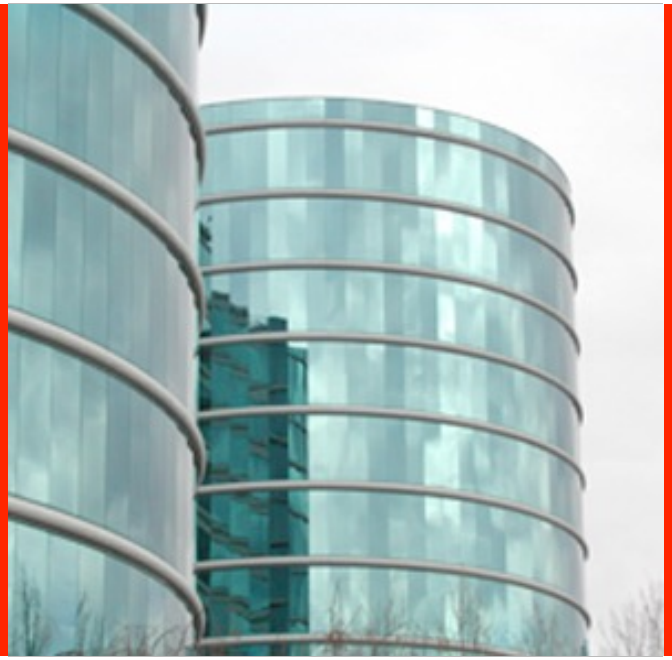


ORACLE®

The Design of a Fast Energy-Efficient Stack

Jo Ebergen
Oracle Labs
Joint work with Daniel Finchelstein, Jon Lexau, Russell Kao, David Hopkins

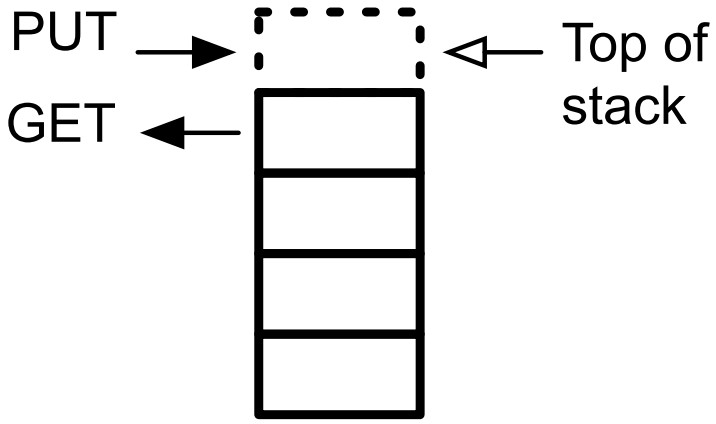
Once Upon a Time….

Stacks and Clockless Circuits

- Stack = Last-In First-Out (LIFO) Data Structure
  - Pushdown automaton
  - Like FIFOs, an often used data structure
- Why clockless circuits?
  - Clockless circuit aka asynchronous circuit
  - Speed, Energy Efficiency, …
  - “There is some meat in there”

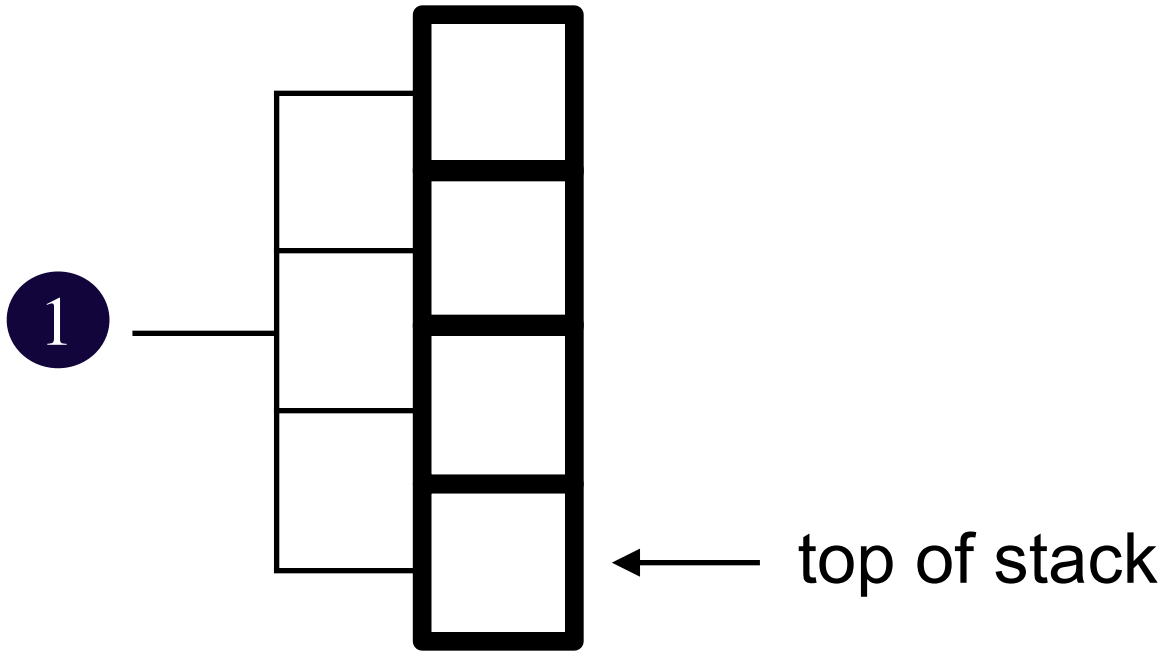
Stacks

- Have many applications
  - function call/return
  - expression evaluation
- Software implementation
  - linked lists, arrays
- Hardware implementation
  - SRAMs, shift registers
- Can we find a better implementation?
  - “better” = less delay
  - “better” = less energy
- Delay = # gate delays per PUT or GET
- Energy = # data moves per PUT or GET



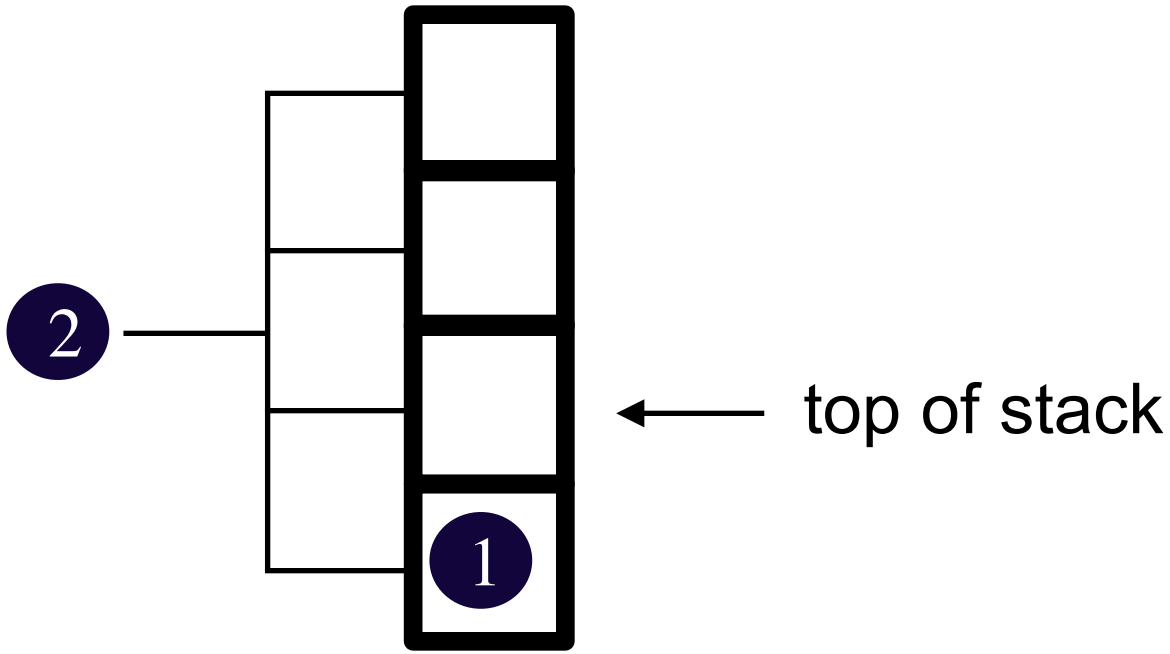
A Pointer Stack

- PUT(1)



