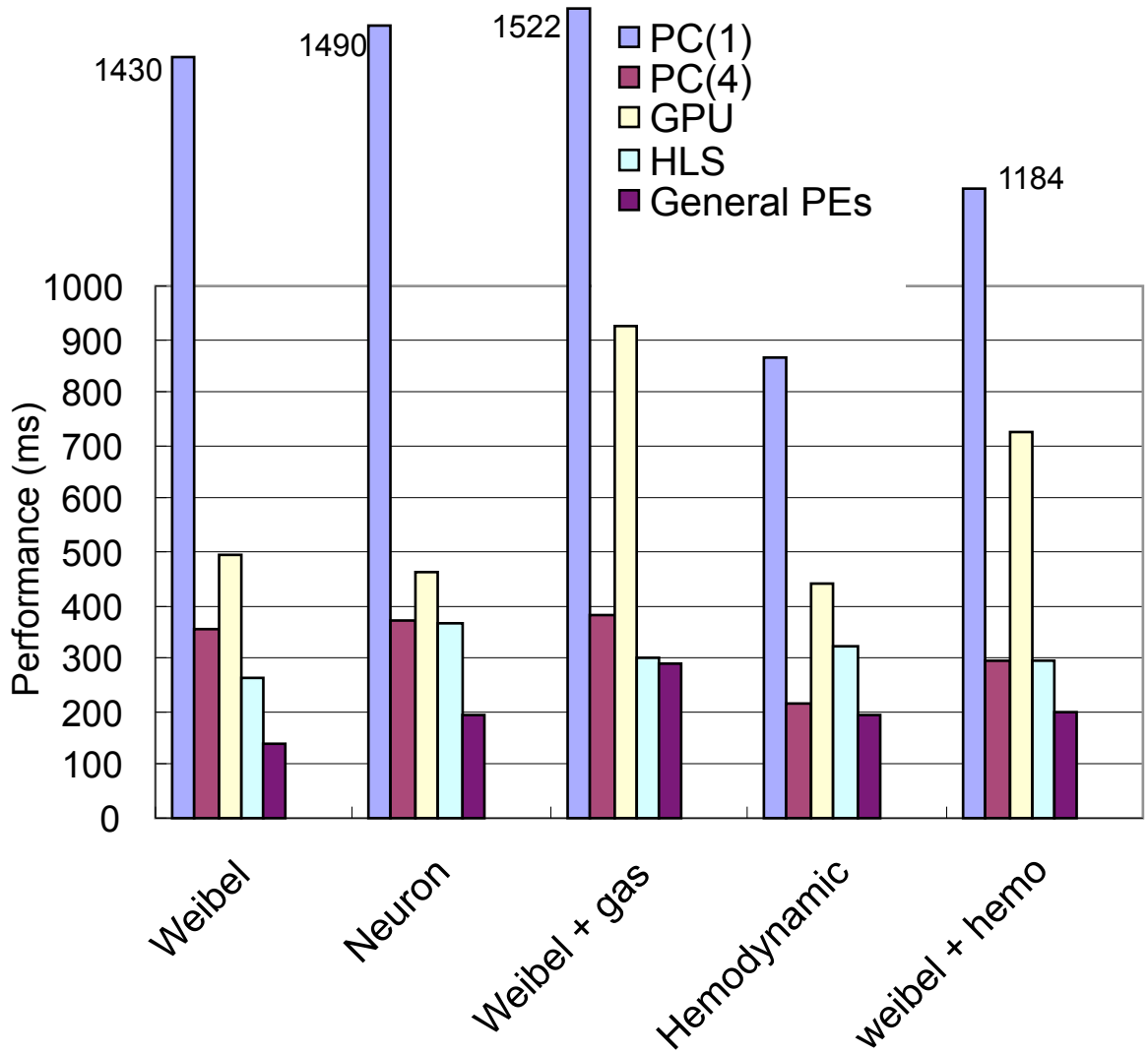
General Processing Element

- Iterative ODE solver (Euler/RK4)
  0.1 ms / 0.01 ms timestep

1 PE: 300 MHz



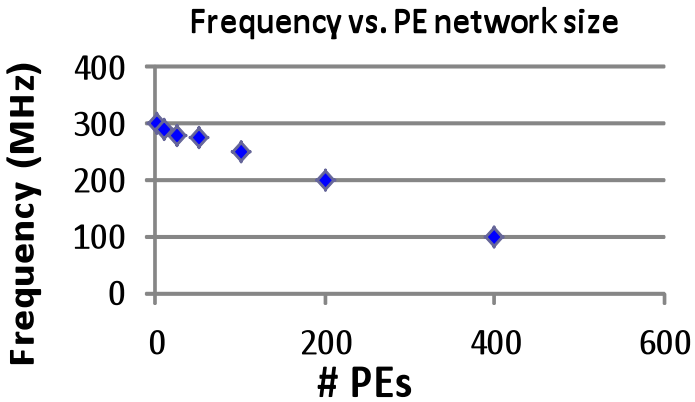
Earlier results

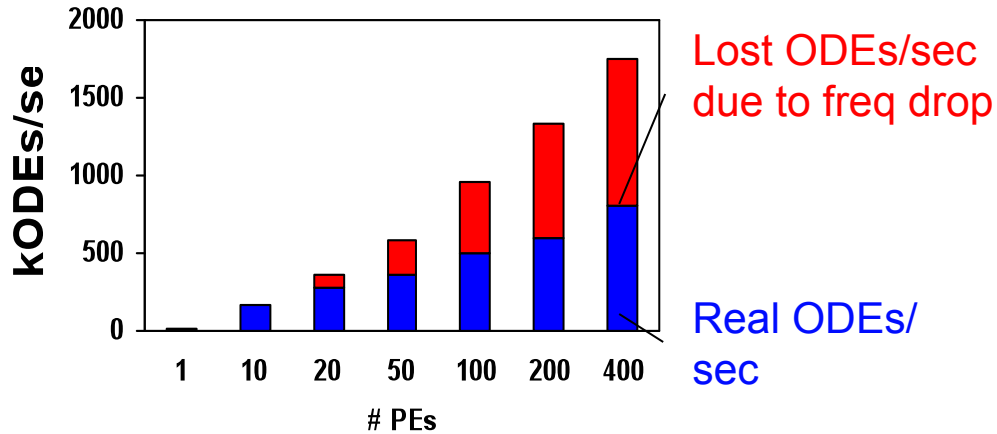


Speedup vs real-time	
PC(1):	0.8x
PC(4):	3.1x
GPU:	1.6x
HLS:	3.2x
General PE:	4.9x

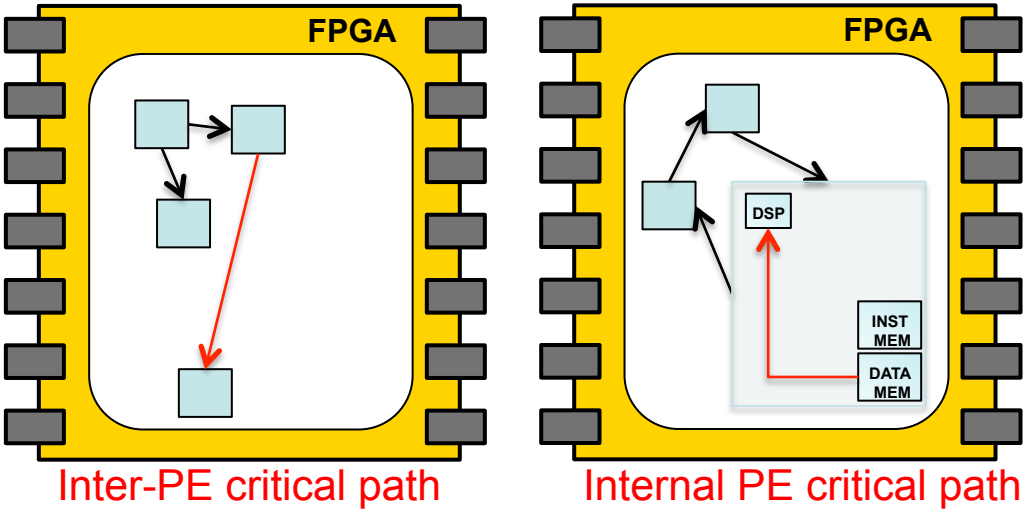
Current work

Problem: More PEs → Lower frequency





*11-gen Weibel model, Virtex6 240T FPGA, **general PEs***



Earlier approach

Phase

$$P_{1P} = \begin{cases} P_0 + A\left(1 - e^{-\frac{t}{tauR}}\right), & t = 0 \quad \text{at beginning of inhalation} \\ P_0 + A\left(1 - e^{-\frac{t}{tauP}}\right), & t = 0 \quad \text{at beginning of exhalation} \end{cases}$$

$$P_{1P} = P_0 + A\left(1 - \cos(2\ PI\ ft)\right)$$

$$uQ_1 = if\left(Q_1 > 0\right)\ 1\ else\ 0$$
$$uQ_2 = if\left(Q_2 > 0\right)\ 1\ else\ 0$$
$$uQ_3 = if\left(Q_3 > 0\right)\ 1\ else\ 0$$
$$umQ_1 = if\left(-Q_1 > 0\right)\ 1\ else\ 0$$
$$umQ_2 = if\left(-Q_2 > 0\right)\ 1\ else\ 0$$
$$umQ_3 = if\left(-Q_3 > 0\right)\ 1\ else\ 0$$

