Embedded DRAM for a Reconfigurable Array

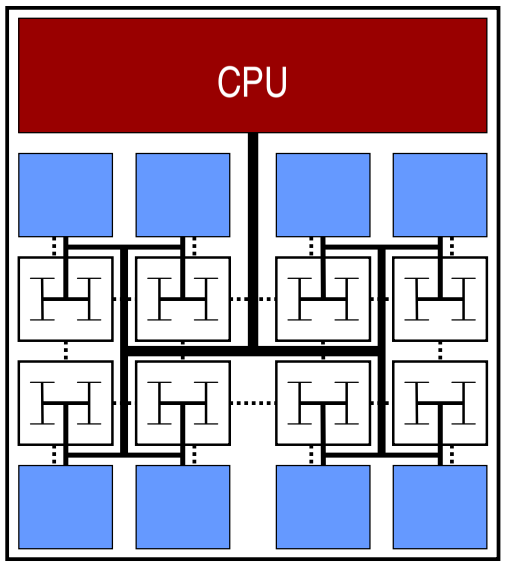
S.Perissakis, Y.Joo[1], J.Ahn[1],
A.DeHon, J.Wawrzynek

University of California, Berkeley
[1]LG Semicon Co., Ltd

Outline

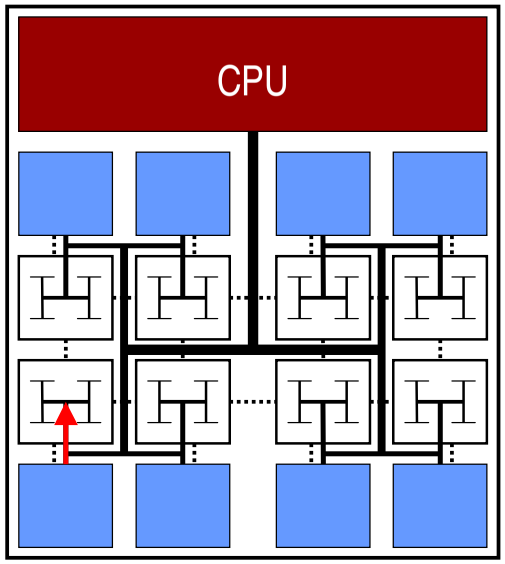
- Reconfigurable architecture overview
- Motivation for on-chip DRAM
- Configurable Memory Block (CMB)
- Evaluation
- Conclusion

Long Term Architecture Goal



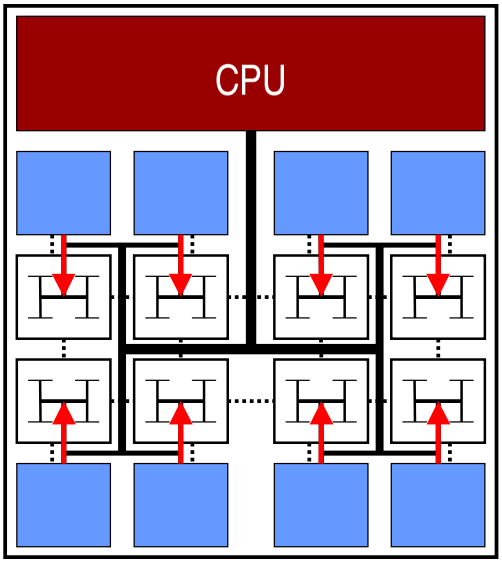
- On-chip CPU
- LUT-based compute pages
- DRAM memory pages
- Fat pyramid network
  - fat tree + shortcuts