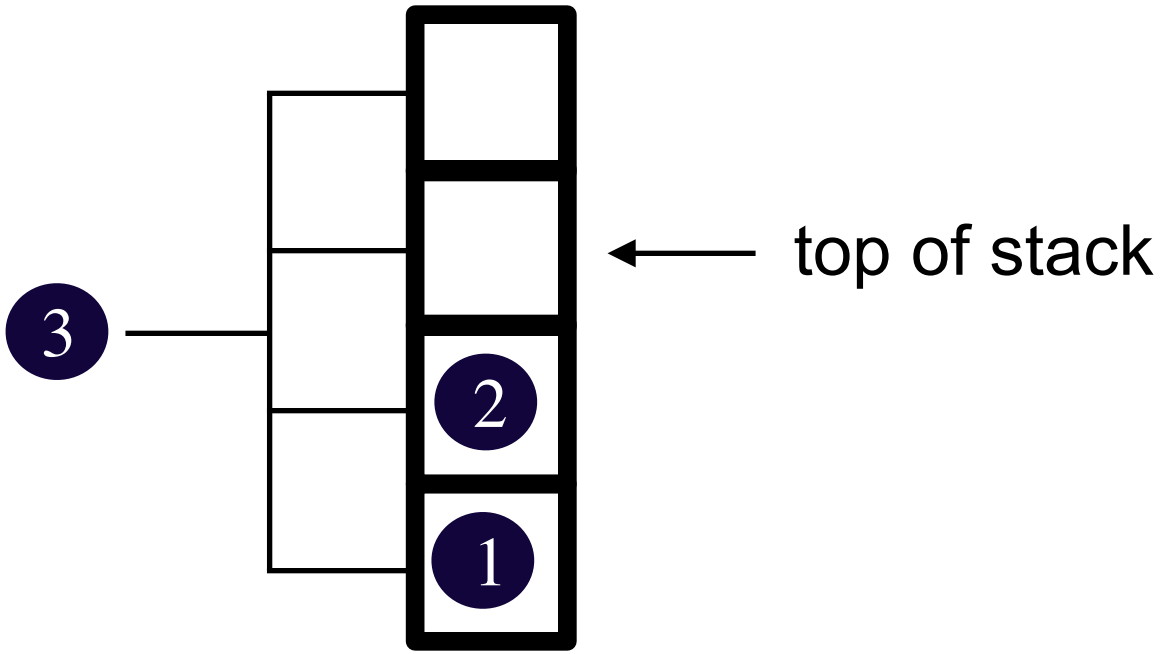
A Pointer Stack

- PUT(2)



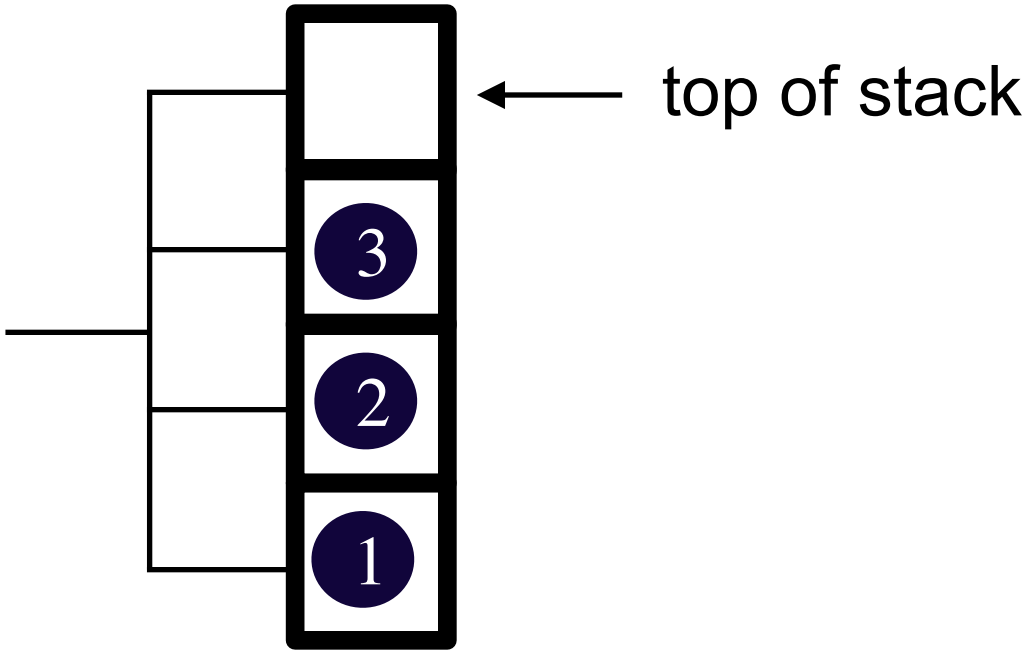
A Pointer Stack

- PUT(3)



A Pointer Stack

- GET

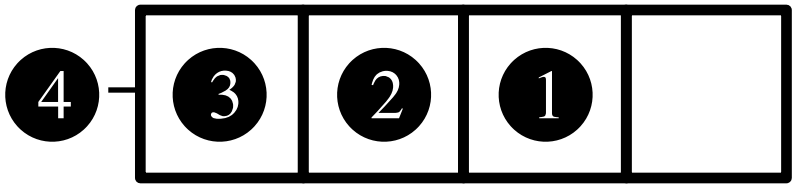


A Pointer Stack

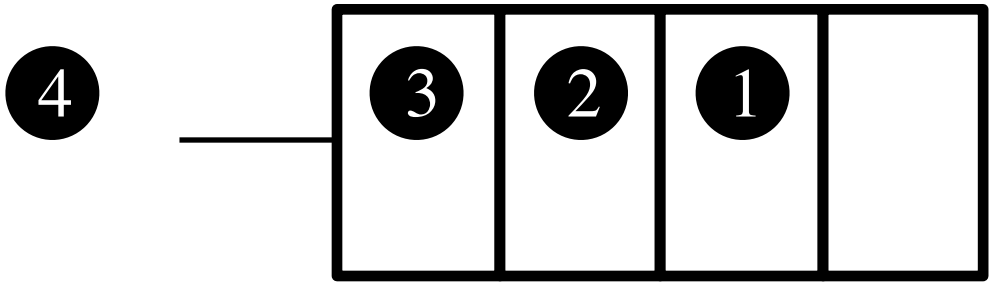
- Has only one move per PUT or GET, but ...
- Has large fan-out and fan-in
- Has large delay
  - SRAM access time > 4 ns, in TSMC 180nm
- Consumes much energy per PUT or GET
  - energy per SRAM access > 15 pJ in TSMC 180nm
- A stack doesn't need random access

