Interconnection Issues between Memory and Logic in IRAM Systems

J. Golbus, B. Gribstad, C. Kozyrakis and K. Wang

Spring 1997

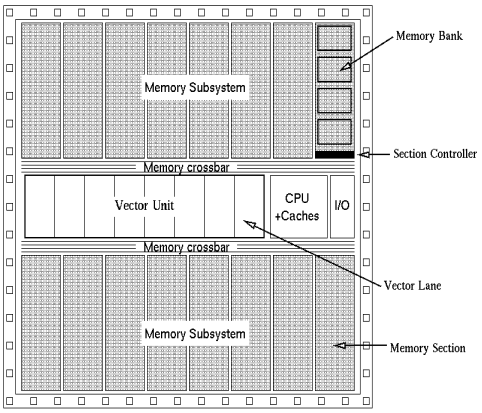
http://www.cs.berkeley.edu/~kozyraki/project/ee241

Introduction

- IRAM: integration of general purpose processor and DRAM memory on same chip
- This work: address circuit and architecture issues on the processor-DRAM interconnect
- Desired properties of on-chip interconnect:
  - high bandwidth
  - low latency
  - multiple independent address/data streams
  - low energy
  - area efficiency

Potential IRAM System Floorplan

- Processor: multiple *load-store units*
- Memory divided into *sections*: minimum memory partition addressed independently
- Section: memory banks and section controller
- Load-store units and sections connected through *memory crossbar*

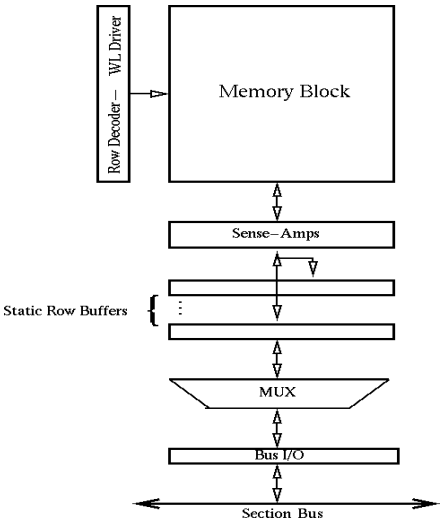


Outline

- Memory System Architecture
- Memory System Interface
- Evaluation of Bus Structures in a DRAM Process
- Design Implementations of the Memory Crossbar
- Summary

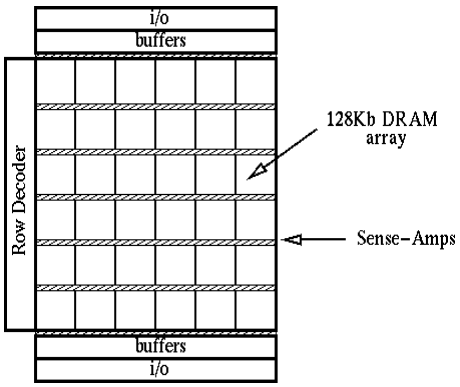
Memory Bank Organization

- Multiple data words per memory row
- One or more row buffers for:
  - decoupling of consecutive accesses
  - caching and prefetching (reduced number of RAS accesses)
- Single word interface to section controller
- No need for large column decoder



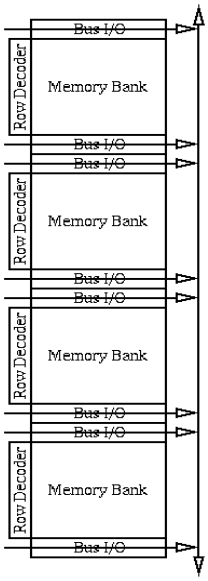
Memory Bank Implementation

- Basic DRAM array: 128Kb - 256Kb
- Rectangular scheme to amortize area of periphery
- Buffers at top/bottom for higher degree of sharing (associativity)
- Transfer from SAs to buffers with segmented bit-lines or additional layer of metal
- Single memory row activated per RAS access for energy efficiency



Traditional Interconnection of Banks

- Small number of SA outputs selected with column decoder and driven to narrow I/O bus
- I/O bus runs on the side of memory banks
- Either limited bandwidth or high-energy penalty for accessing more data
- Bandwidth restricted by narrow external interface (few pads) anyway