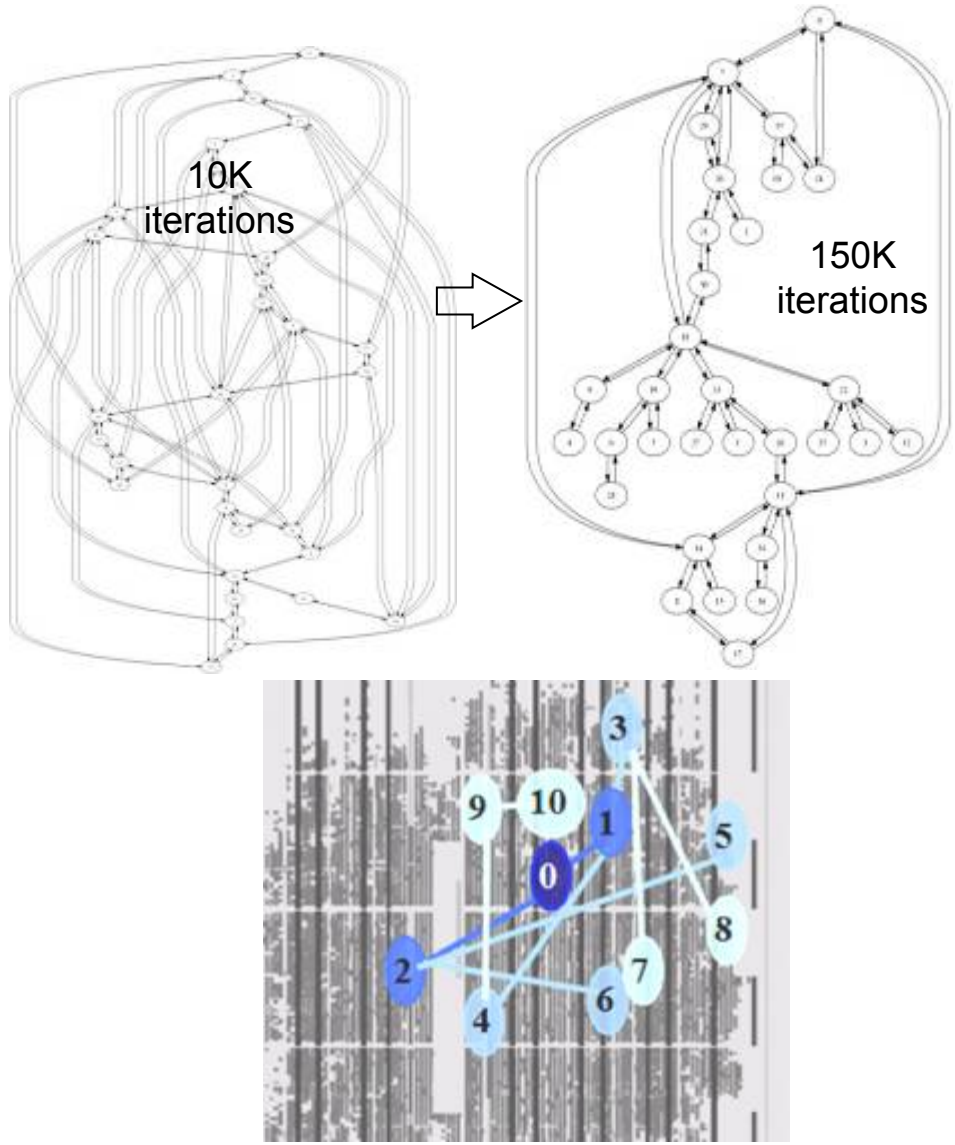
$$\frac{d}{dt}V_3 = V_3\left(-R_1 \frac{d}{dt}V_2 - h_3\, e^{\frac{V_3}{epsilon_3}} + P_{1P}\right)\left(V_3\, R_1 + b_3\right)^{-1}$$

$$V_2 \frac{d}{dt}C_2 = Q_2\,(C_m - C_2)\,uQ_2$$
$$V_3 \frac{d}{dt}C_3 = Q_3\,(C_m - C_3)\,uQ_3$$

