

Embedded Memories

CSE 291E / EE260C

Spring 2002

University of California

San Diego

The logo for the University of California San Diego, featuring a stylized graphic of horizontal bars and a swoosh behind the text.

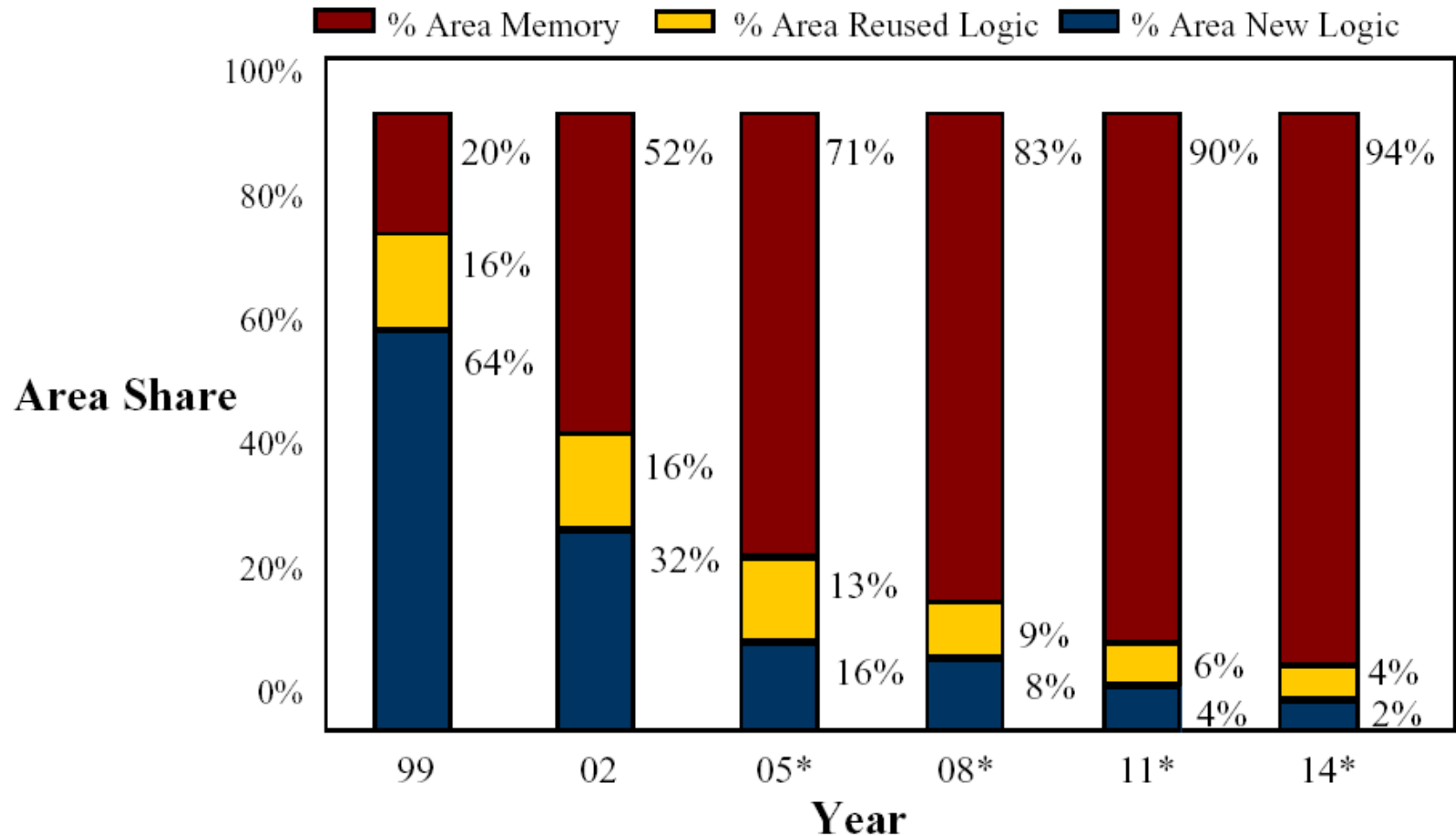
Outline

- Embedded Memories
 - How memory fits into the roadmap
- Memory Technology
 - SRAM, DRAM, ROM, Flash
- DRAM Memory Architectures
 - Page Mode, SDRAM, Rambus