Memory Section Organization

•Wide section bus running on top of memory banks

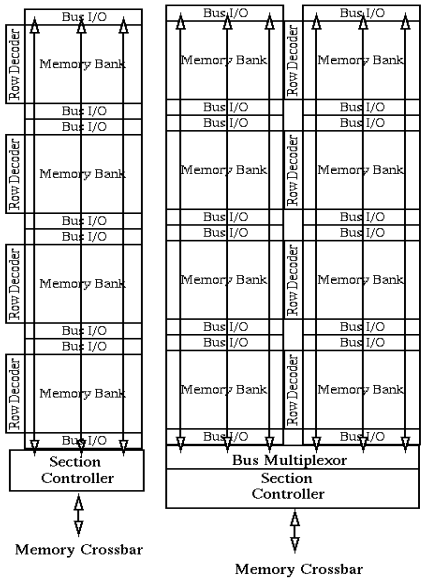
•High-bandwidth at the energy cost of a single bank access

•Requires an additional layer of metal, even two if noise to memory is high

•If (few) large sections needed, multi-column structure:

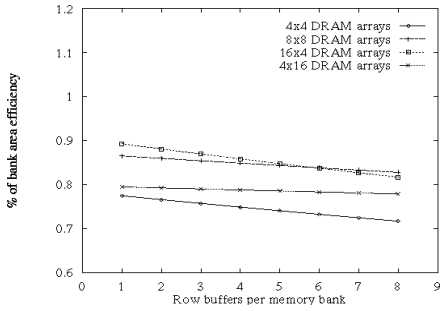
–Bus multiplexor needed

–Sharing of single decoder



Area Efficiency Considerations

- Trade-offs:
  - size of banks: higher energy efficiency Vs higher area efficiency
  - number of row buffers: cache Vs memory
  - size of sections: available parallelism Vs processor and crossbar complexity
- Used data from NEC 256Mbit DRAM (0.25um CMOS)
- Bank Area Efficiency:
  - decoder area amortized if 4 columns of DRAM arrays used
  - periphery area amortized if 8 rows of DRAM arrays used



Memory Area Considerations (Examples)

- Assuming a 20mm x 20 mm chip with 1/4 of the area for the processor
- Case 1:
  - Memory bank: 4x4 DRAM arrays, 4 row buffers
  - 192 banks on-chip (2Mbits/bank) organized as 32 sections with 6 banks per section or 16 sections with 12 banks per section etc
  - Area efficiency: 74%
- Case 2:
  - Memory bank: 8x8 DRAM arrays, 4 row buffers
  - 48 banks on-chip (8Mbit/bank) organized as 8 sections with 3 banks per section
  - Area efficiency: 75%

Memory Subsystem Interface

- Interface: commands for accessing memory banks
- Goal: small flexible interface that takes advantage of architecture both for performance and energy
- RDRAM-like interfaces:
  - Packet protocol adds delay and circuit overhead
  - optimized for off-chip narrow interconnects
  - On-chip requirements and limitations different
- SDRAM-like interfaces:
  - easy to pipeline
  - easy to adapt to on-chip requirements
  - Prefetched interfaces are energy inefficient. Explicit prefetching preferred. Within a single bank, prefetching done due to wide rows

Memory Interface Commands

Command	Arguments	Operation
Precharge	#bank	Precharge bit-lines
Read2B	#bank, #row, #buffer	Read memory row to buffer – data restoring optional
Write2B	#bank, #row, #buffer	Write-back buffer to memory row
Read2S	#bank, #buffer, #word	Read word from buffer to section bus
Write2S	#bank, #buffer, #word	Write word from section bus to buffer
Refresh	#bank, #row	Refresh memory row

- Restoring data that may soon be updated is energy inefficient
- Refresh does not go through buffers to avoid thrashing cache; this is why SAs not used as buffers
- Interface not dependent on processor architecture; commands can be generated at processor or section controller