Long Term Architecture Goal



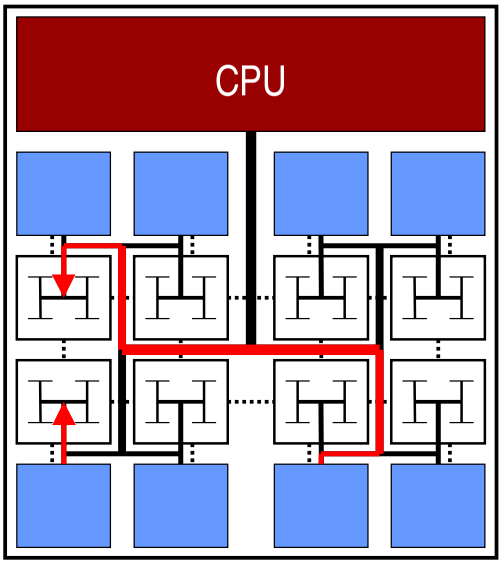
- On-chip CPU
- LUT-based compute pages
- DRAM memory pages
- Fat pyramid network
  - fat tree + shortcuts

Long Term Architecture Goal



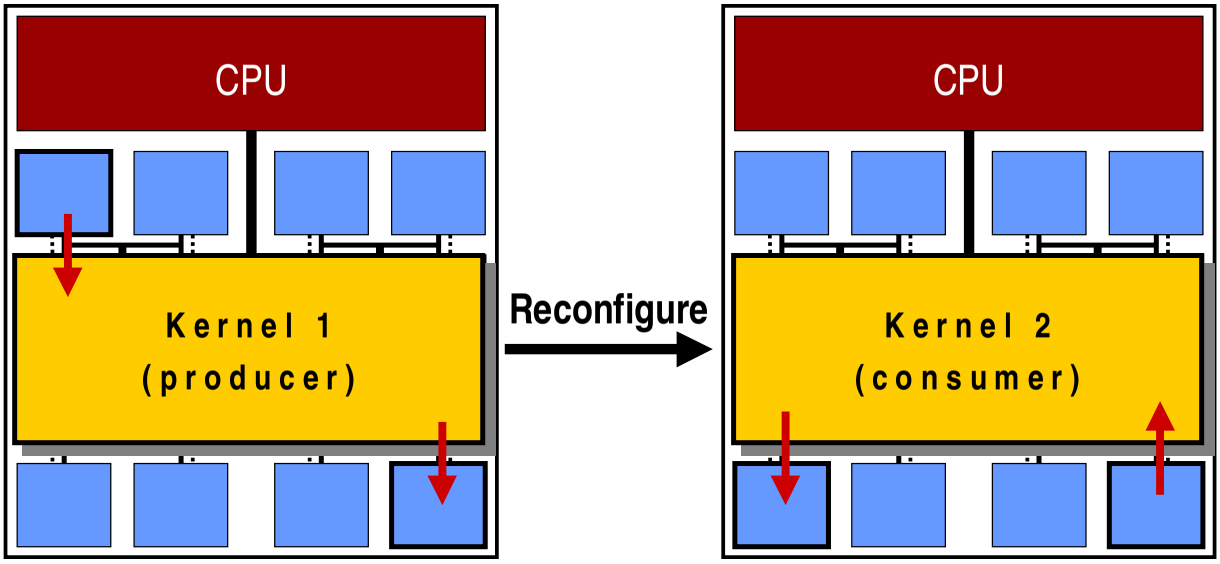
- On-chip CPU
- LUT-based compute pages
- DRAM memory pages
- Fat pyramid network
  - fat tree + shortcuts

Long Term Architecture Goal



- On-chip CPU
- LUT-based compute pages
- DRAM memory pages
- Fat pyramid network
  - fat tree + shortcuts

Long Term Architecture Goal



Motivation

Need large on-chip memory for:

- Stream buffers
  - Reduce reconfiguration frequency
- Configuration memory
  - Speed up reconfiguration
- Application memory
  - Speed up individual kernels