A Simple Linear Stack

- Shift register, synchronized to clock



- Ripple shift register, clockless



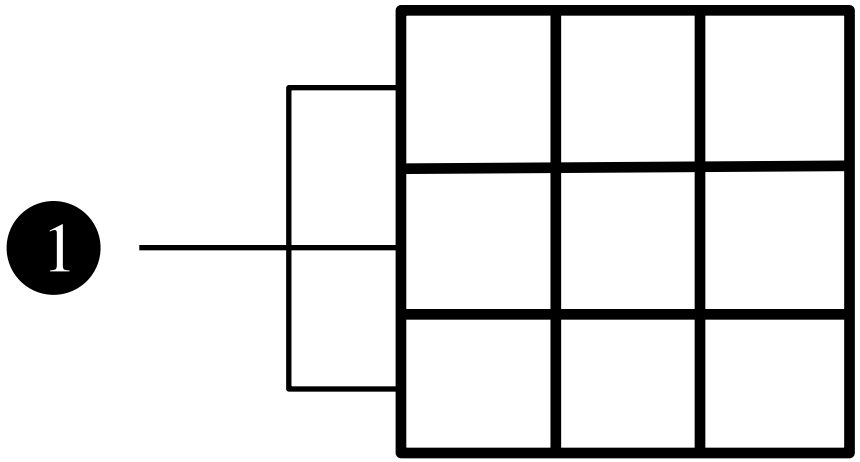
Simple Linear Stack

- Has small fan-in and fan-out
  - Enables short cycle time, but ...
- Has many moves per PUT or GET
  - Consumes much energy per PUT or GET

- Can we combine good properties of pointer and linear stack?
- Main goals:
  - Keep number of moves per PUT or GET small
  - Keep fan-in and fan-out small

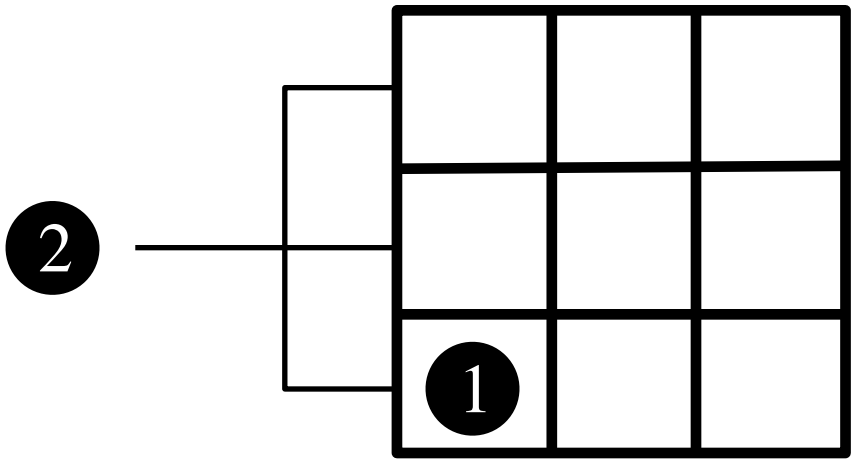
Linear + Pointer Stack

- PUT(1)



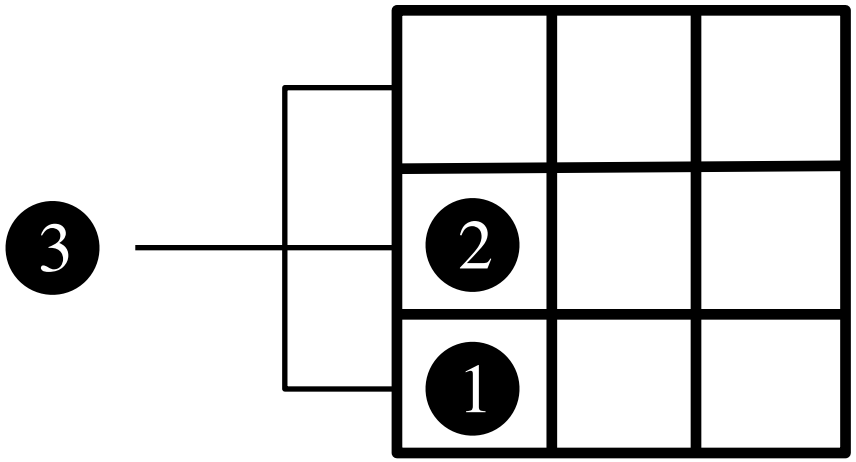
Linear + Pointer Stack

- PUT(2)



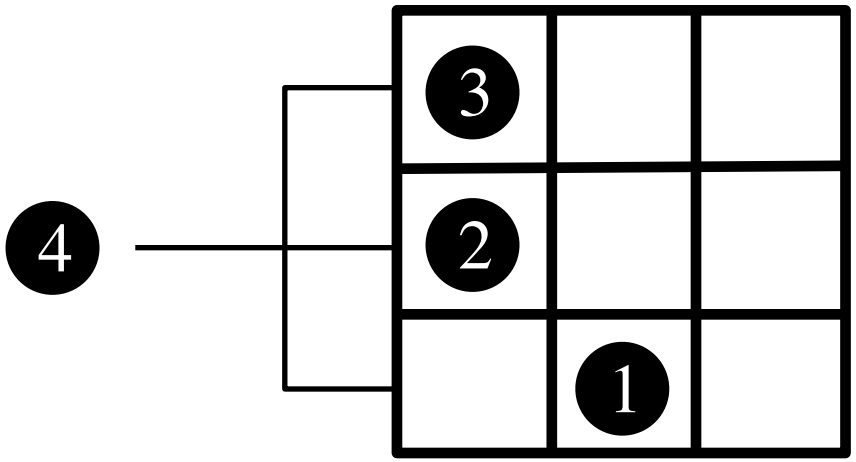
Linear + Pointer Stack

- PUT(3)



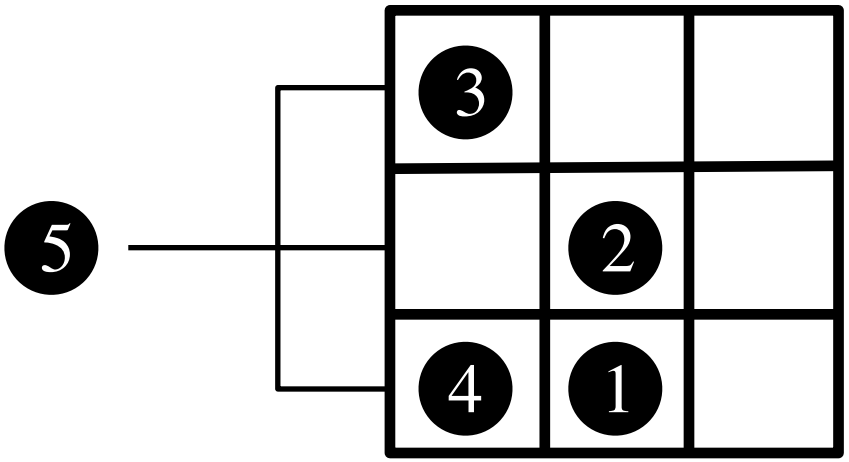
Linear + Pointer Stack

- PUT(4)



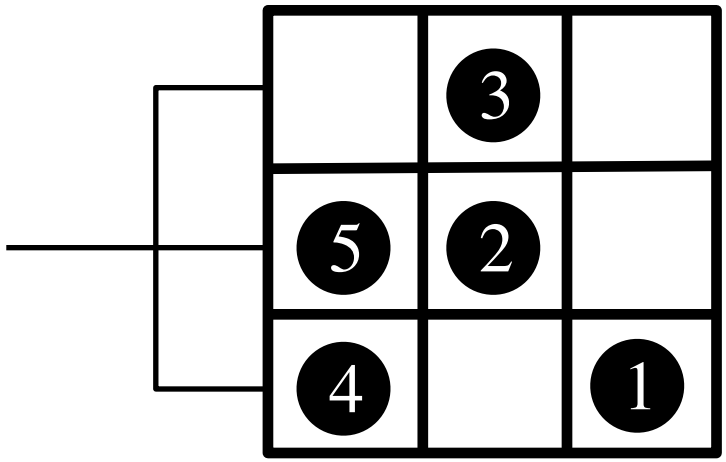
Linear + Pointer Stack

- PUT(5)