Pipeline Stages of Interface Commands

Command	Stages
Precharge	Precharge bit-lines
Read2B	Row decoder RAS access and amplification Row buffer latch – (*) row restoration
Write2B	Row decoder – Load SAs from buffer RAS access
Read2S	Word selection/multiplexing Drive section bus
Write2S	Drive I/O bus Latch word to proper buffer
Refresh	Row decoder RAS access and amplification Row restoration

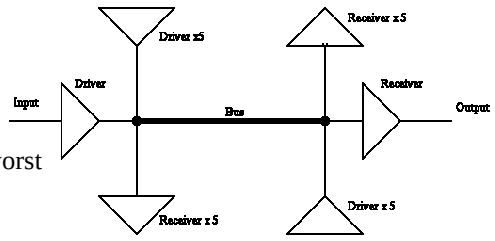
•Longest stage determines clock cycle of synchronous interface

•In most SDRAM implementations, transactions over the I/O bus are the slowest steps of the interface

Memory Section Bus

- Motivation: section bus expected to be the slowest stage in the synchronous interface
- Requirements for section bus
  - high speed
  - low energy
  - low noise levels
- For a 20mm x 20mm IRAM chip, worst case length 8mm
- Transistor models and parasitics (resistance, capacitance, coupling) for bus simulations from the TI 16Mbit DRAM (0.5um CMOS) process
- Metal 2 wires used in all cases, since metal 1 used in DRAM arrays

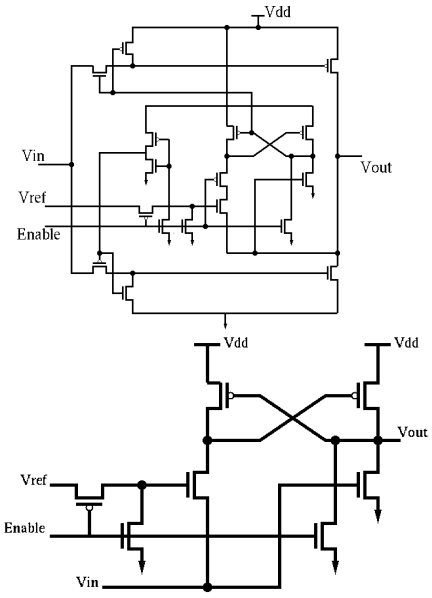
Simulated Circuit



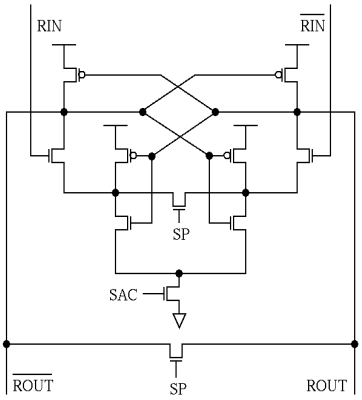
Bus Schemes

- Conventional case:
  - driver: single clocked inverter
  - receiver: single clocked inverter
  - full-swing
- Precharged Bus
  - driver: clocked pull-down network
  - receiver: clocked inverter
  - wires precharged to Vdd/2
  - low-swing
- Single-ended Low-swing Bus:
  - driver: inverter controlled by a Voltage Sense Translator to adjust output voltage
  - receiver: 3-state Voltage Sense Translator
  - reference voltage is Vdd/2
- Differential Low-swing Bus:
  - driver: minimum sized clocked inverters
  - receiver: source controlled sense-amplifier
  - wires precharged to Vdd/2
  - clock generation circuits required both at driver and receiver for lower (1 per bus)