$$Q_2 = \frac{d}{dt}V_2$$
$$Q_3 = \frac{d}{dt}V_3$$
$$Q_1 = Q_2 + Q_3$$

1 Maps ODEs to ***virtual*** PEs using simulated annealing

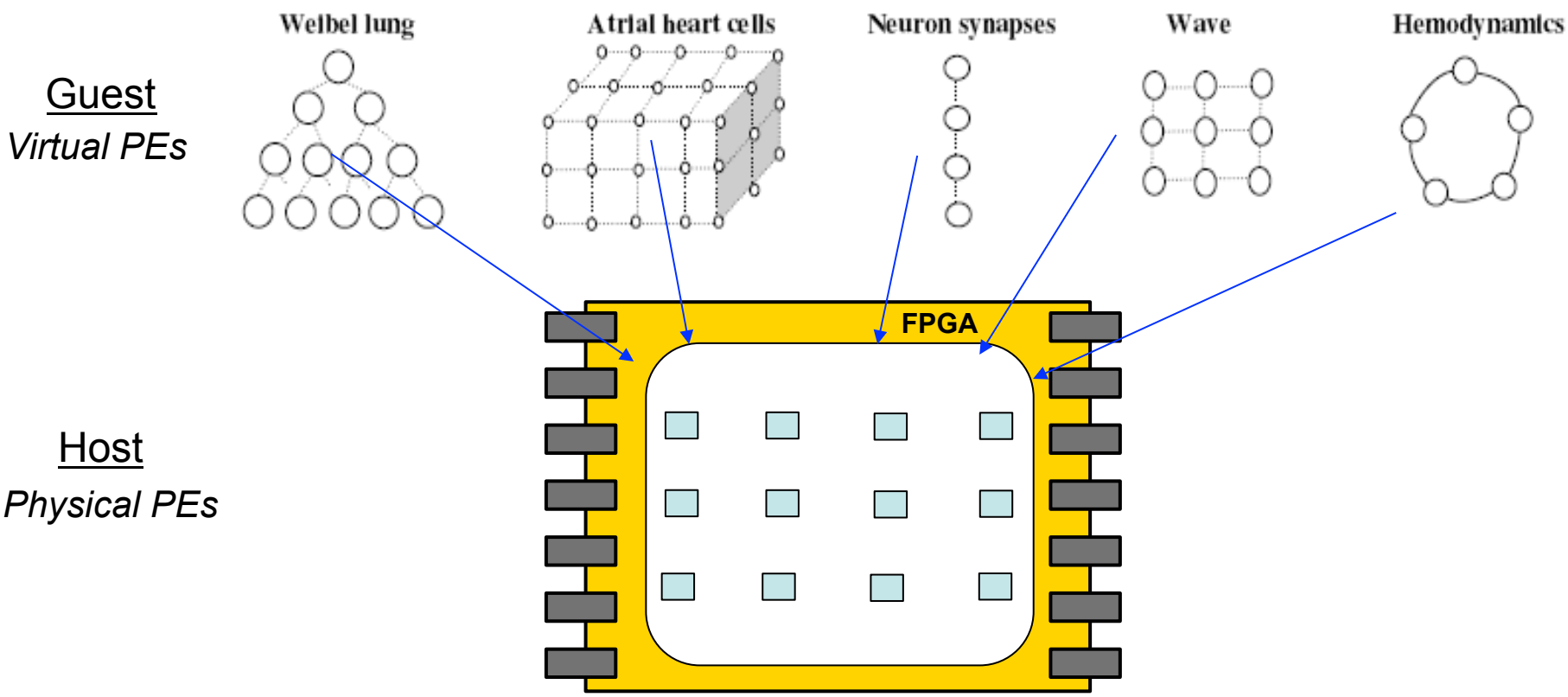
2 Convert virtual PEs to physical circuits using FPGA place-route



This work: Main idea

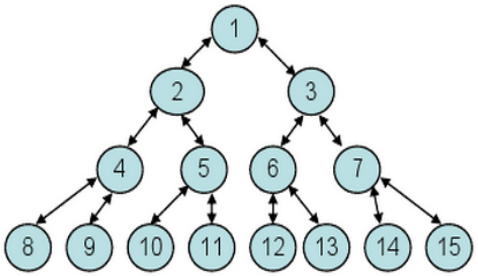
Use physical model structure to avoid using FPGA placement (Phase 2)

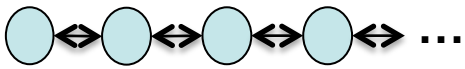
Graph embedding: Map guest graph to host graph, minim. max wire length

