Reversible Processor Architectures

Source

Dr. Mike Frank

Why reversible architectures?

- What about automatic emulation algorithms?
  - E.g.: Ben73, Ben89, LMT, Frank02.
    - Transform an irreversible algorithm to an equivalent reversible one.
  - But, these do not yield the most cost-efficient reversible algorithms for all problems!
    - Finding the **best reversible algorithm** requires a *creative algorithm discovery process*!
- An optimally cost-efficient general-purpose architecture must allow the ***programmer*** to specify a custom reversible algorithm that is specific to his problem.

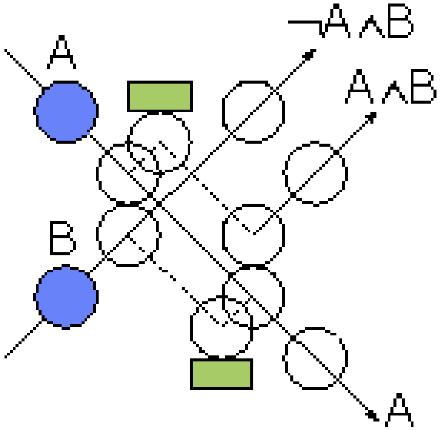
All Known Reversible Architectures

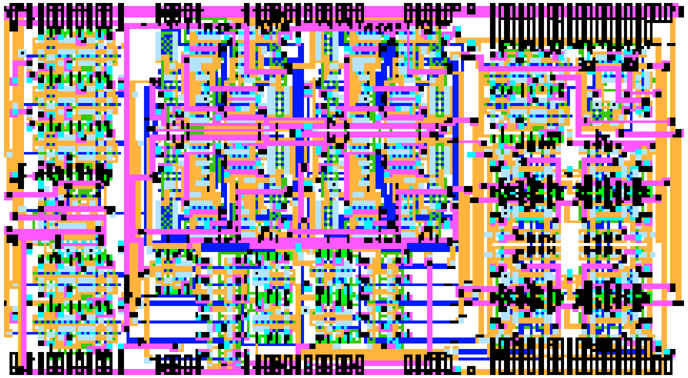
- **Ed Barton** (MIT class project, 1978)
  - Conservative logic, with garbage stack
- **Andrew Ressler** (MIT bachelor’s thesis, 1979; MIT master’s thesis, 1981)
  - Like Barton’s, but more detailed. Paired branches.
- **Henry Baker** (1992)
  - Reversible pointer automaton machine instructions.
- **J. Storrs “JoSH” Hall** (1994)
  - Retractile-cascade-based PDP-10-like architecture.
- **Carlin Vieri** (MIT master’s thesis, 1995)
  - Early Pendulum ISA, irrev. impl., full VHDL detail.
- **Frank & Rixner** (MIT class project, 1996)
  - Tick: VLSI schematics & layout of Pendulum subset, w. paired branches
- **Frank & Love** (MIT class project, 1996)
  - FlatTop: Adiabatic VLSI impl. of programmable reversible gate array
- **Vieri** (MIT Ph.D. thesis, 1999)
  - Fully adiabatic VLSI implementation of Pendulum w. paired branches

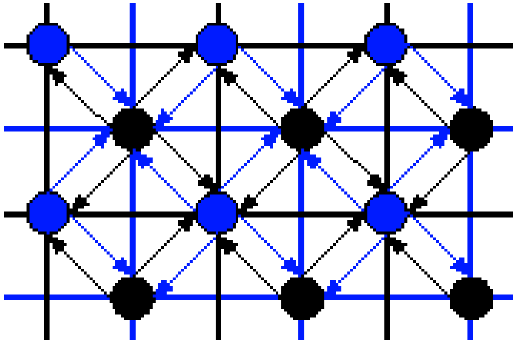
Not much work reported since then, Dr.Frank?