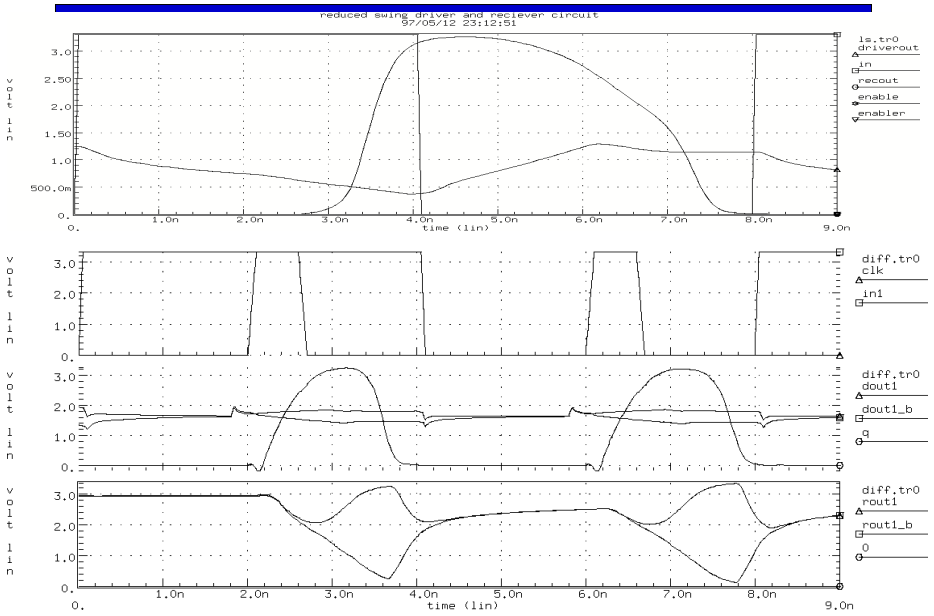
Single-ended Bus



Differential Bus



Low-swing Busses: Waveforms



Section Bus Simulation Results

	Conventional	Precharged	Low Swing Single Ended	Low Swing Differential
Swing (V)	3.3	0.5	0.76	0.4
Average Energy Per Transition (pJ)	13.1	3.93	7.44	2.06
Static Power (nW)	0.231	4.95	1.5	151
Input Capacitance (fF)	46	1.76	0.46 /9	23.8
Area (um^2)	377 (D) 213 (R)	40 (D) 88 (R)	1062 (D) 330 (R)	342 (D) 673 (R)
Delay (ns)	3.79	1.9	3.91	0.87

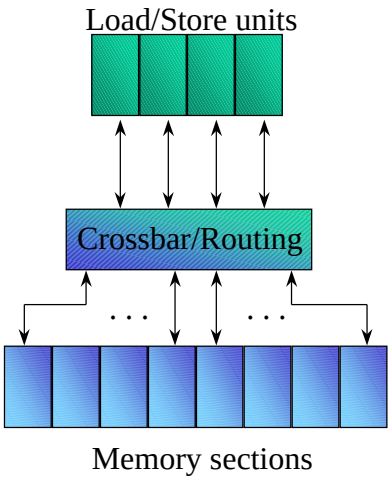
- Goal: achieve 4ns (250MHz) transfer delay for all cases
- Precharged bus is extremely sensitive to noise and device variation
- Differential bus requires twice the number of wires

Memory Crossbar

Serves as the interface between the processor and the memory sections

Requirements include:

- Arbitrary connections
- Simultaneous loads and stores
- Sufficient bandwidth



Crossbar Design Challenges

The requirements of the processor/memory bus present special challenges

- Delay

  - Typical wires are ~20mm long

- Area

  - Area α Number of wires (768 wires)

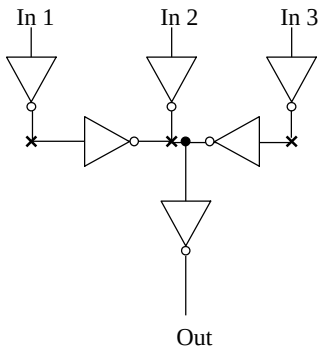
- Control

  - Control signals may need to be distributed throughout the crossbar

What about Buffers?

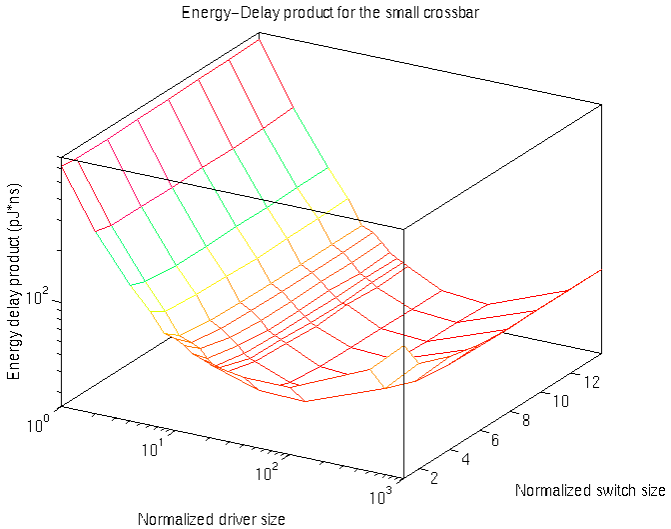
Buffers are often used to effectively drive long wires, *however:*