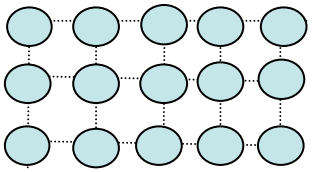


Phase 2 – Map virtual PEs to physical PEs

Guest







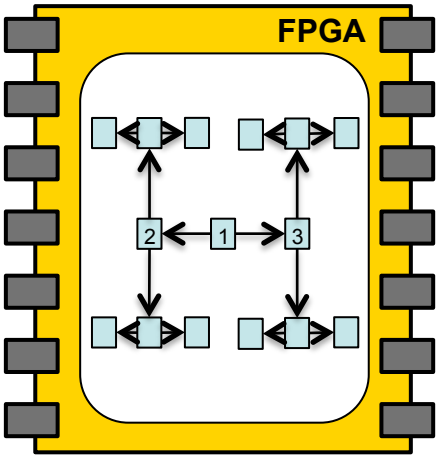
Embedding algorithm

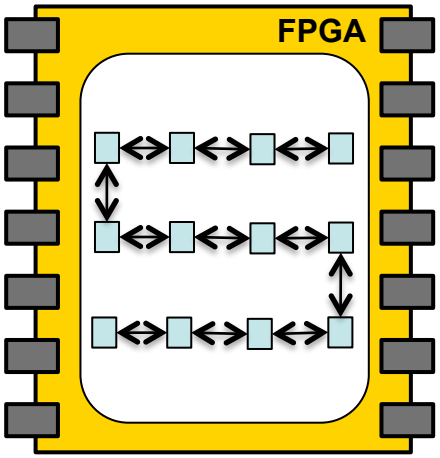
H-tree embedding

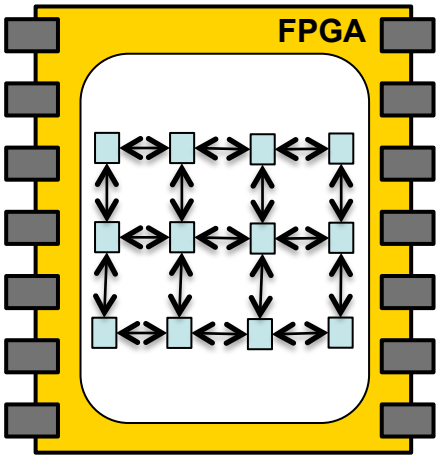
Linear embedding

Direct map embedding

Host

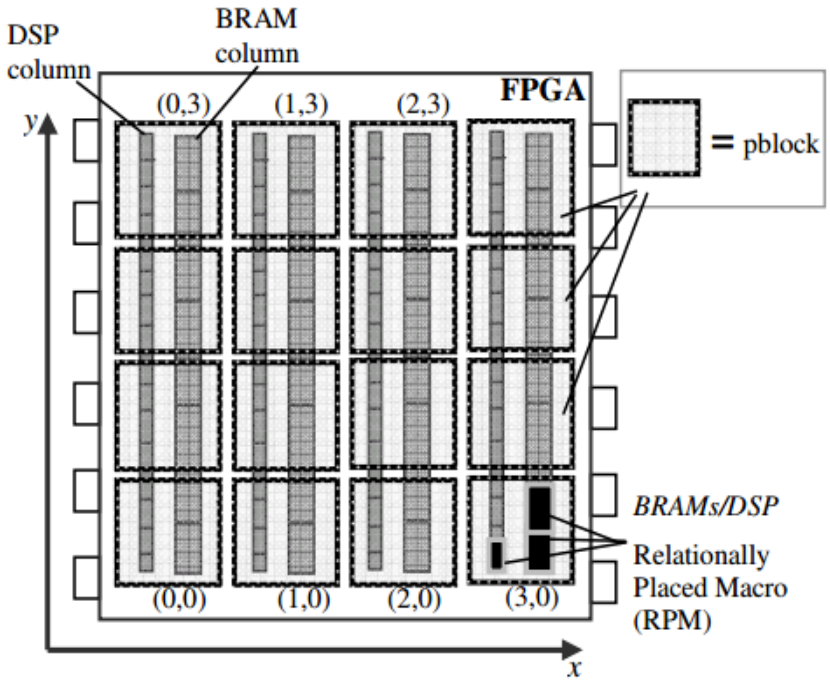






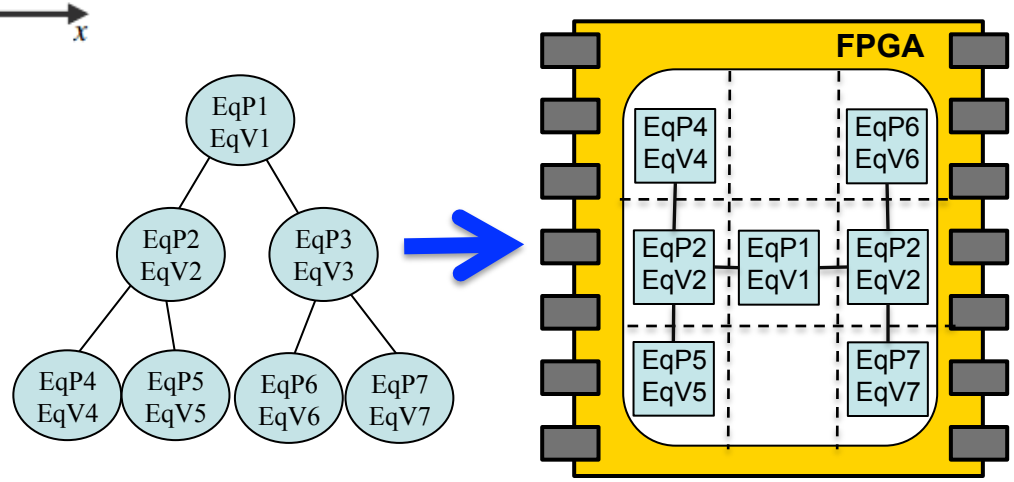
[1] Zienicke, P. 1990. Embeddings of Treelike Graphs into 2-Dimensional Meshes. (WG '90).
[2] Aleliunas, R., and Rosenberg, A.L. 1982. On Embedding Rectangular Grids in Square Grids. (Computers '82).