Reversible Programmable Gate-Array Architectures

FlatTop reversible mesh processor

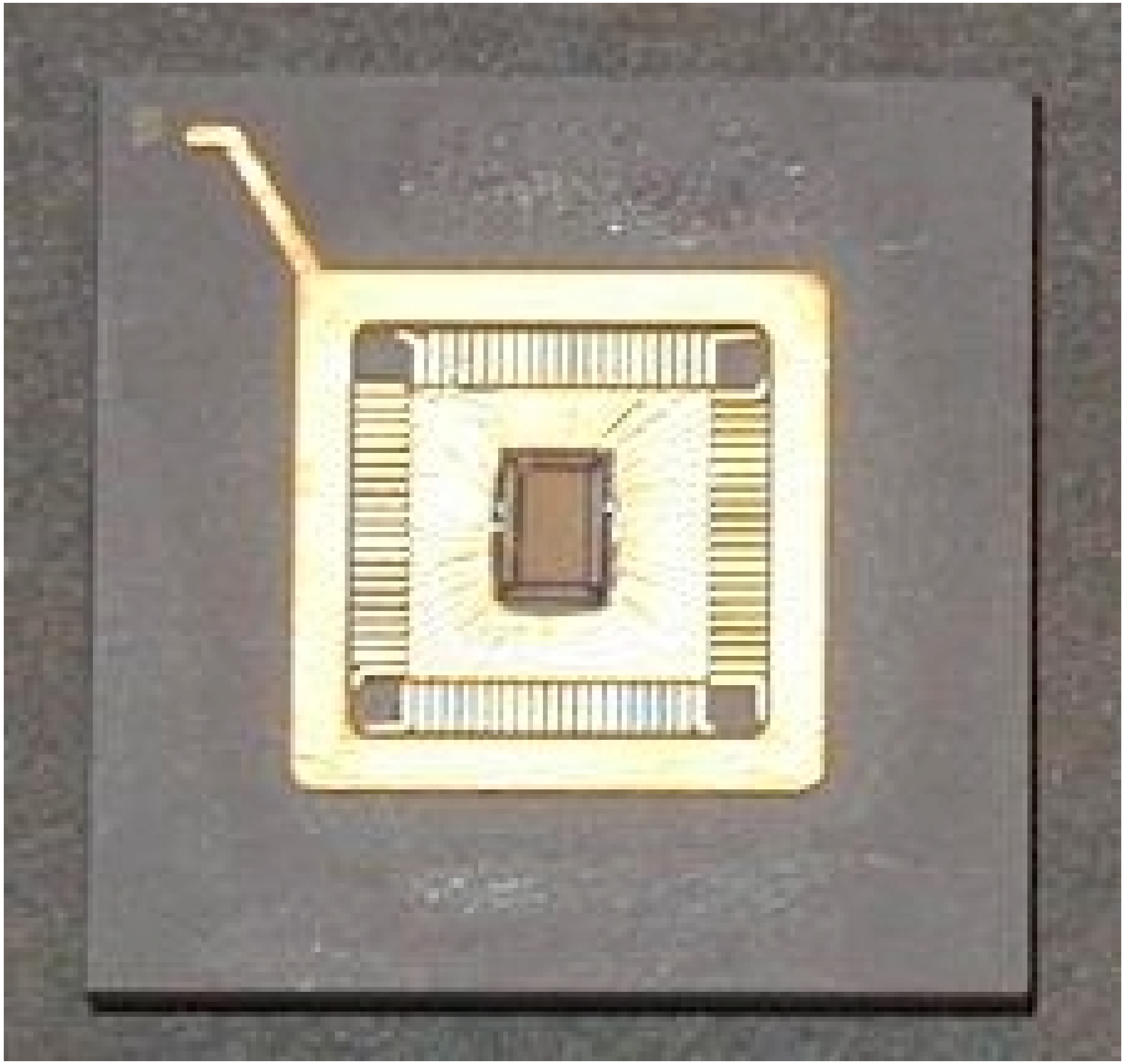




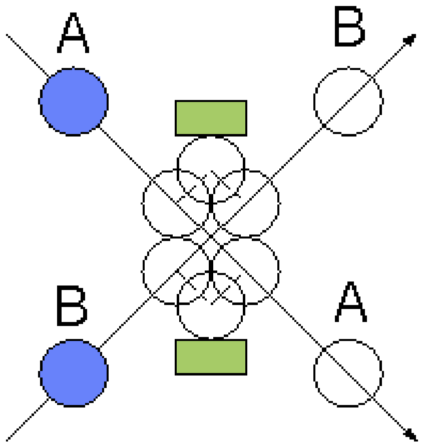


- FPGA-like; simulates any reversible circuit.
- Encodes Fredkin “Billiard Ball Model” of computation
- ~ 300 transistors / processing element.
- Est. min. room-T diss.: $\sim 2000\times$ less than irrev.
- Chips w. 20×20 array just back from fab. (as of May ‘99)

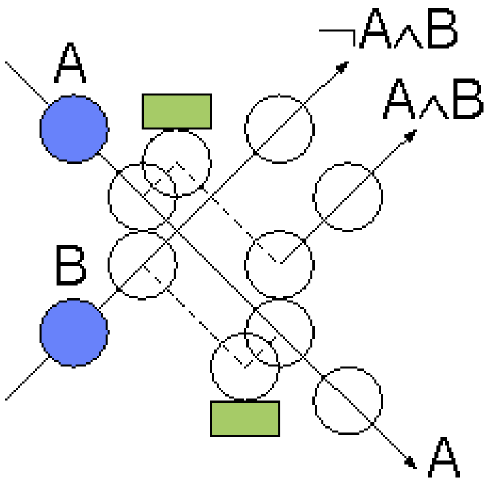
Photo of packaged FlatTop chip



The Billiard Ball Model of computation



Crossover Gate

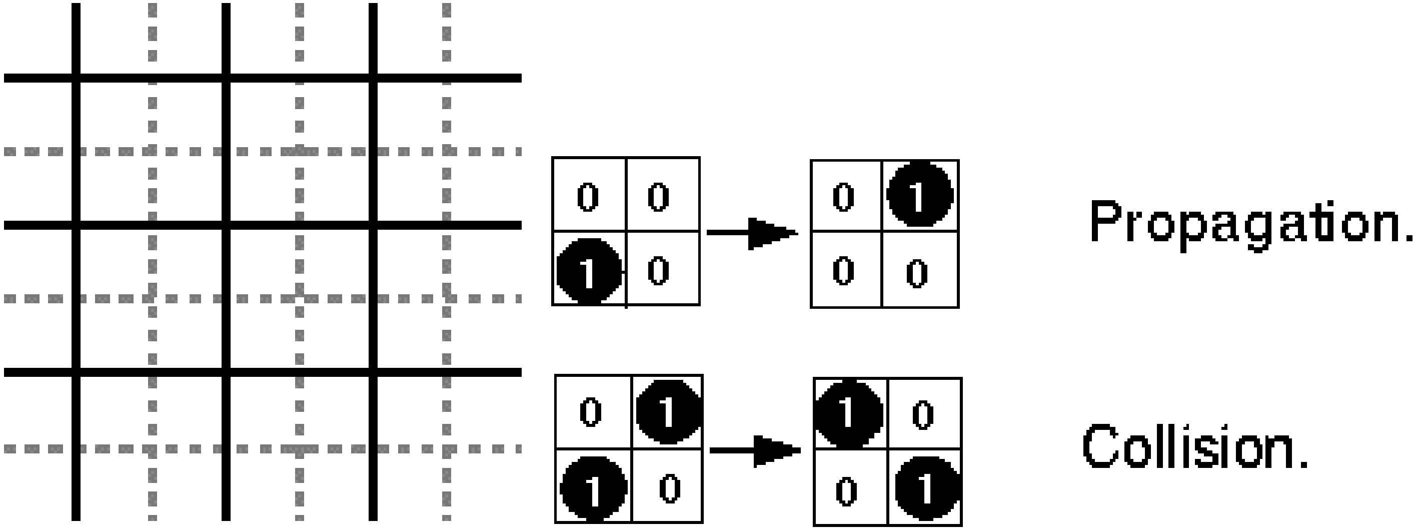


Feynman Gate

Idealized "billiard balls" bounce of walls & each other in digitally-precise trajectories. (Fredkin & Toffoli, '82)