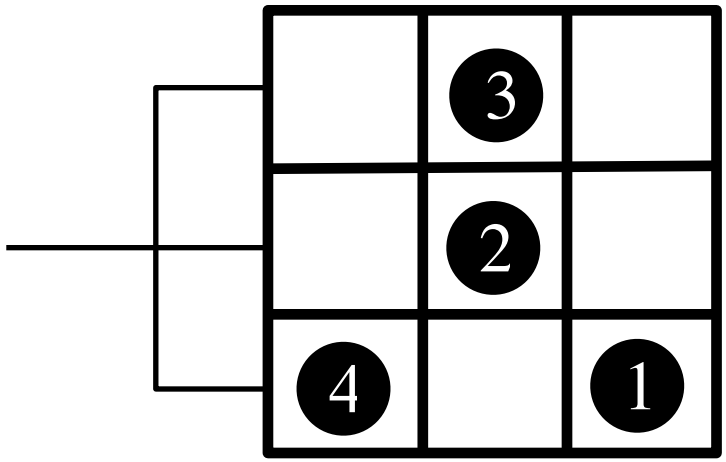
Linear + Pointer Stack

- GET



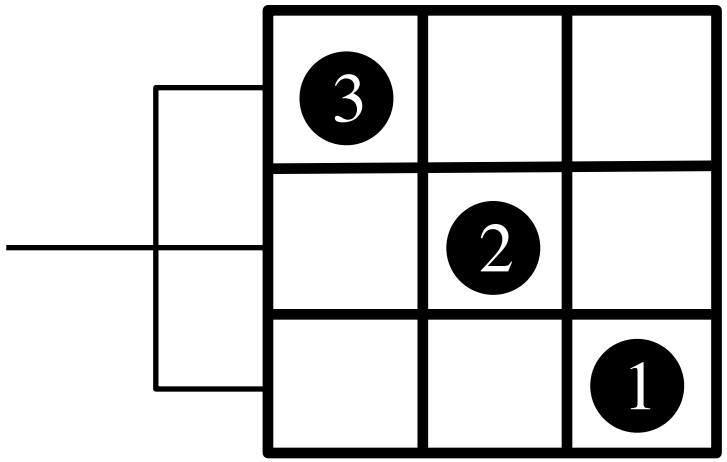
Linear + Pointer Stack

- GET



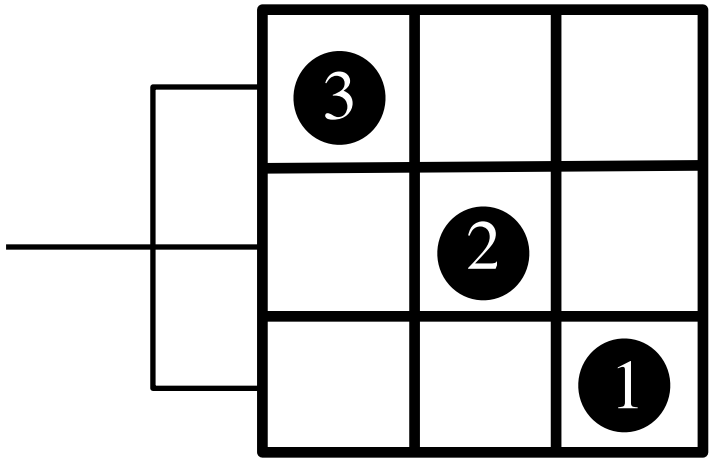
Linear + Pointer Stack

- GET



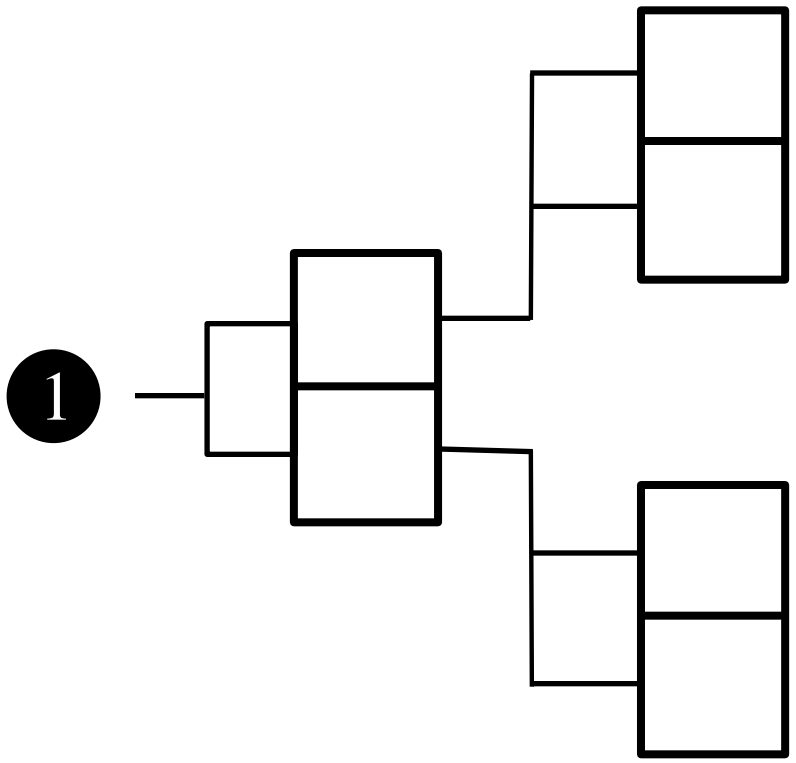
Linear and Pointer Stack

- Has small fan-in and fan-out
- Has small cycle time
- Other attractive properties
  - Moves occur only between neighboring cells
  - Only a fraction of all stack items move per PUT or GET
  - a PUT propagates further only when column-stack has 2 items
  - a GET propagates further only when column-stack has 1 item



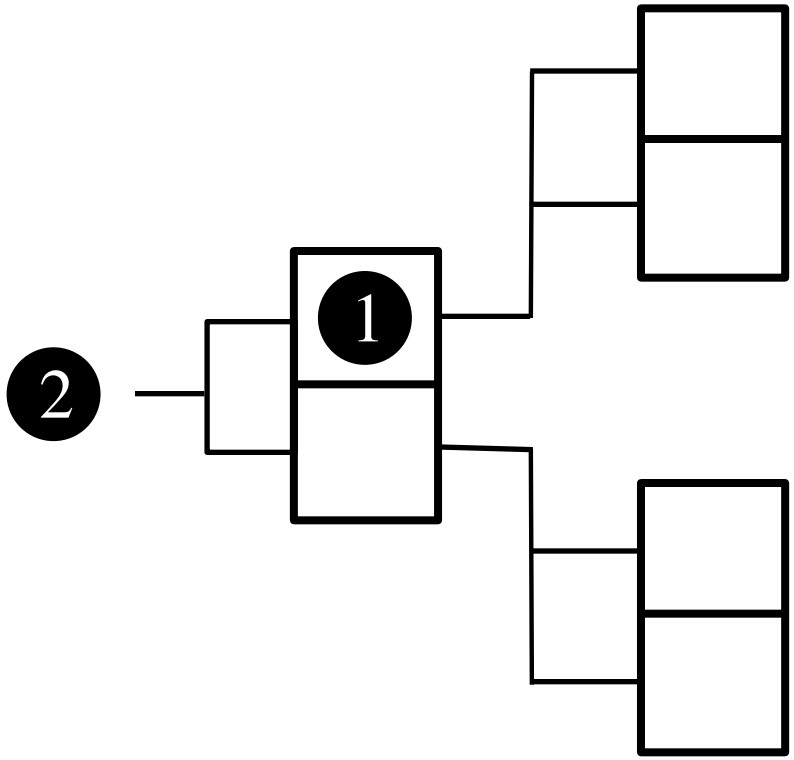
Tree Stack

- PUT(1)