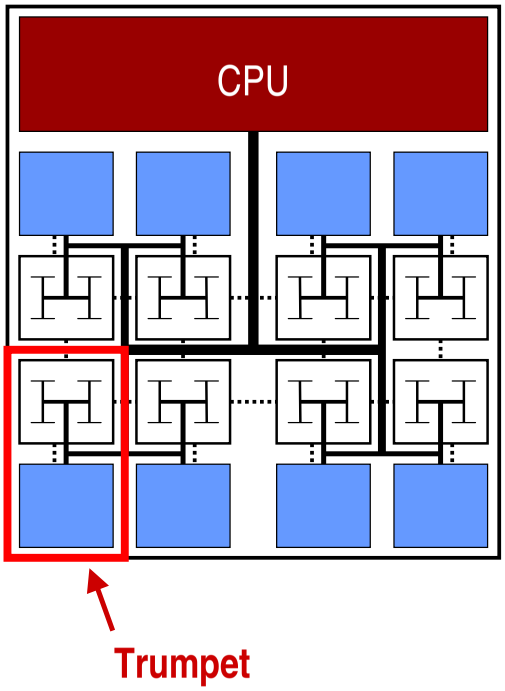
Challenges

DRAM offers increased density (10X to 20X that of SRAM), but:

- Harder to use
  - Row/Col accesses & variable latency
  - Refresh
- Lower performance
  - Increased access latency

Q: Is it worth the trouble ?

Trumpet test chip

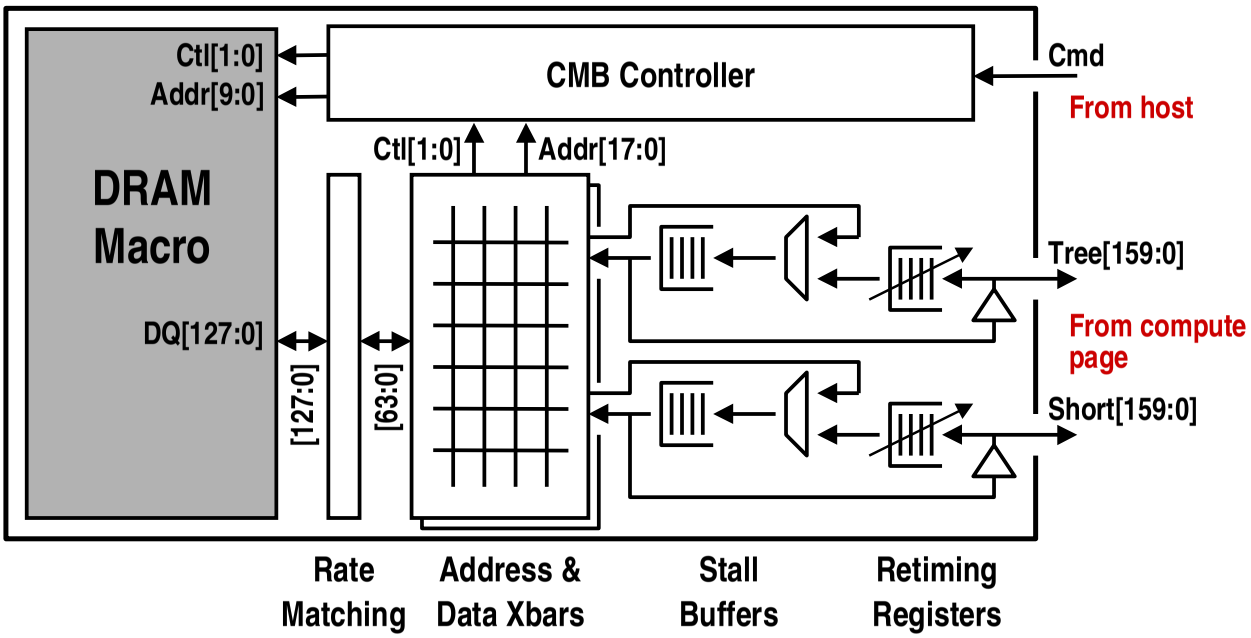


- One compute page
- One memory page
- Corresponding fraction of network

CMB Functions

- Configuration source
- State source/sink
- Data store
- Input/output

CMB Overview



DRAM Macro

- 0.25µm, 4 metal eDRAM process
- 1 to 8 Mbits (2 Mbits in test chip)
- 128-bit wide SDRAM interface
- Up to 125 MHz clock → 2 GB/s peak B/W
- 36ns/12ns row/col latencies
- Row buffers to hide precharge & refresh