Introduction

- Every embedded system needs memory
 - Buffering, Work Space, Instruction and Data Store
- Fast growing, increasingly integrated
- What are the needs of system?
 - High Bandwidth
 - Low Latency
 - Low Area/Cost
 - Non-volatility (in some cases)
 - Writable (in most cases)

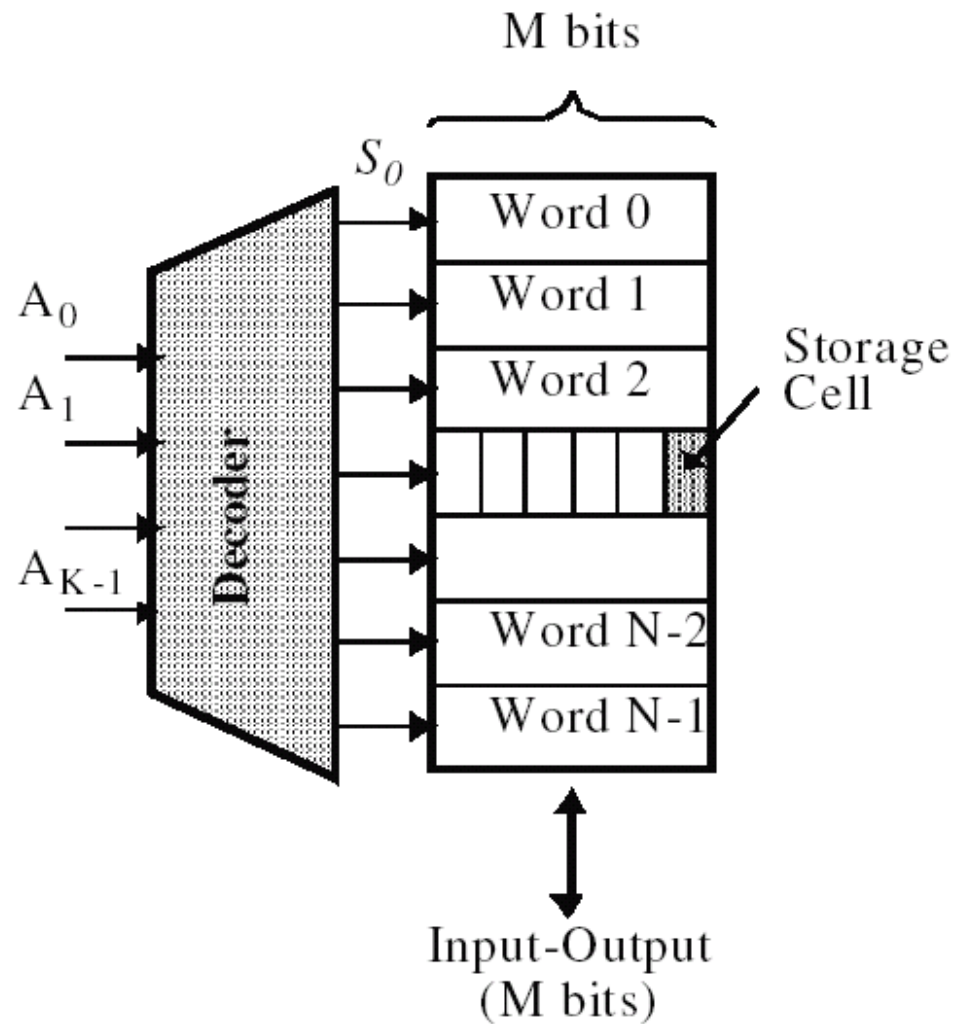