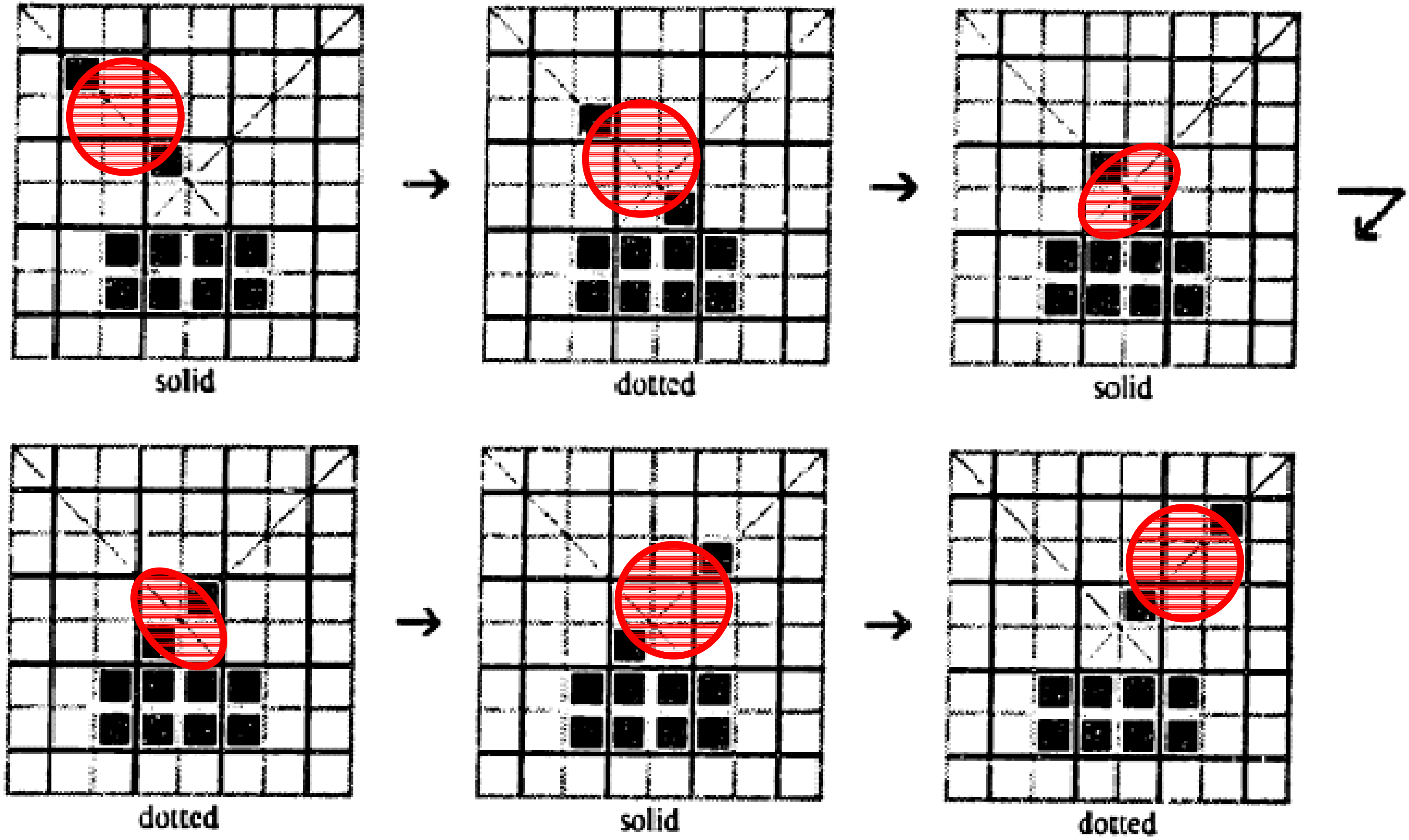
- Extremely simple.
- Reversible.
- Universal.
  - Efficiently simulates arbitrary 2-D reversible cellular automata or reversible circuits.
  - Extensible to 3-D.

The BBM cellular automaton (Margolus '88)



- Updated in 2 alternating, overlapping partitionings of array into 2×2 blocks of cells. ("Margolus neighborhood.")
- Each block updated independently & reversibly.

A Bouncing BBMCA “Ball”



A BBMCA Fredkin Gate

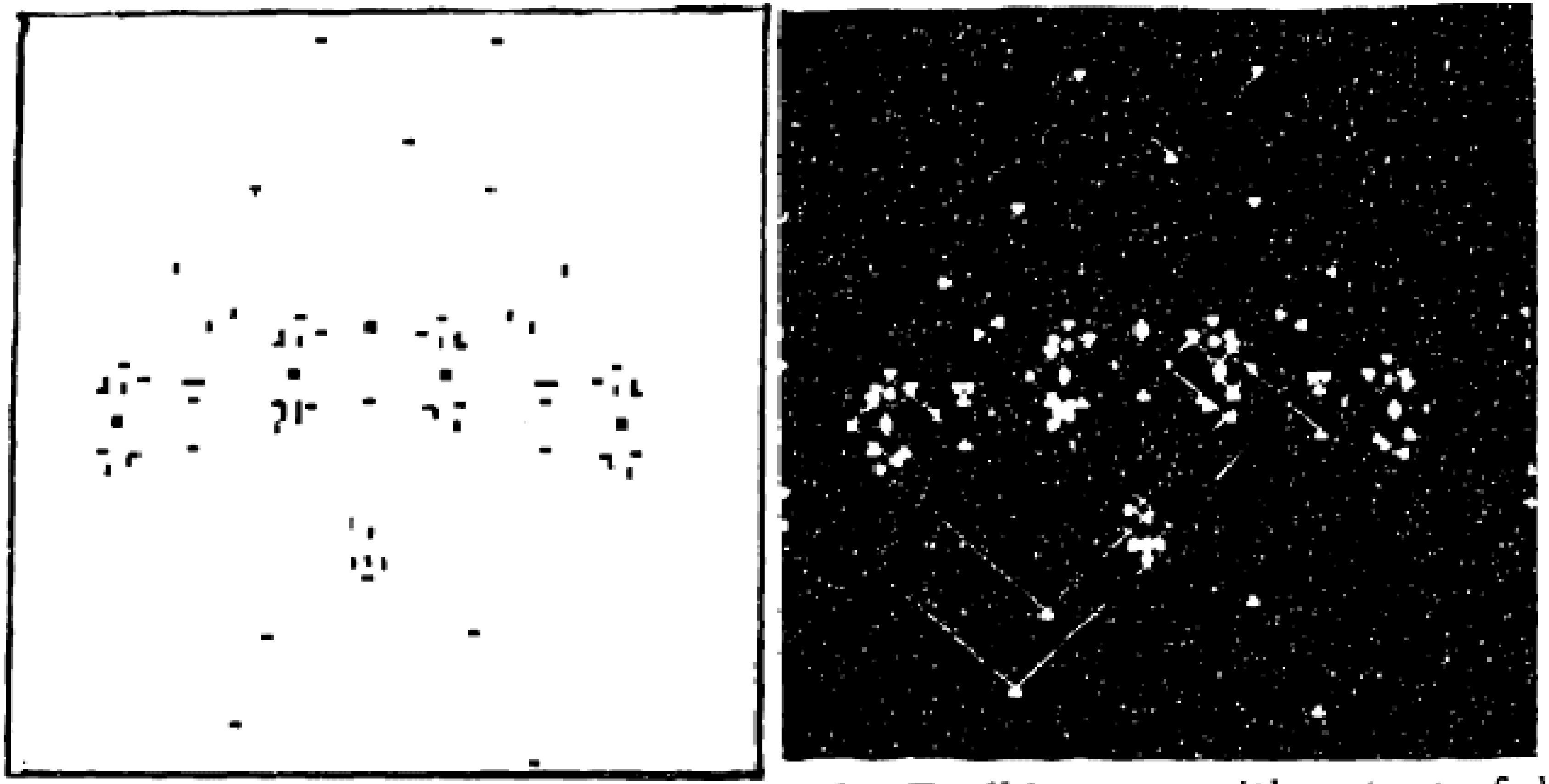
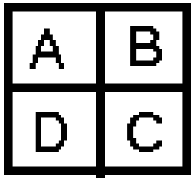


Figure 2.8: BBMCA implementation of a Fredkin gate, with outputs fed back to inputs via signal paths with co-prime lengths, to perform a pseudo-random permutation.