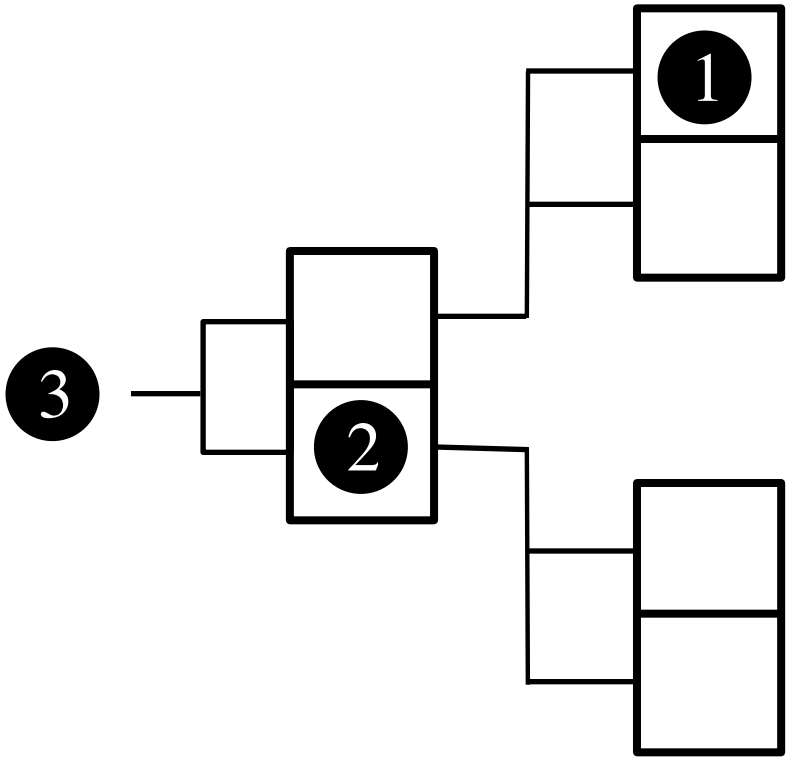
Tree Stack

- PUT(2)



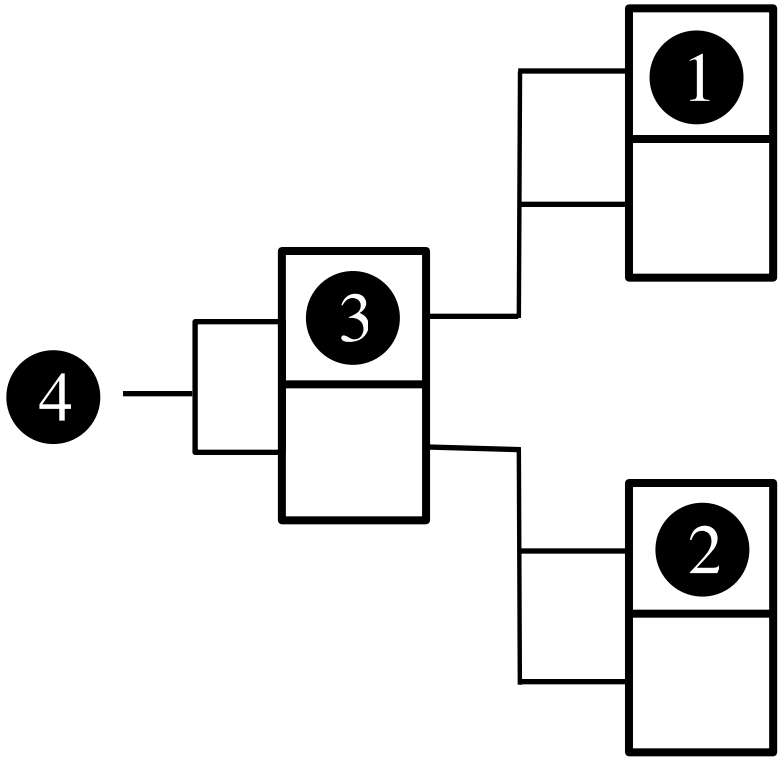
Tree Stack

- PUT(3)



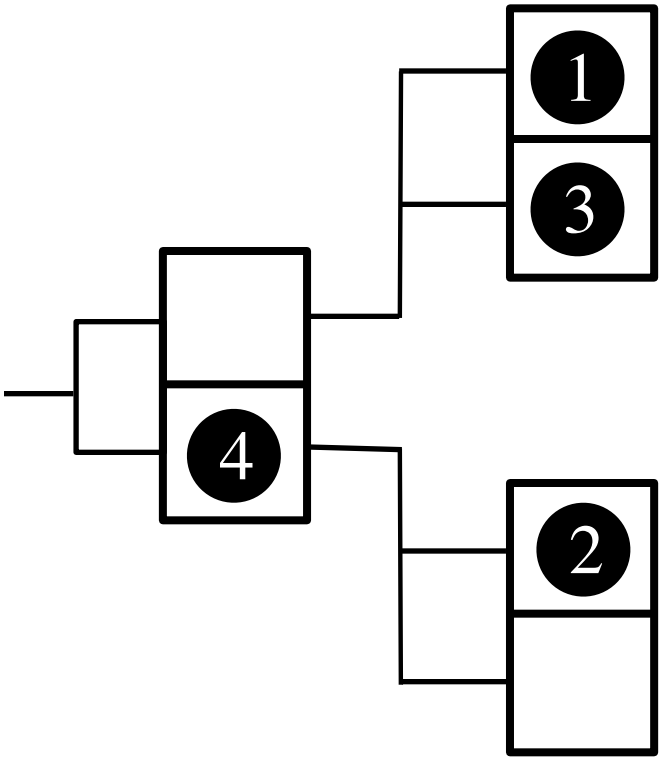
Tree Stack

- PUT(4)