Embedded Memory Importance



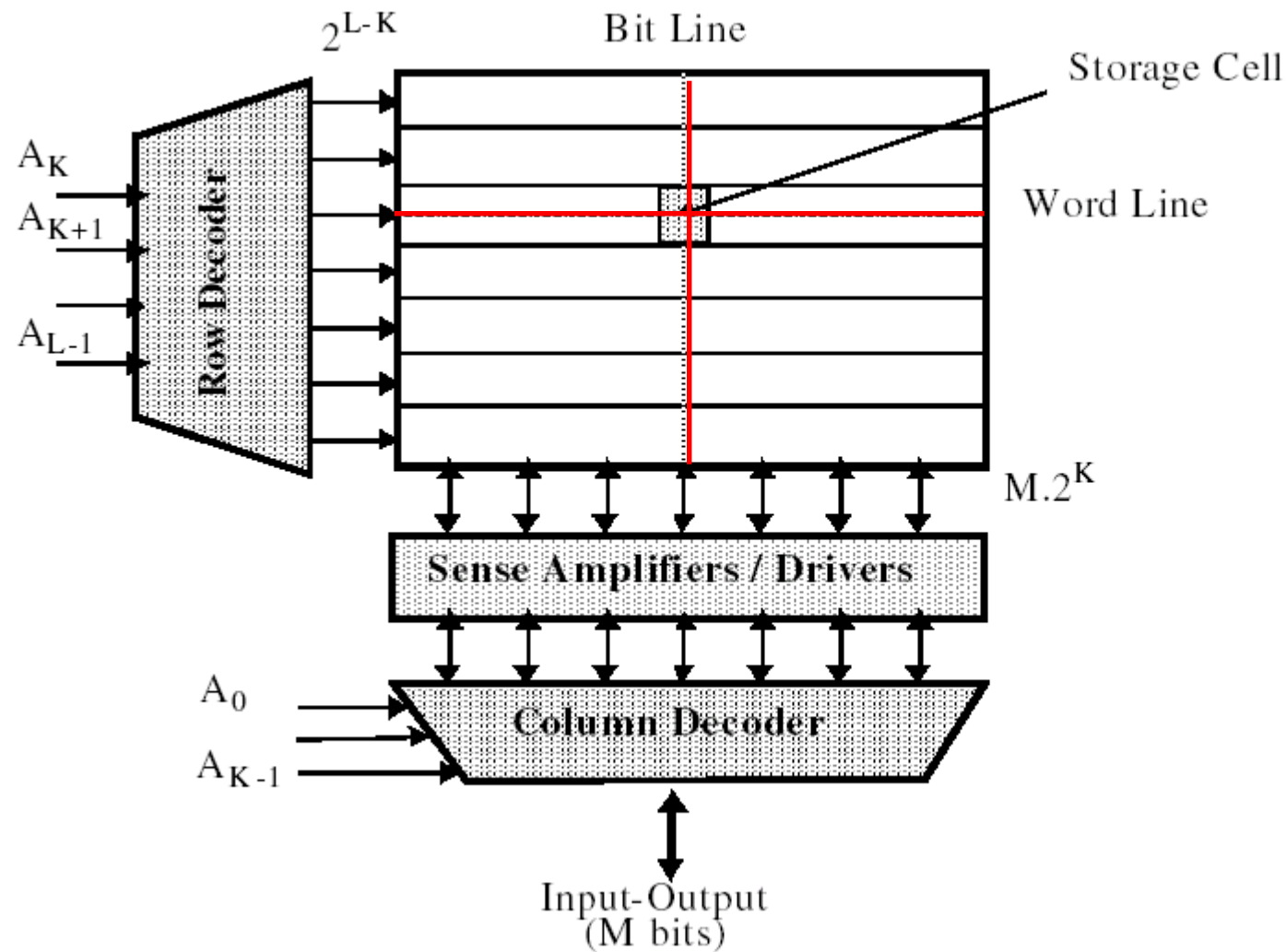
Source: SIA, ITRS 2000

* Forecast

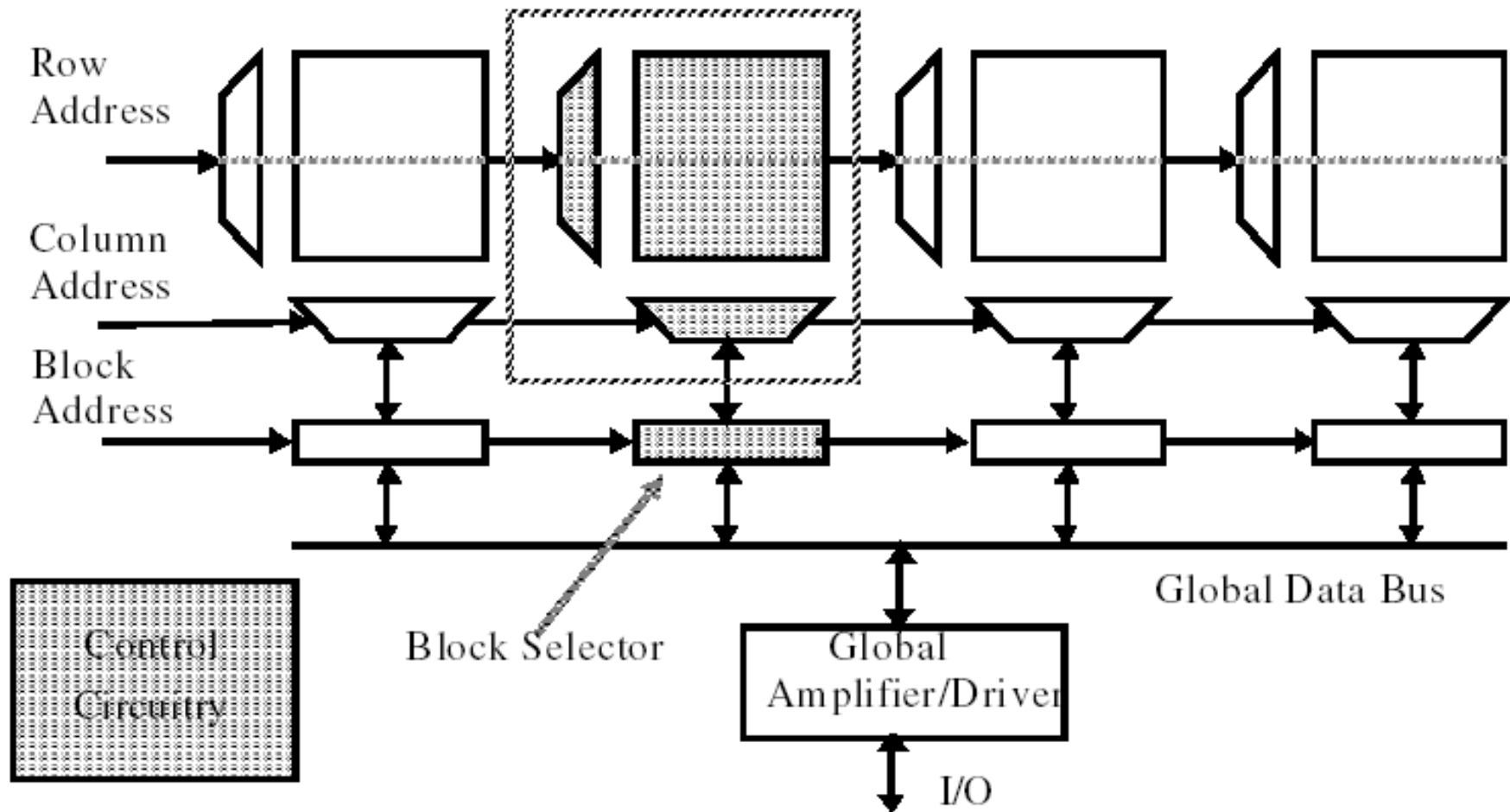
Memory



Memory with col



Memory with Col and Bank

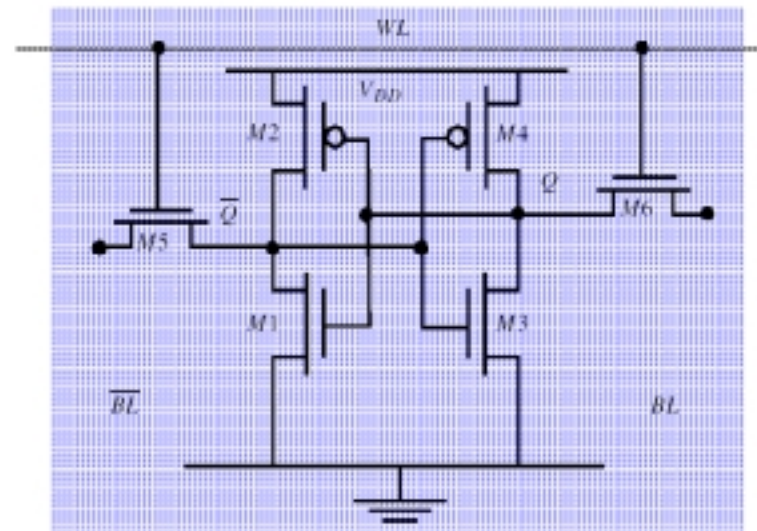
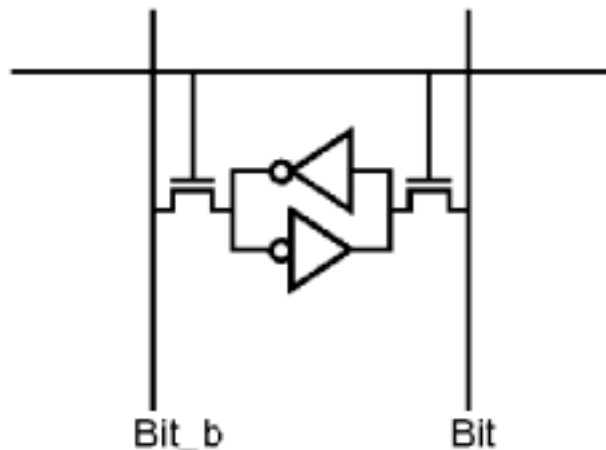


Options

- SRAM
 - Pros: Fast, available
 - Cons: Volatile, Large Area
- ROM
 - Pros: Fast, Non-volatile, Extremely dense
 - Cons: Can't write, Requires data at fabrication
- DRAM
 - Pros: Very Dense, Writable
 - Cons: Slow, Volatile, Requires refresh, Special process
- Flash
 - Pros: Dense, Writable and Non-Volatile
 - Cons: Special Process, Slow