Boolean logic form of BBMCA update rule



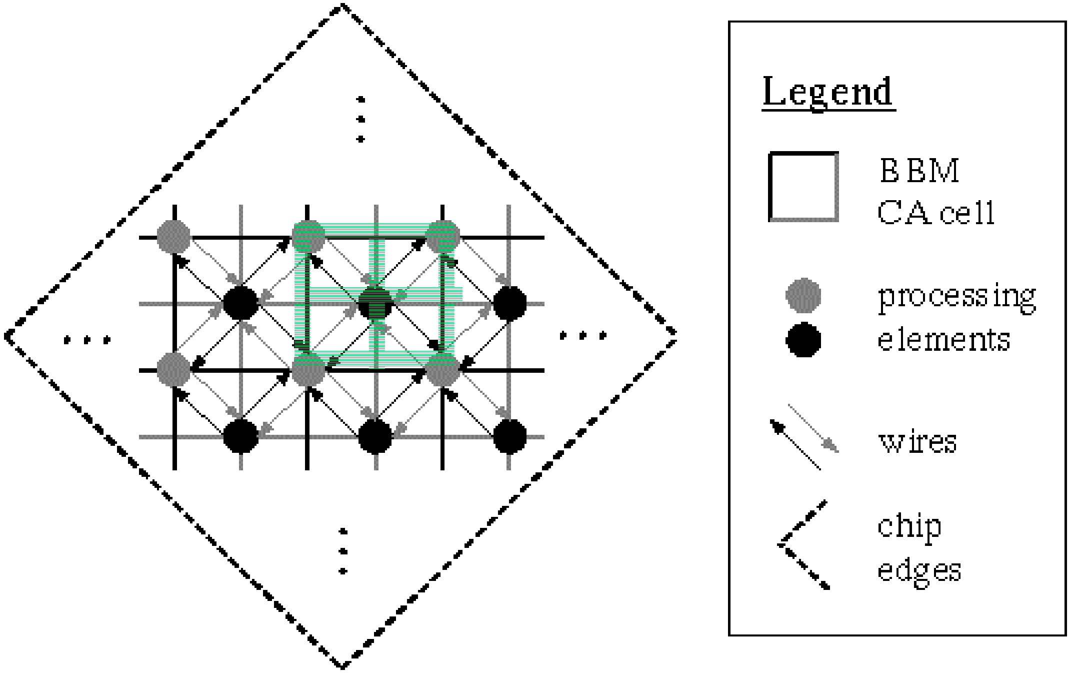
$$S = (A + C)\,(B + D)$$

$$A' = S\,A + \overline{S}\,\overline{A}\,(C + B\,D)$$

... and similarly for B, C, D

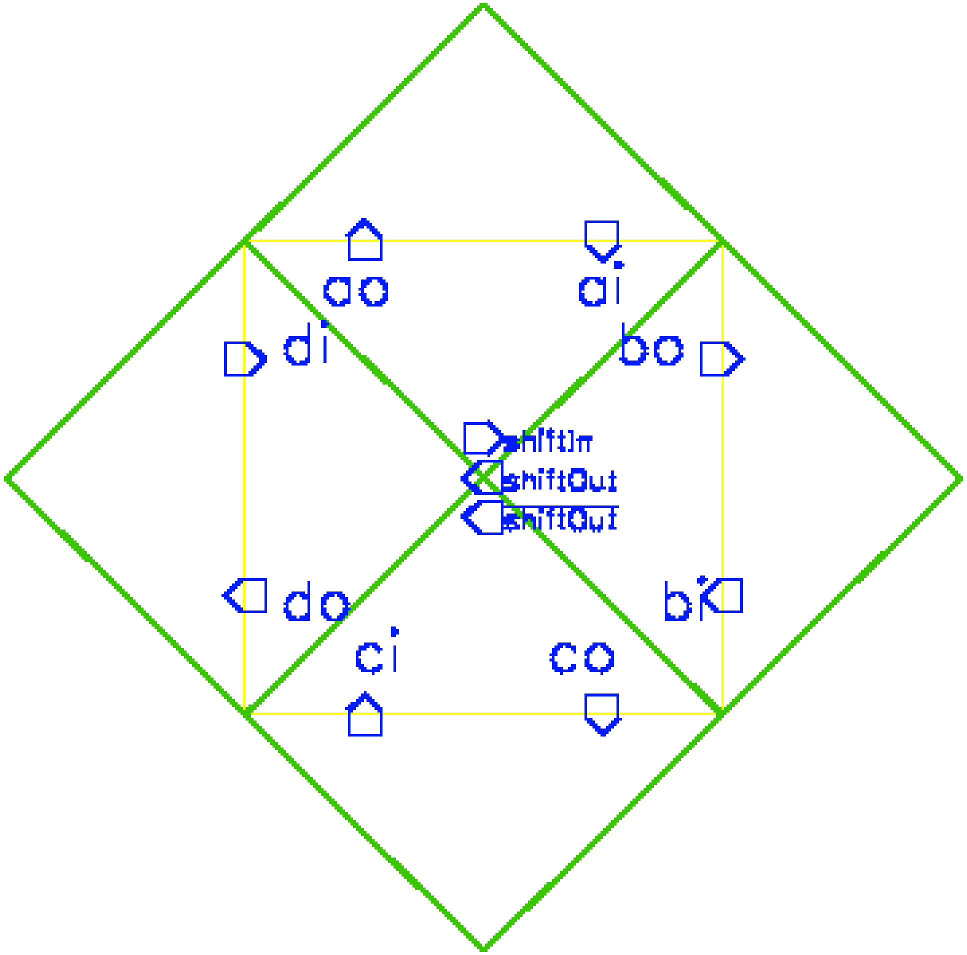
- Simplest logic found to date (in terms of # transistors).
- Computed in two logic levels: first for S, and second for A’,B’,C’,D’.