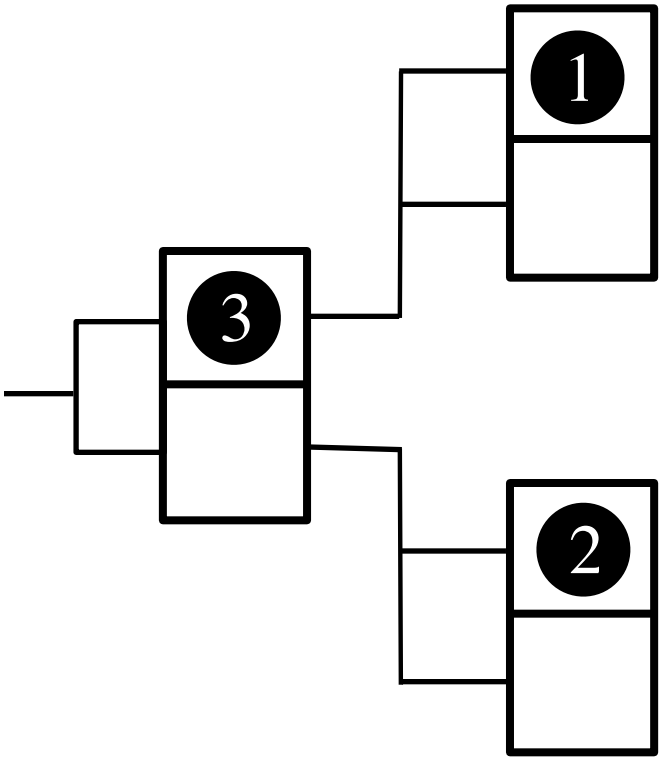
Tree Stack

- GET



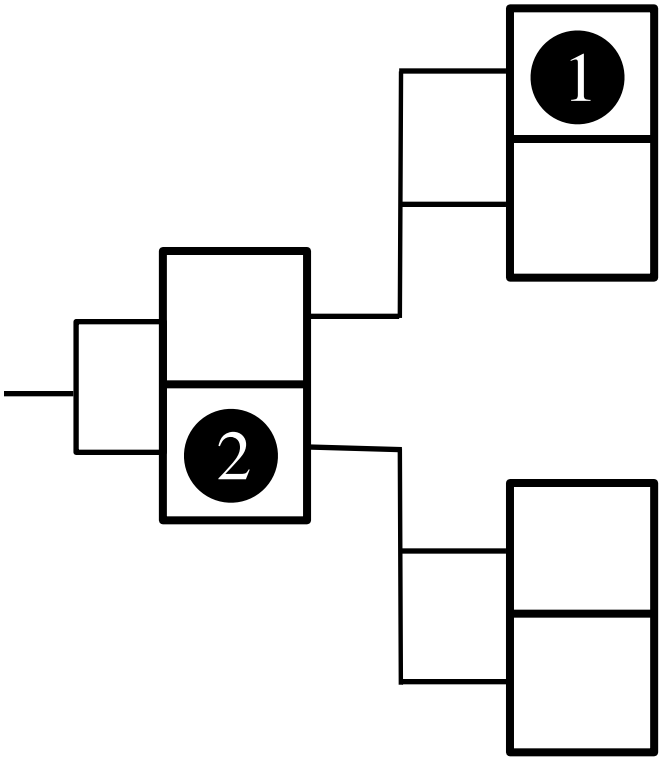
Tree Stack

- GET



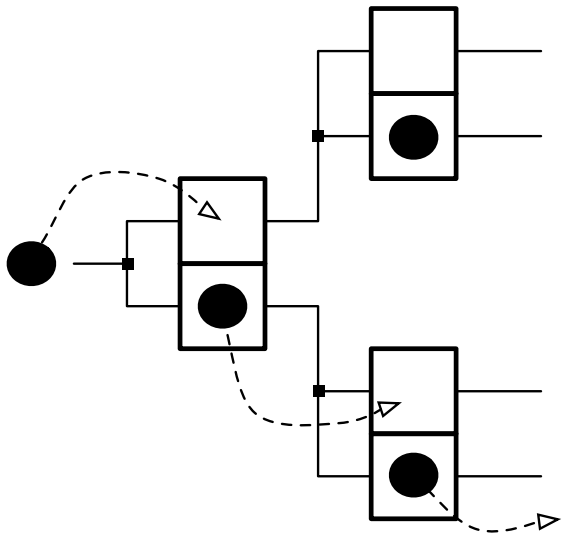
Tree Stack

- GET



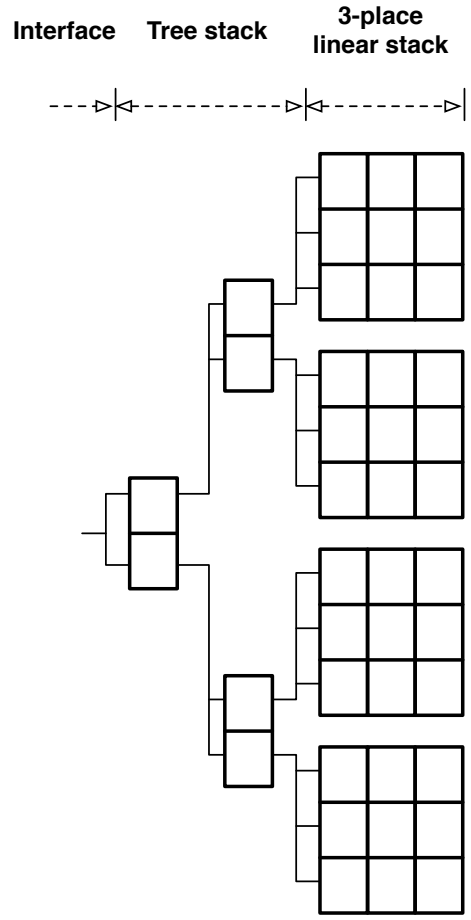
Tree Stack

- Has small fan-in and fan-out
- Has small cycle time
- Other attractive properties
  - Moves occur only between neighboring cells
  - Only a fraction of all stack items move per PUT or GET



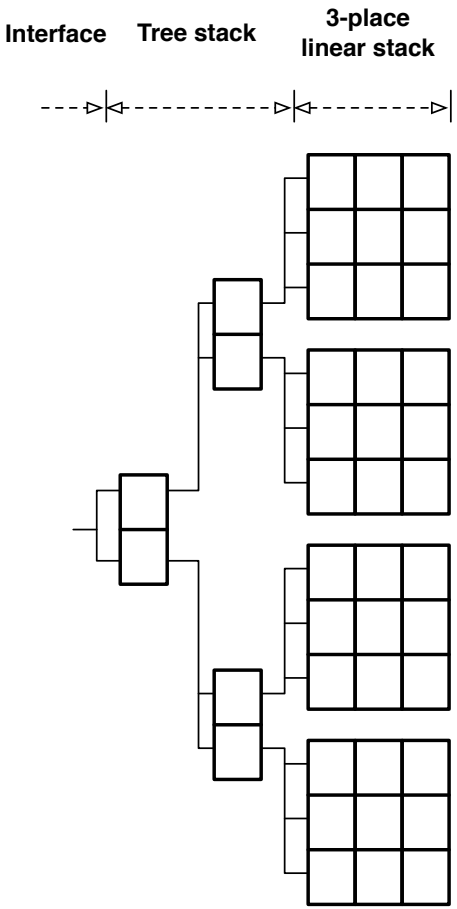
Hybrid Stack

- Combination of pointer, linear, and tree stack



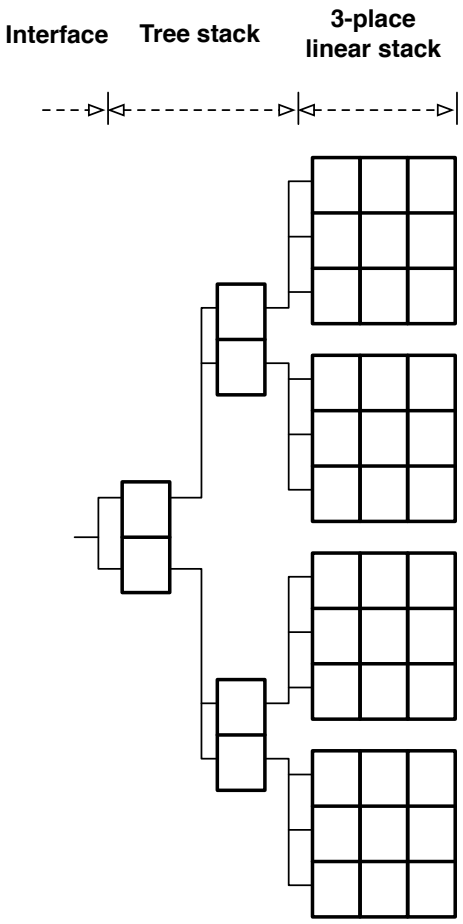
Properties of Hybrid Stack: Response Time