[3] Berman, F., and Snyder, L. 1987. On mapping parallel algorithms into parallel architectures, (PDC, '87).

2D grid of physical PEs



Bypass FPGA placement

(Phase 1: May require "graph folding" first to reduce #PEs)

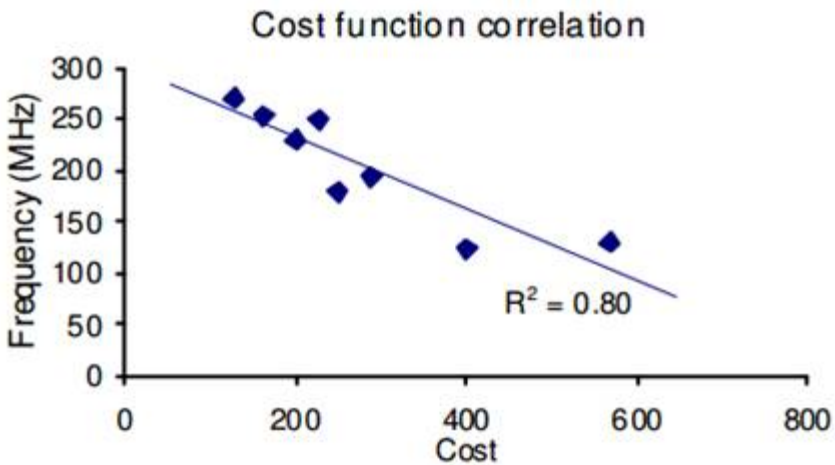


Compare/backup: Simulated annealing

Cost function:

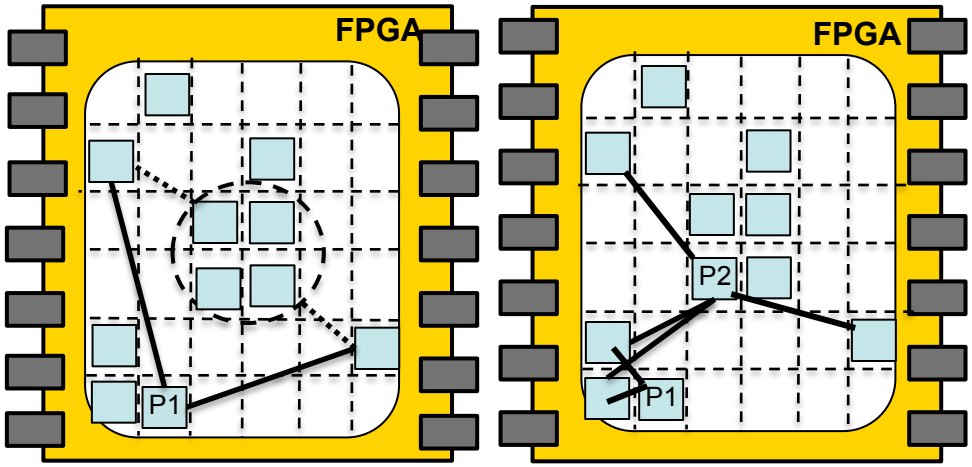
$C = w1*sum + w2*max + w3*gaps$

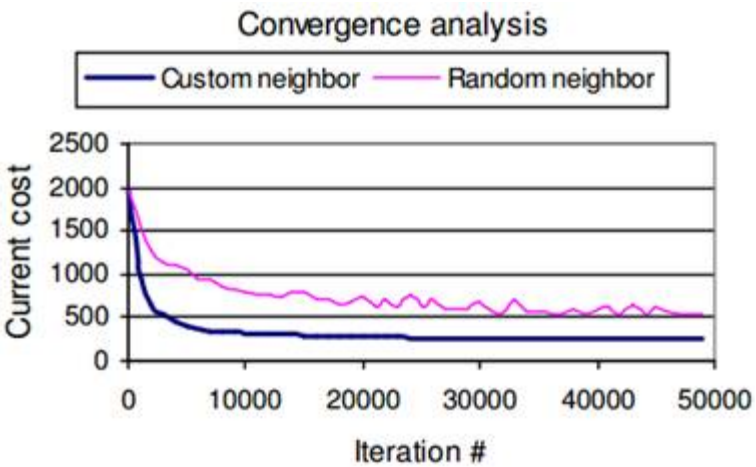
Sum = sum of wire distances
Max = max wire length (Euclidean dist.)
Gaps = wires across architectural features