Grid of FlatTop processing elements



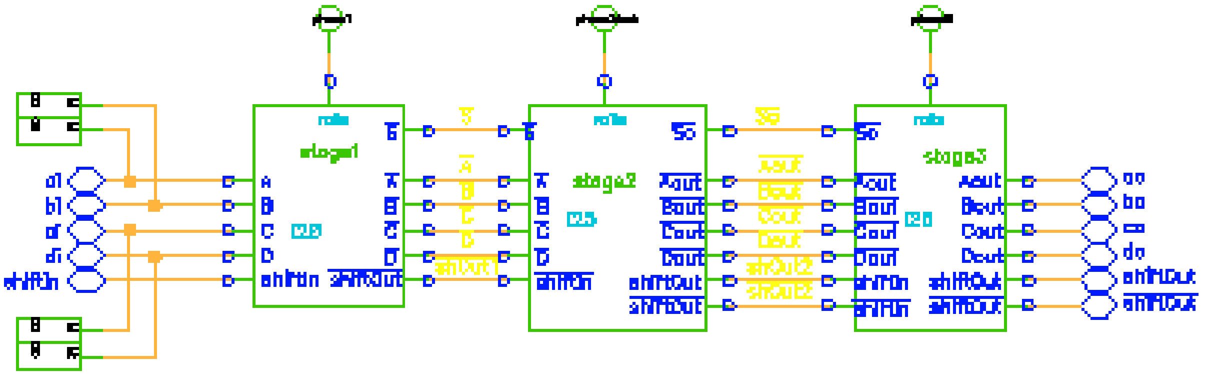
- Each 2×2 block of cells handled by one PE.
- PE visualized at center of each block.
- Cell state passed between diagonally adjacent PEs.
- PE mesh oriented 45° to cell grid.
- Chip edges chosen parallel to PE mesh.