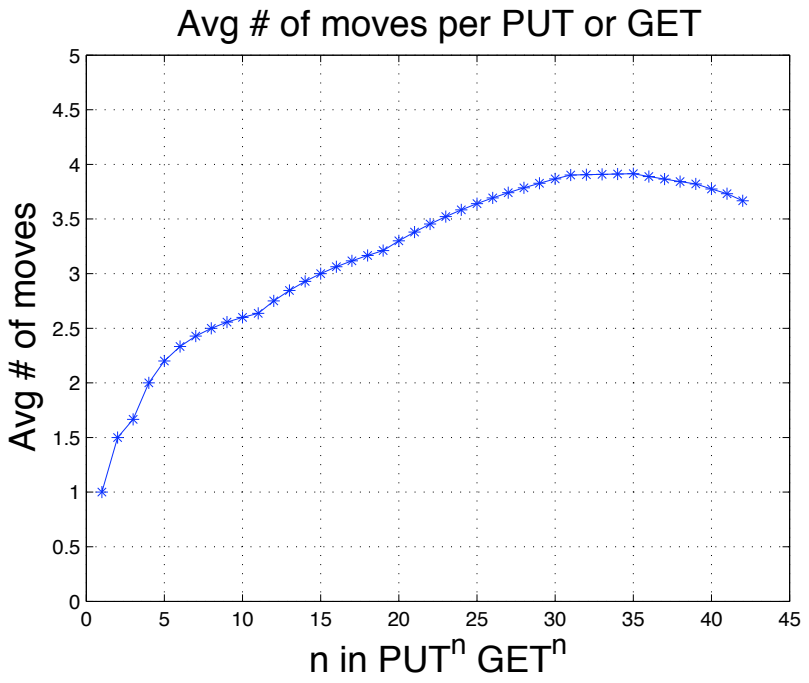
- Constant response time



Properties of Hybrid Stack: number of moves

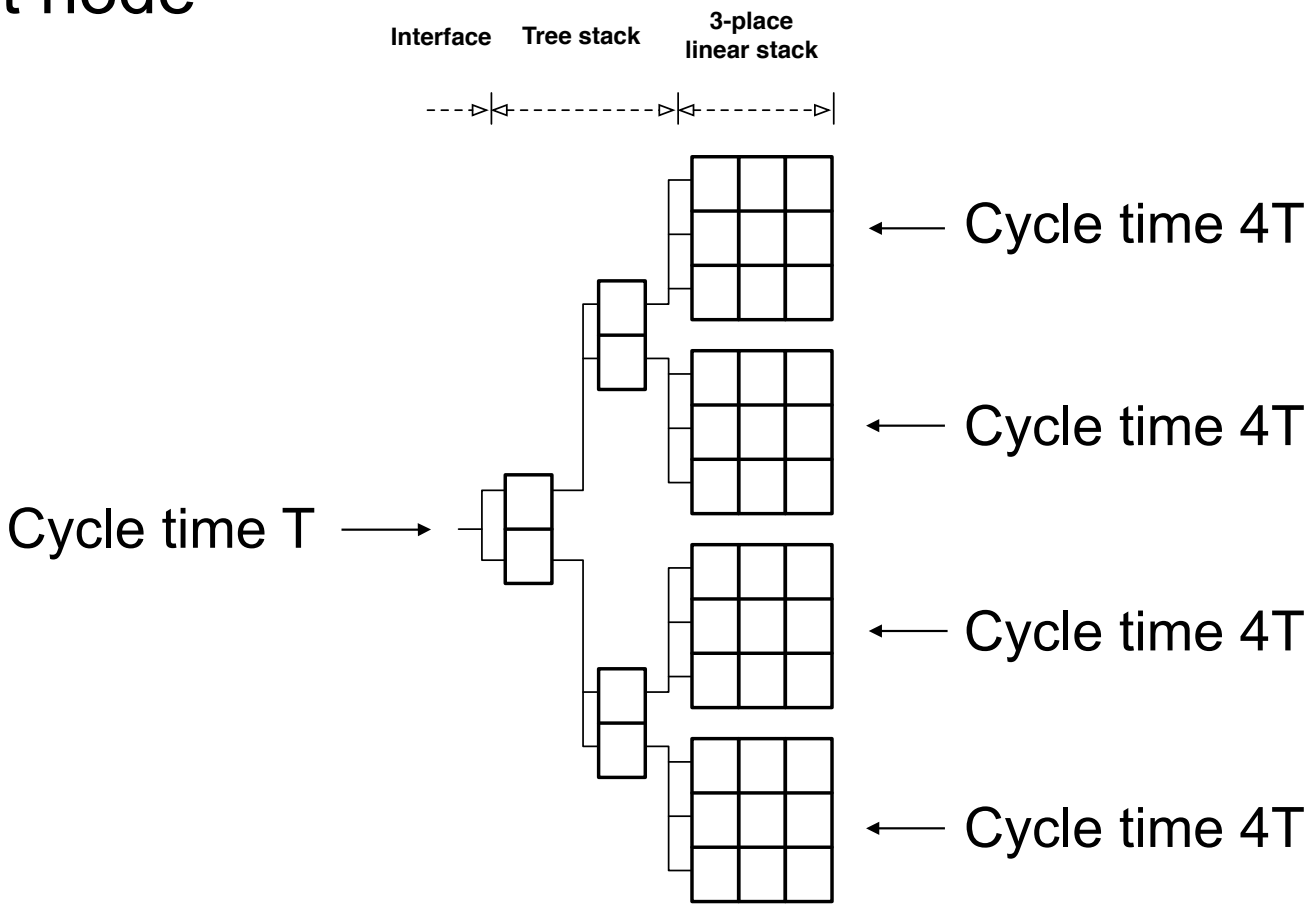
- Average number of data moves per PUT or GET