Designed by LG Semicon

SRAM Abstraction

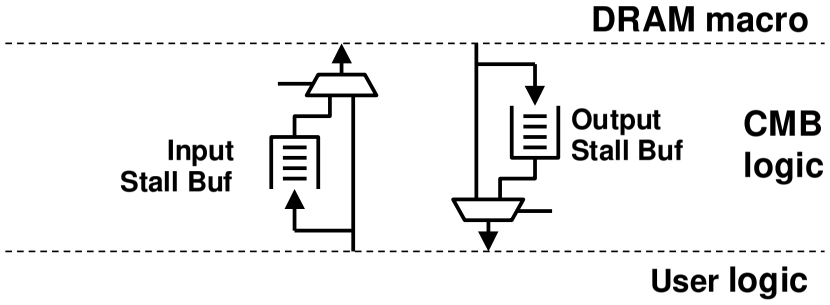
- SRAM-like interface
  Req, R/W, Address, Data
- Row buffers → simple direct-mapped cache
- 6-cycle minimum latency, pipelined
- Misses handled by logic stalls
- 10-cycle miss latency "hidden" from logic

Stalls

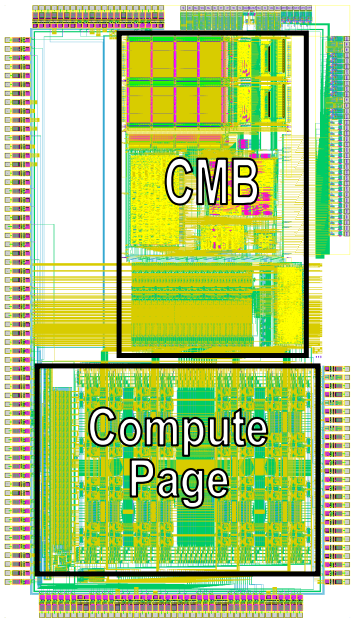
- Stall sources:
  - Row buffer miss (10 cycles)
  - Write after read (4 cycles)
  - DRAM/logic clock alignment (1 cycle)
  - Refresh (*Halt* from host)
- Multicycle stall distribution

Stall Buffers

- Memory page is never stalled
  - Must buffer read data during stall
  - Must buffer requests during stall distribution

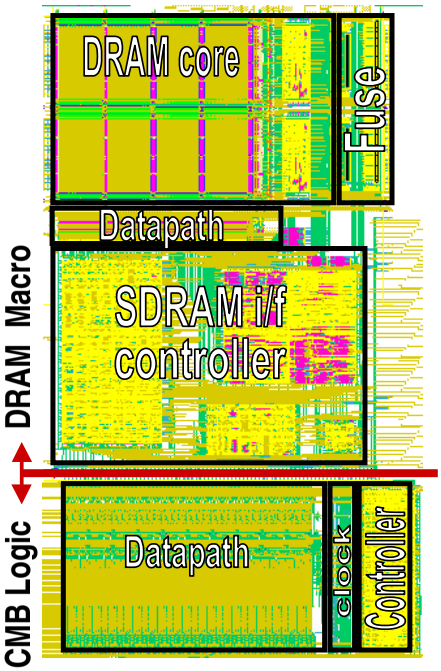


Trumpet Test Chip



- 0.25µ DRAM, 0.4µ logic
- 2 Mbits + 64 LUTs
- 125 MHz operation
- 1 GB/sec peak bandwidth
- 10 µsec reconfiguration
- 10 x 5 mm^2 die
- 1 W @ 125 MHz