- Some wires are bi-directional
- Due to switching, some buffers may be left floating
- Even/Odd number of inversions

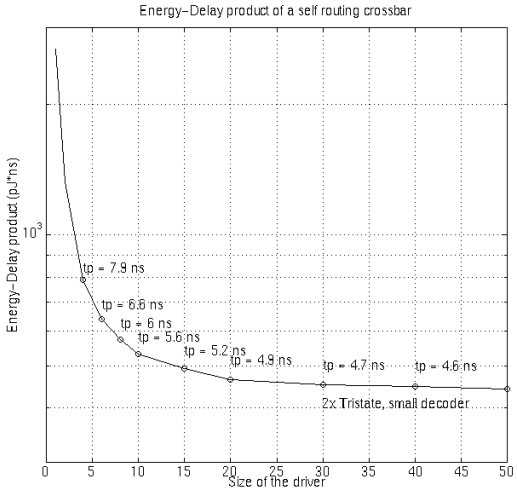


⇒ Buffers are difficult to place

Energy Delay Product Vs Device Size

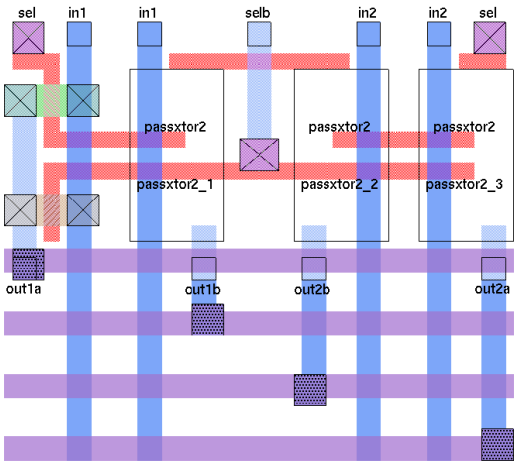


Energy Delay Product Vs Driver Strength

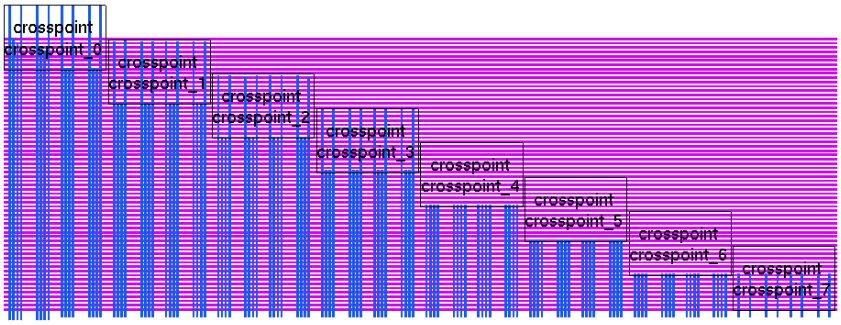


Crossbar Switch Layout

- Metal2 can be routed over switches
- Area determined only by the number of horizontal metal2 lines, and the spacing between them
- If switch height is equal to height of *n* horizontal wires, put *n* switches in a row



Section of Crossbar

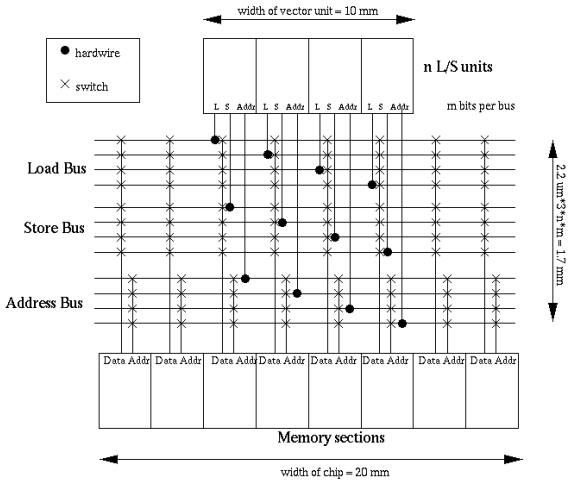


Crossbar Alternatives

Several alternatives were also considered:

- Differential bus
  - Number of wires double
- Butterfly network
  - Fewer switches
  - Less connectivity
- Serial bus
  - Significantly fewer wires
  - Loss of bandwidth unless overclocked

Large Fully-Connected Crossbar



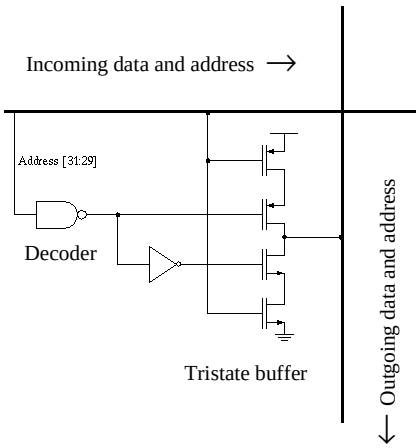
Self Routing Crossbar

Advantages:

- Control signals are generated locally
- Tristate buffer breaks the line
- No pass transistors
- Reduces critical path

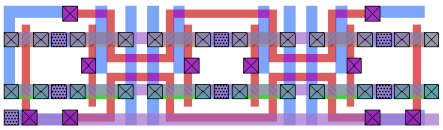
Disadvantages:

- 13% larger than a pass-transistor crossbar
- Added complexity

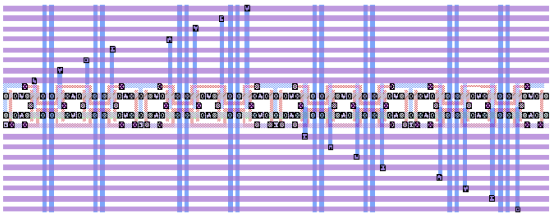


Self Routing Crossbar

The self routing crossbar consists of a core of buffers:



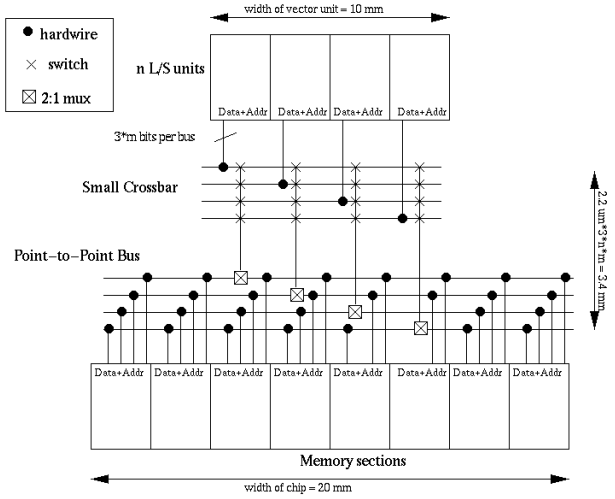
Two inverters and four tristate buffers



16x16 implementation (450λx135λ)

which are then tiled to create larger structures.

Small Crossbar and Bus



Crossbar Area and Speed Results

	Small crossbar and bus		Large fully- connected crossbar		Self-Routing Large Crossbar
	full swing bus	low swing bus	full swing	low swing	Full swing
Delay (ns)	7.7	10.0	7.9	9.5	6.0
Energy/ transition (pJ)	84.5	66.5	91	53.7	95
Static Power (nW)	2.5	17.7	1.3	65.2	5
Energy-Delay Product (10^{-21} J·s)	651	665	719	510	570
Area per bus (mm^2)	4.22		2.82		3.16

Summary

- Proposed an architecture and interface for high-bandwidth, low-energy memory system on IRAM
- Evaluated the speed, energy and area of several different bus schemes in a DRAM process
- Studied design options for memory crossbar
- Quantified the effect of design decisions and trade-offs