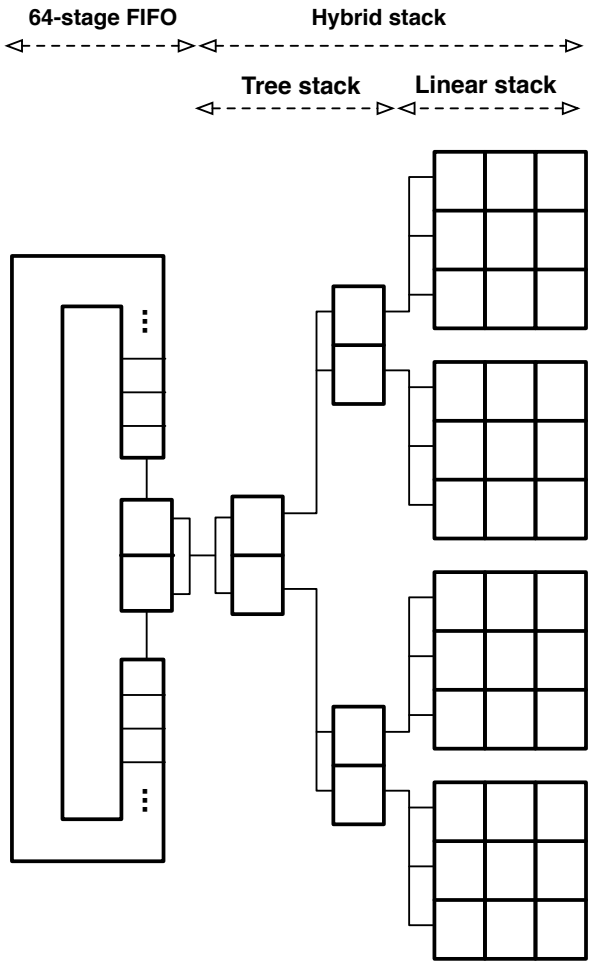


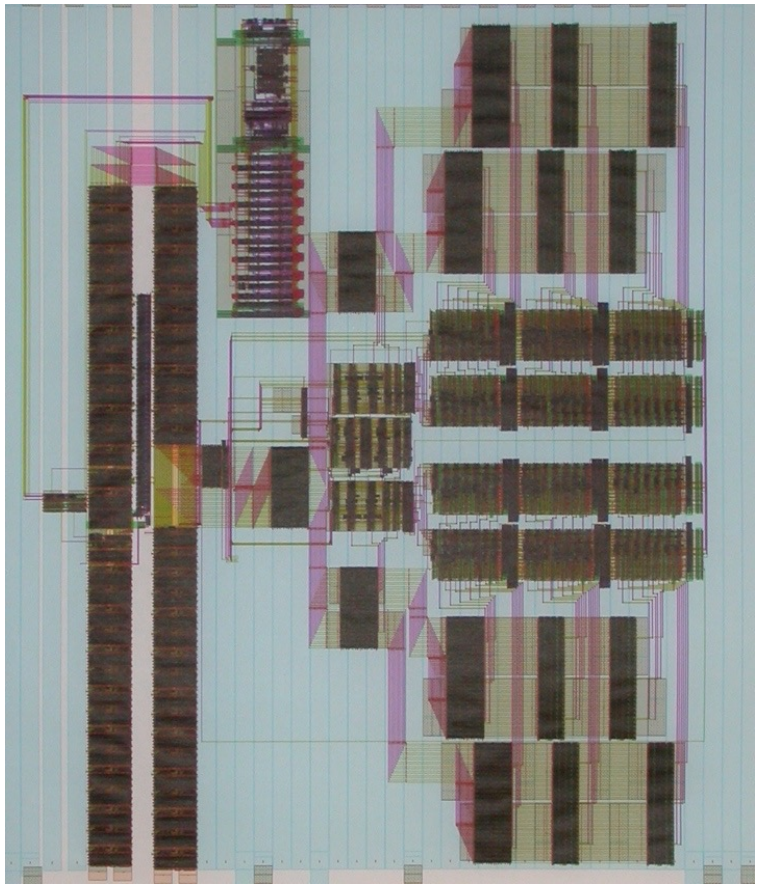
Properties of Hybrid Stack: Cycle Times

- Cycle times at leaf nodes can be 4X larger than cycle time at root node



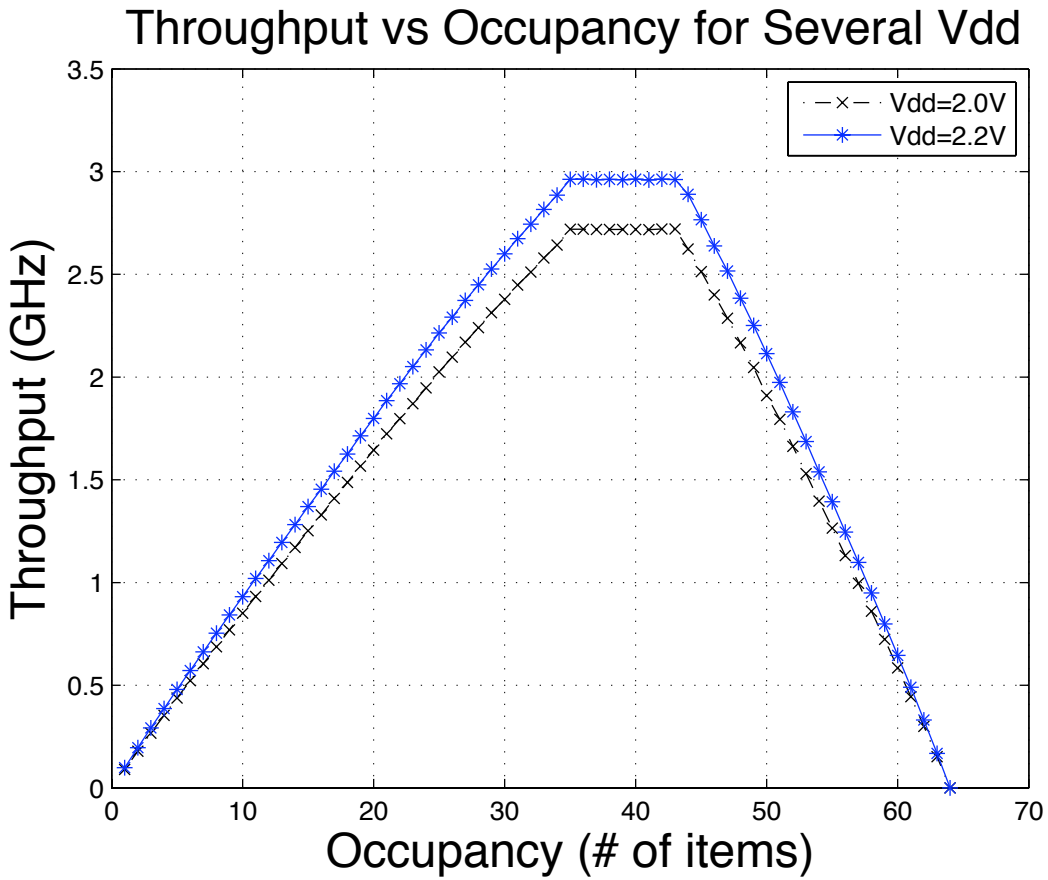
Stack Implementation





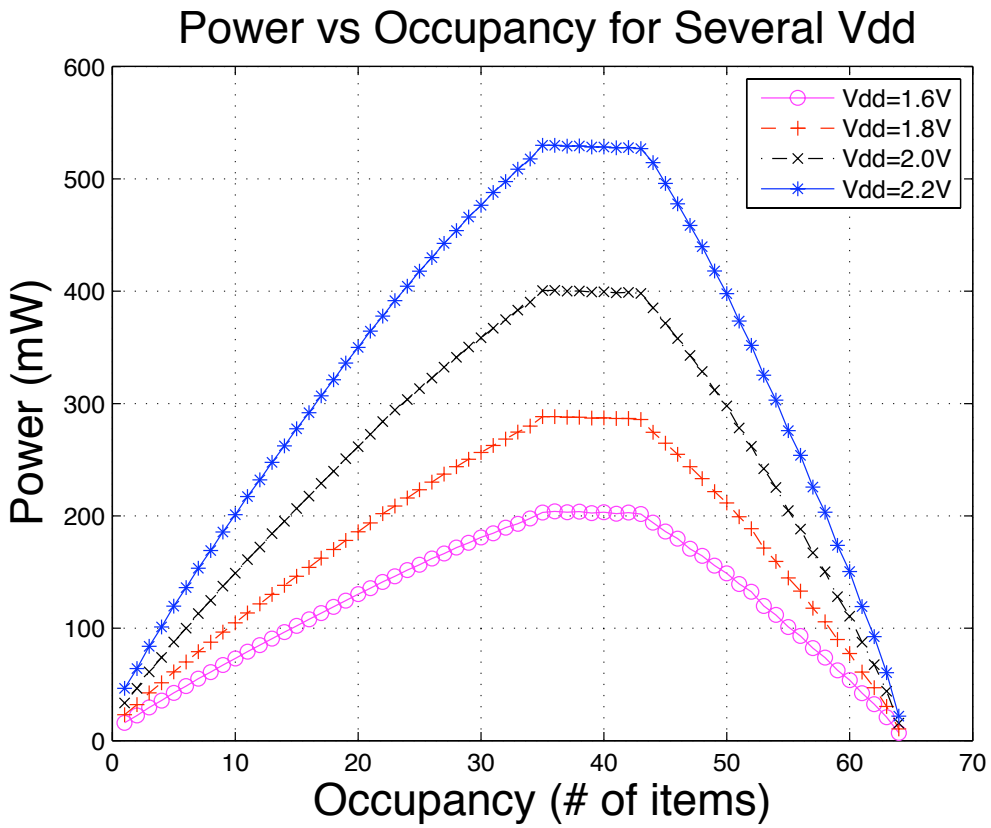
Throughput vs Occupancy

- 2.75GHz at 2.0V



Power Consumption vs Occupancy

- Power consumption is proportional to number of moves



Concluding Remarks

- A stack implementation
  - Cycle time of 6 gate delays, (368ps at 2V or 2.72GHz in 180nm)
  - Cycle time is independent of # items or data path width
  - Average number of moves per PUT or GET grows (very) slowly
- Dynamic energy per move is small (6-8pJ)
  - and can be made smaller by using more efficient latches
- Exhibits much concurrency
- A problem for you
  - How does avg number of moves depend on input sequences of PUTs and GETs?
  - What is lower bound for avg number of moves for a stack implementation given bounded fan-out?

More Reading

- "An Evaluation of Asynchronous Stacks," IEEE Design and Test, Sept/Oct 2011, pp 52--61.
- A Fast and Energy-Efficient Stack, "Proc of 10th IEEE Int'l Symp. Advanced Research in Asynchronous Circuits and Systems (ASYNC)," IEEE CS Press, 2004, pp. 7--16.


ORACLE®