Icon for a single FlatTop cell

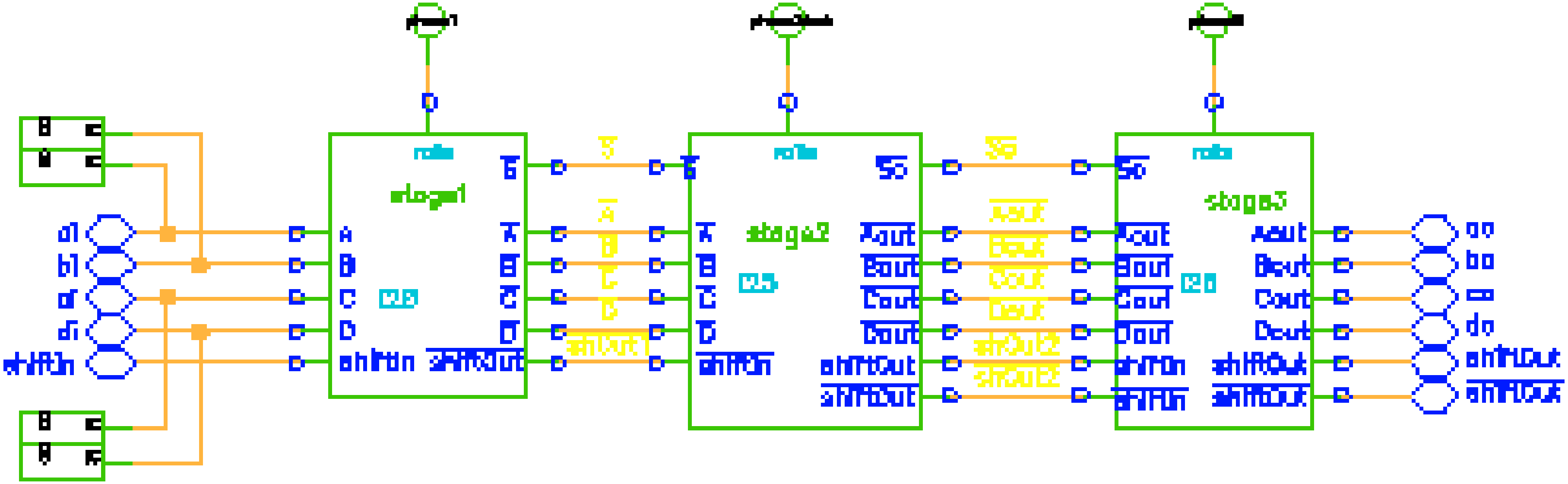


- Mnemonic graphic evoking CA functionality

Block diagram of a FlatTop processing element

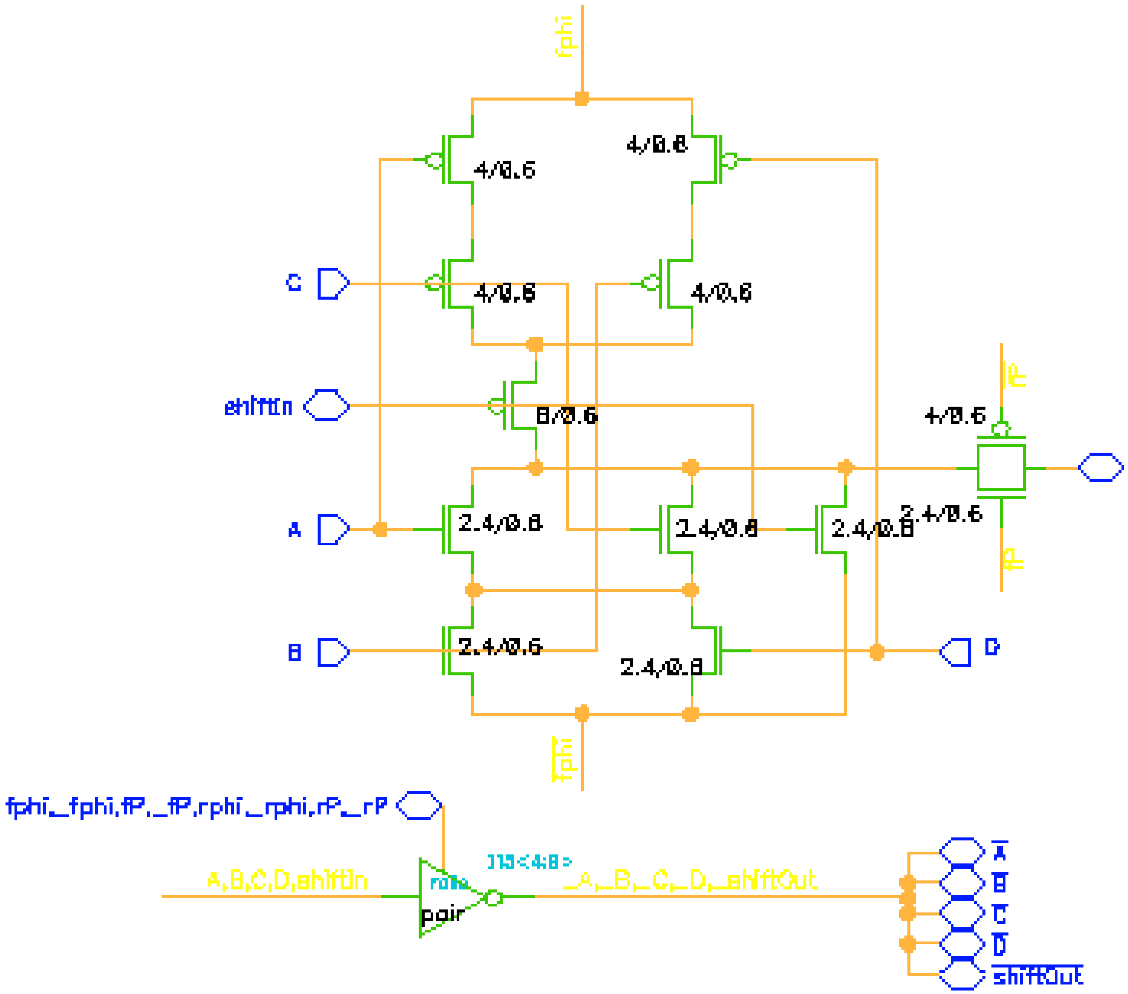


- Note 3 SCRL stages proceeding left to right.
- First stage computes S.
- 2nd stage updates the 4 cells.
- Third stage uncomputes S (and handles initialization).

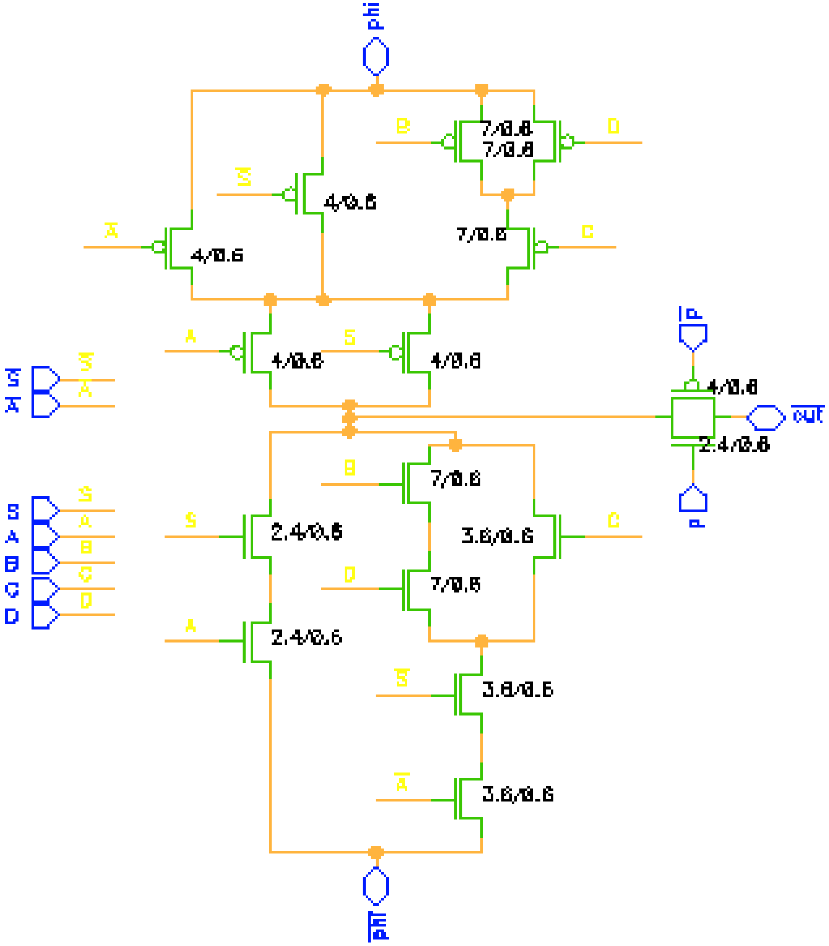


stage1
stage2
stage3
a1
b1
c1
d1
shiftIn
A
B
C
D
shiftIn
shiftOut
S
Aout
Bout
Cout
Dout
shiftOut2
So
ao
bo
co
do
shiftOut