CMB Area Breakdown

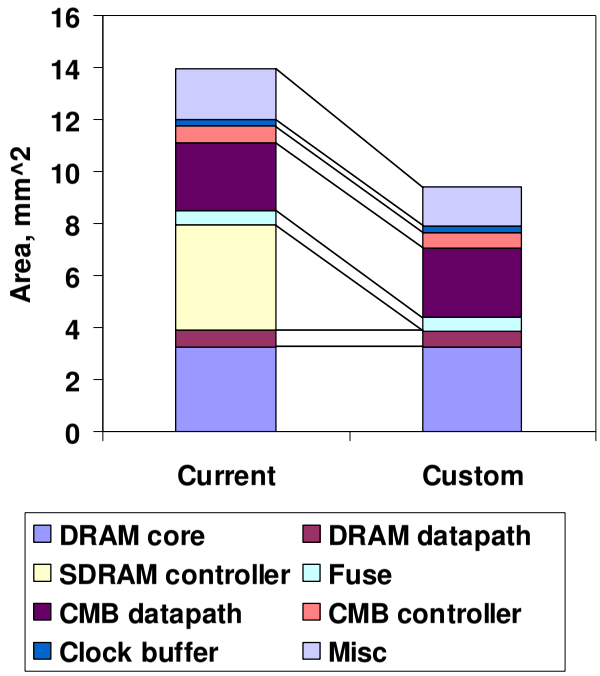


- 13.95 mm^2 total
- 2 Mbits capacity

→ 147 Kbits/mm^2
average density

Compare to 700-900
Kbits/mm^2 commodity DRAM

Using a Custom Macro



- Existing:
  - 13.95 mm^2
  - 147 Kbits/mm^2
- Custom:
  - 9.4 mm^2
  - 218 Kbits/mm^2

Comparison to SRAM CMB

With typical SRAM core densities and:

→ No stall buffers

→ Simplified controller

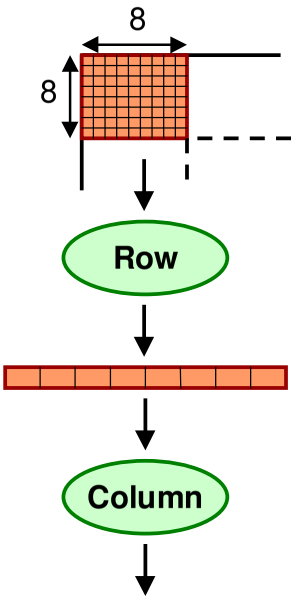
- DRAM (custom macro) → 218 Kb/mm^2
- SRAM (equal area) → 25 Kb/mm^2

∴ Close to 1 order of magnitude density advantage for DRAM

Performance

- Configuration / state swap: peak 1 GB/s
- User accesses: dependent on access patterns
  - Peak if high locality
  - Near peak for sequential patterns (62-93%)
  - Column latency exposed when dependencies exist, or on mixed R/W
  - Row latency exposed on random accesses

Performance (example)



Input image
Scanline order

Row: ~ 4 misses / DCT block

8x8 DCT block
1 Kbit = 1 DRAM row

Col: 2 misses / DCT block

→ 73% efficiency

Refresh Overhead

- 8 to 16 ms retention time expected
- 2.5% to 5.0% bandwidth loss
- Can reduce by refreshing only active part of memory
- May skip refresh for short-lived data

Conclusion

- Q: Is on-chip DRAM advantageous to SRAM ?
- Our experience so far:
  - User-friendly abstraction possible
  - Can maintain density advantage
  - Effect on application performance:
    - » Large buffer space → less frequent reconfiguration
    - » High bandwidth → faster reconfiguration
    - » Effect on individual kernels often limited by DRAM core latency