Neighbor function:

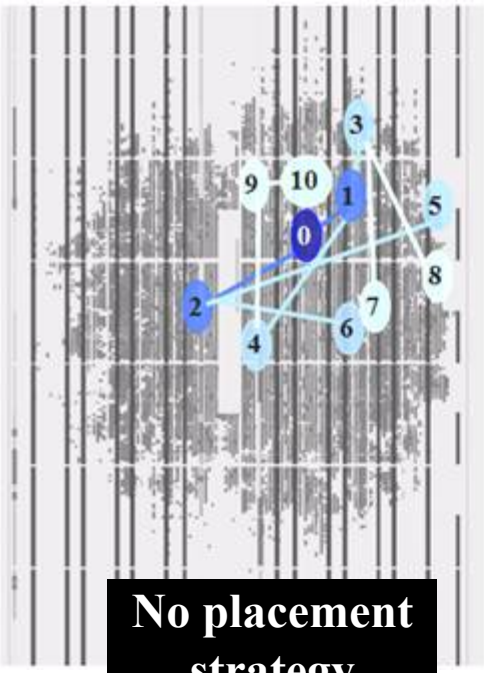
Swap PEs based on distance to neighbors



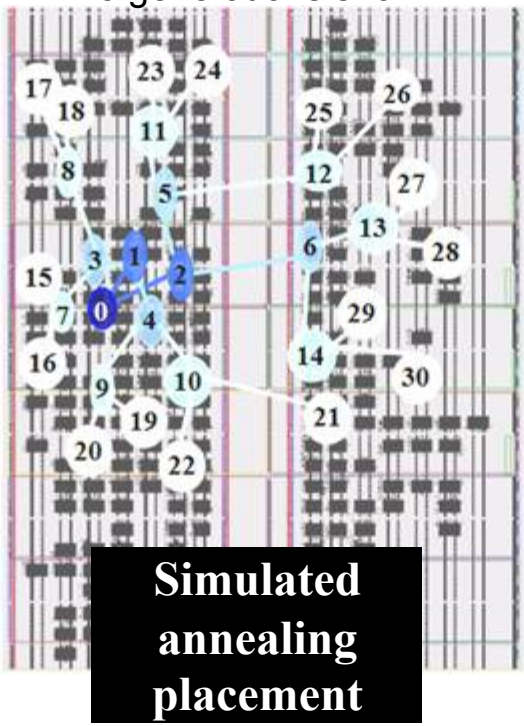


Results

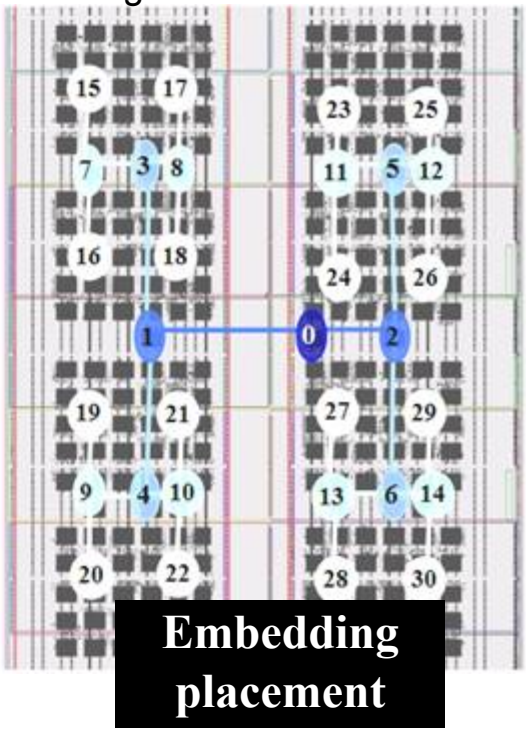
4 generations shown



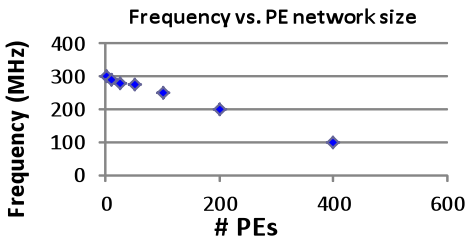
5 generations shown

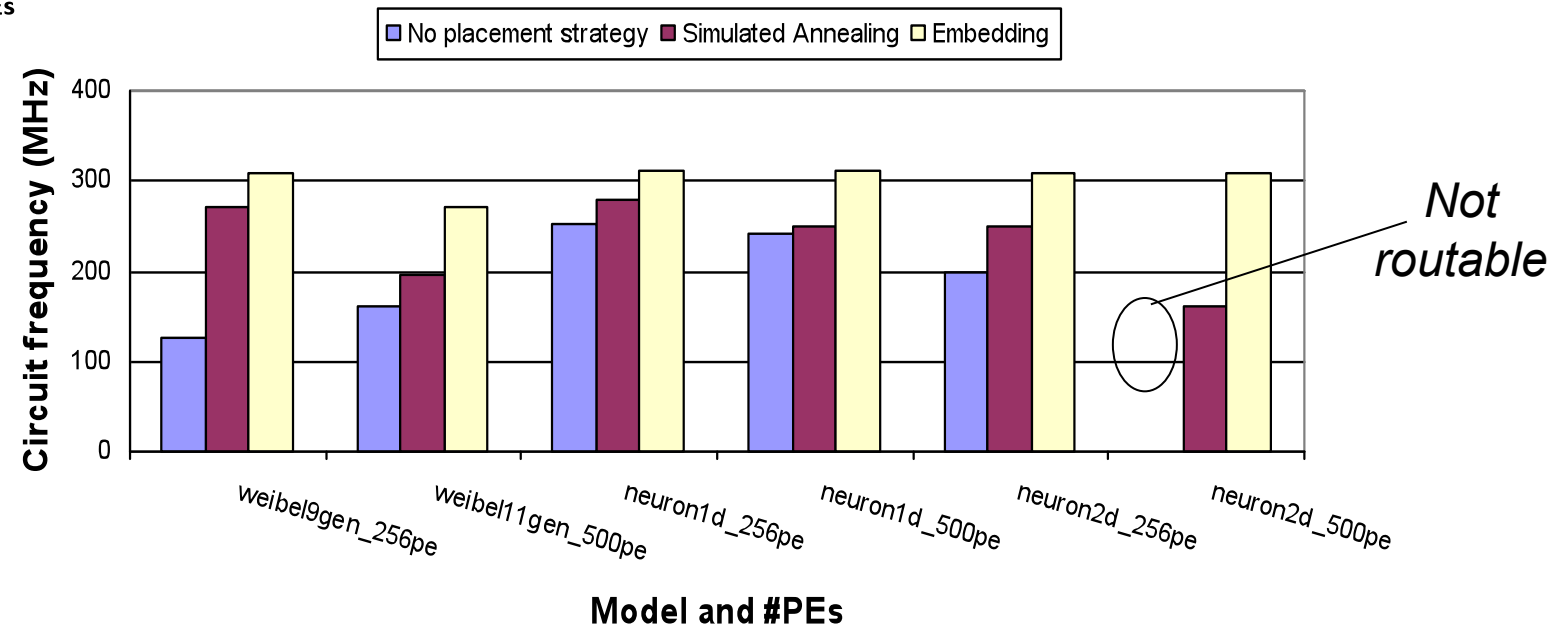


5 generations shown



Results





2D Neuron model - 256PE – Xilinx Virtex6

Strategy	# LUTS	# BRAM	#DSP	Equivalent LUTs
None	58362	512	256	306682
SA	58567	512	256	306887
Embed	58569	512	256	306889

No impact on size

Strategy	Total power (mW)	Dynamic power (mW)	Static power (mW)
None	15525	8744	6481
SA	16604	10013	6590
Embed	19859	12999	6859

20% more power

Using Xilinx XPower Analyzer

Results/Conclusions

- Graph emb. gives additional speedup
  - FPGAs: Fastest cost-effective execution of physical models
  - http://www.youtube.com/watch?v=ThUKVhqoA3Q
- Future
  - Manycore device
  - Beyond testing CPS
    - Implement end-products

Speedup vs real-time (avg)

PC(1):	0.8x
PC(4):	3.1x
GPU:	1.6x
HLS:	3.2x
General PE:	4.9x
Grph emb(GPE):	**11.2x**
Custom PE:	6.1x
Heterog PE:	34.5x
(Grph emb(HPE):	48.5x)

Speedup / dollar

Heterog PE
- 3.0x better than PC(4)
- 4.5x better than GPU

CPU (I7-950 + Intel X58 board):	$480
GPU(GTX460 + I3-540 + H55 board):	$380
FPGA (Xilinx Virtex6 240T-2 board):	$1800

http://www.meti.com/

Questions?