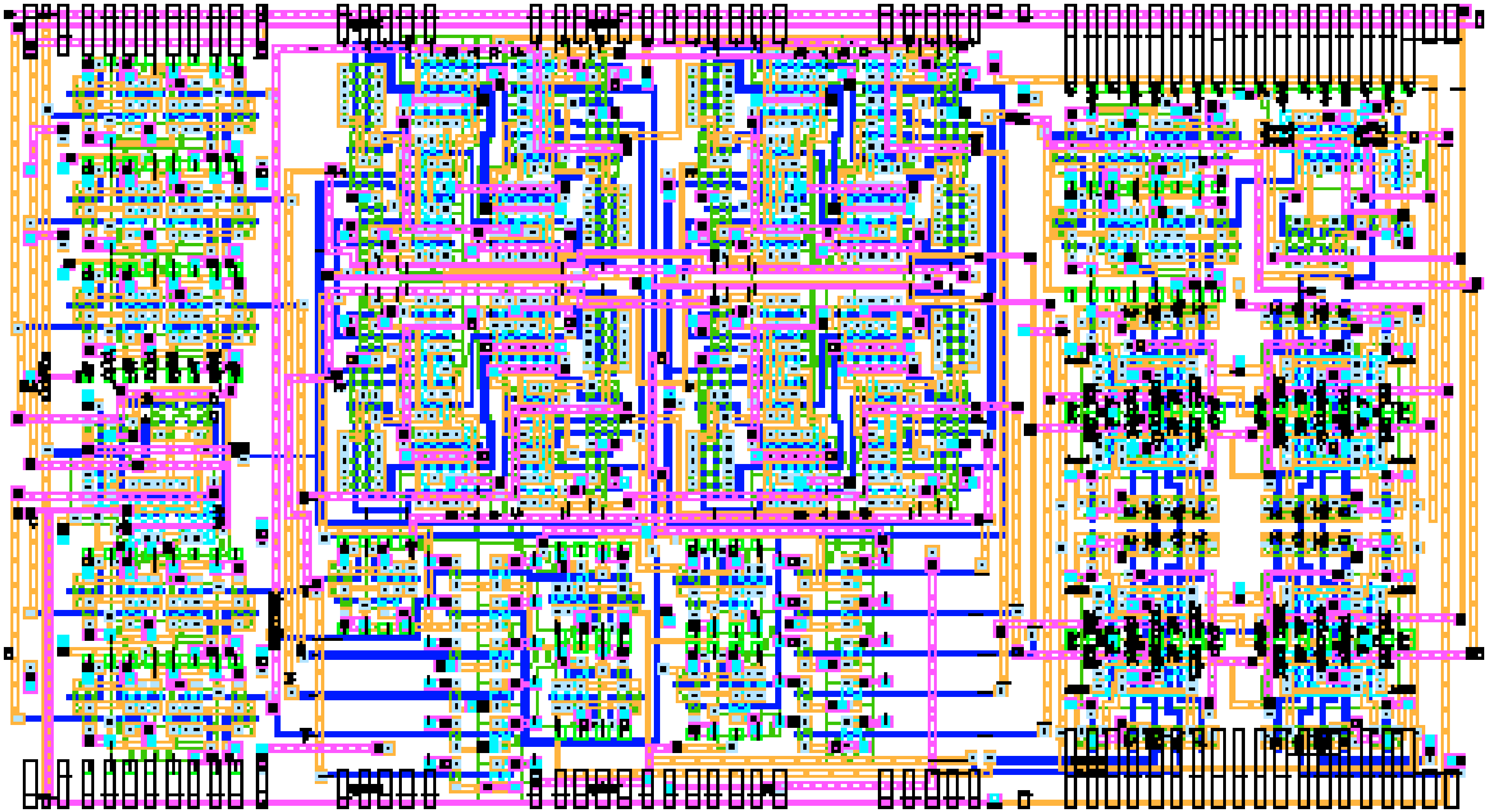
FlatTop stage 1 logic gate



FlatTop stage 2 logic

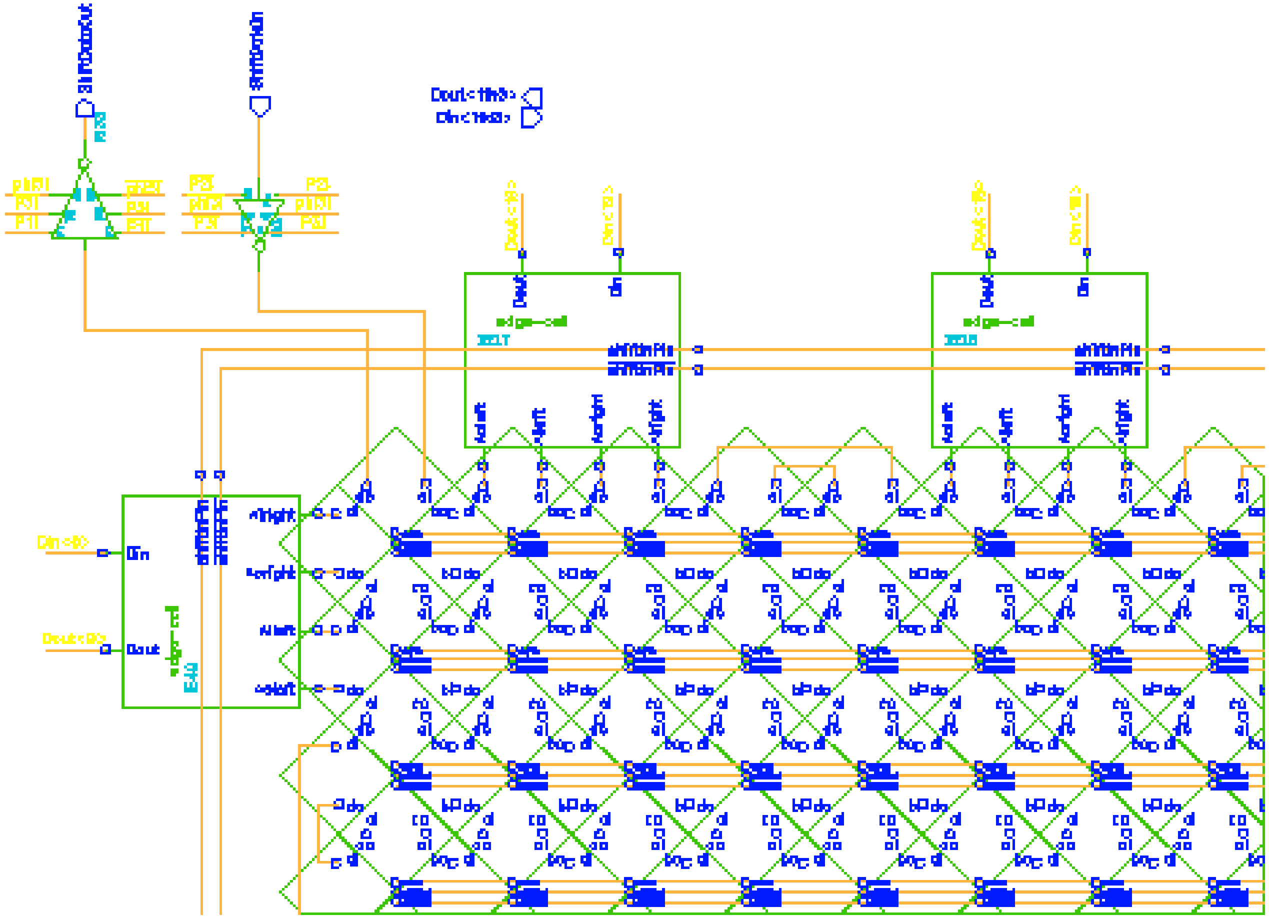


Layout of a FlatTop PE

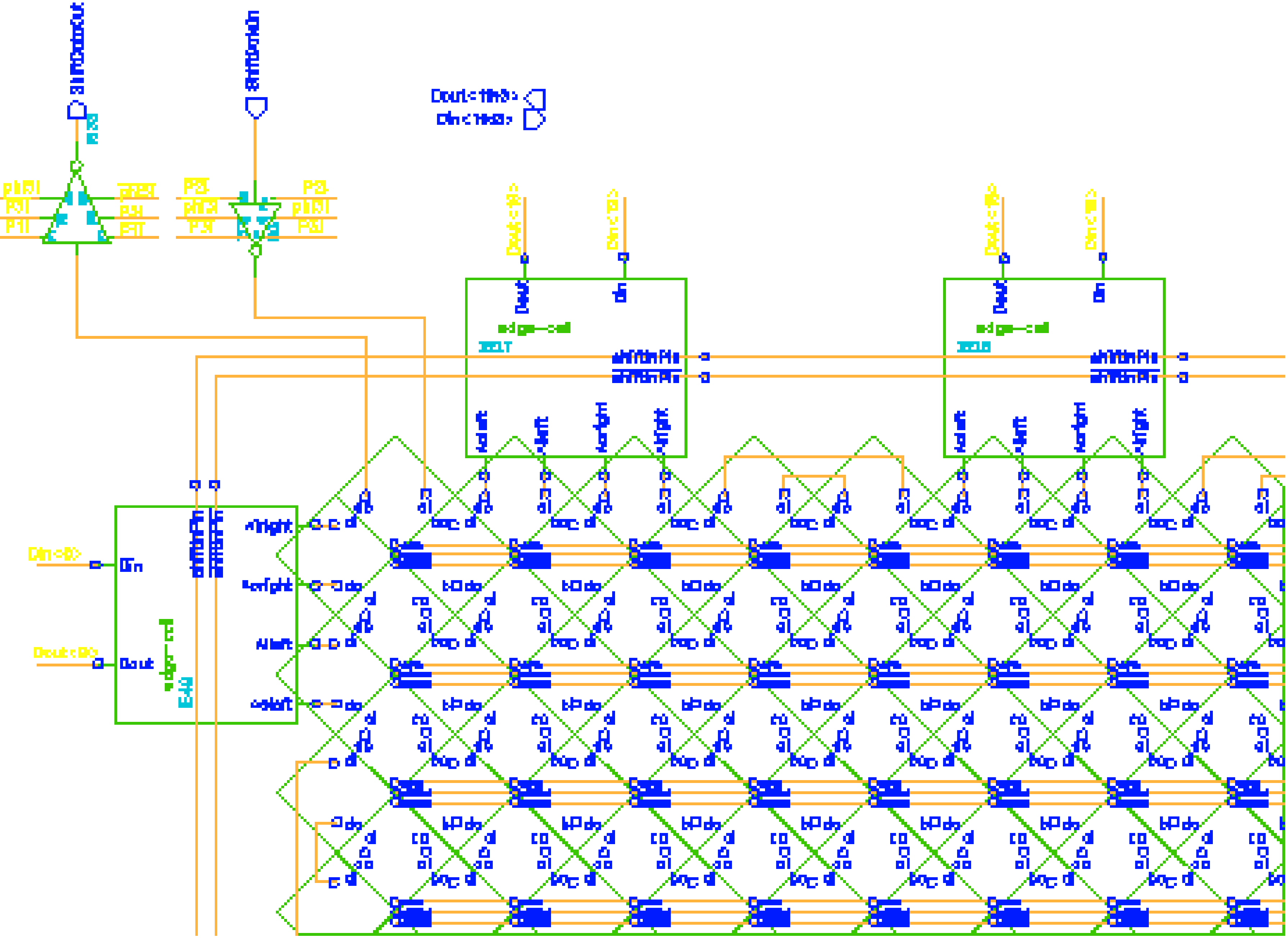


- 0.5 μm process (HP14)
- $\sim$ 300 transistors
- $\sim$ 170 μm × 90 μm

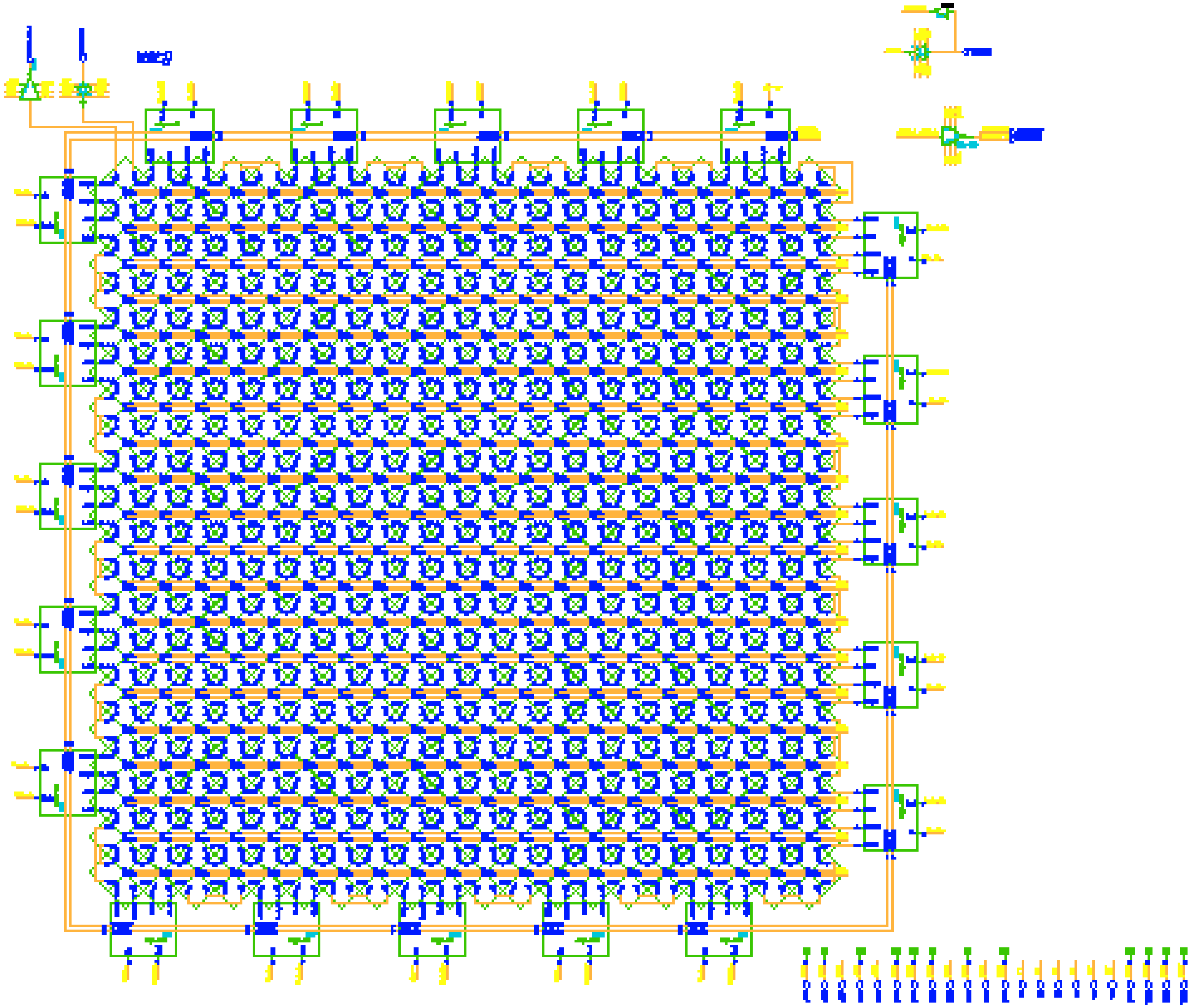
One corner of an array of FlatTop PEs



- PE icons overlap each other at corners
- Note some edge cells go to pads, others wrap around
- Initialization/readout path at upper left.


shiftDataOut
shiftDataIn
Dout<1:16>
Din<1:16>
phi1
phi2
Dout
Din
edge_cell
Din<0>
Dout<0>
Din
Dout
edge_cell
shiftIn
bbc
bb

The full 20×20 array of PEs



Flattop analysis

- Estimated room-T minimum energy/op.
- 10 fJ/cycle/cell.
- iCMOS estimated at $20pJ$.
- Factor of 2000 energy savings!
- Optimal cycle time: 12μs (83 kHz)