SRAM Cell

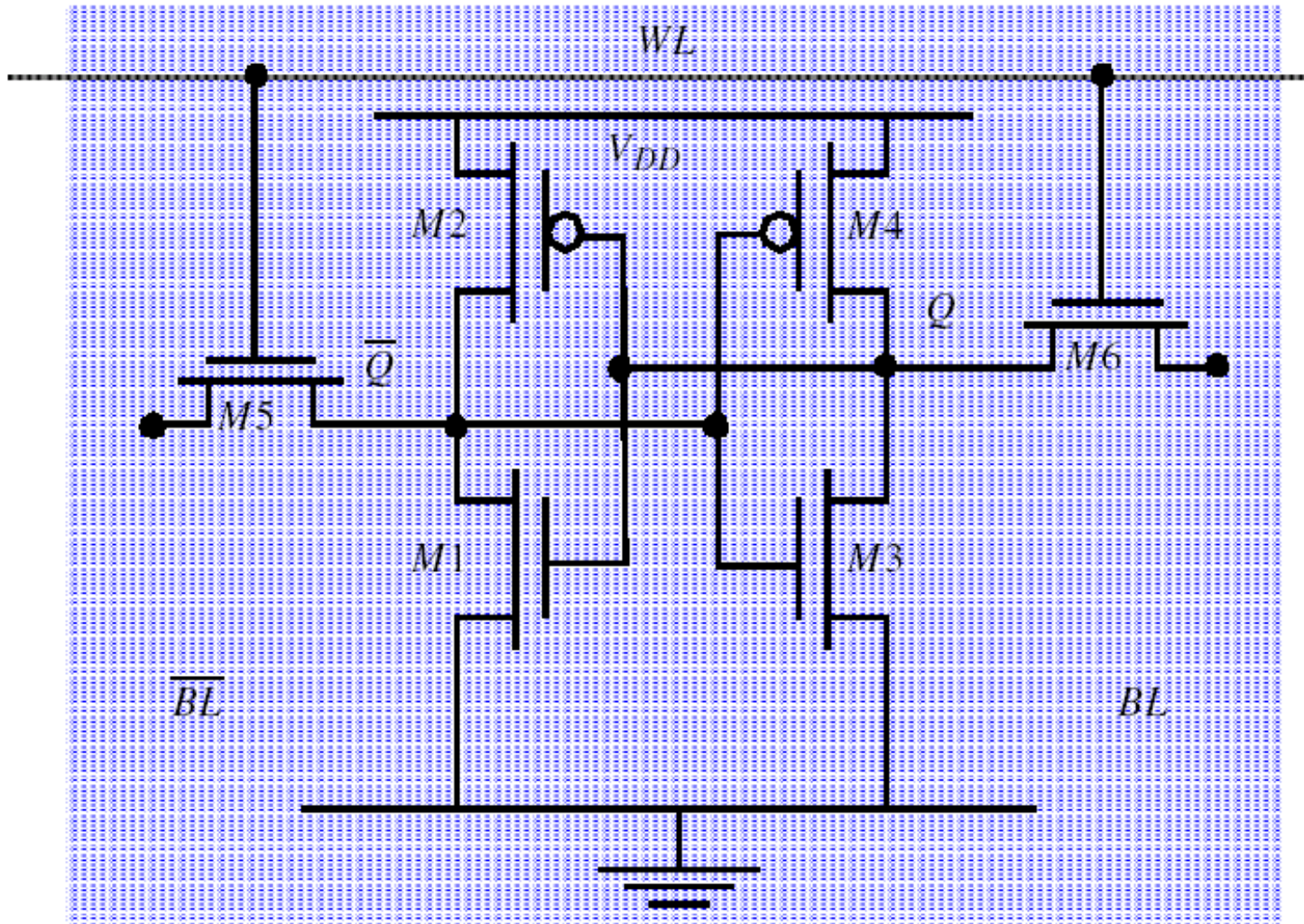
- Basic 6-T SRAM cell
- Stores the value in a pair of inverters
- Both the value and it's complement are read out
- Gates connected to the word line control access



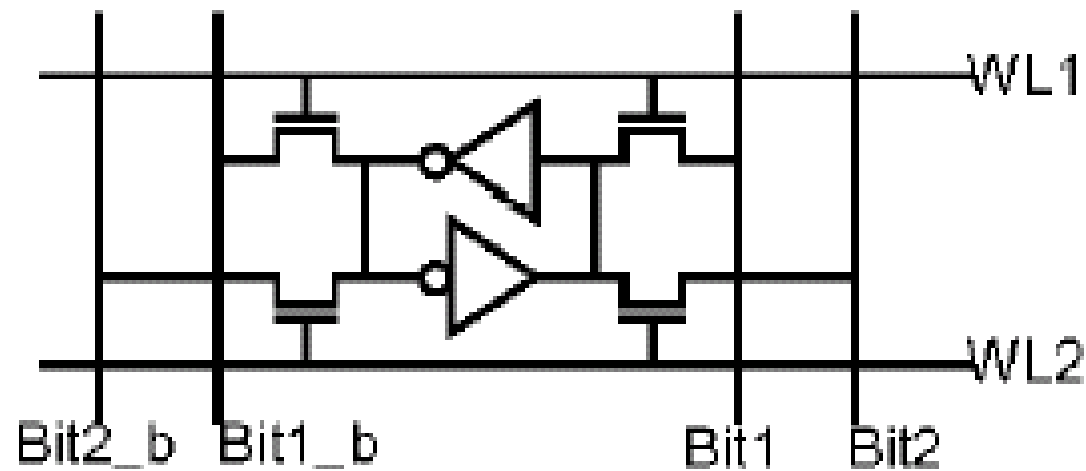
SRAM: How it works

- Two modes of operation that share lines
 - Read and Write
- Read Mode:
 - Both bit-lines must start out high, one of the two lines gets but pulled low (the pull-down is stronger than the bit-line) and this is how the result is read.
- Write Mode
 - In order to perform a read, one bit line is set high and the other is set low. The pull down overpowers the value stored in the cell and you data is now stored in the cell

SRAM Cell



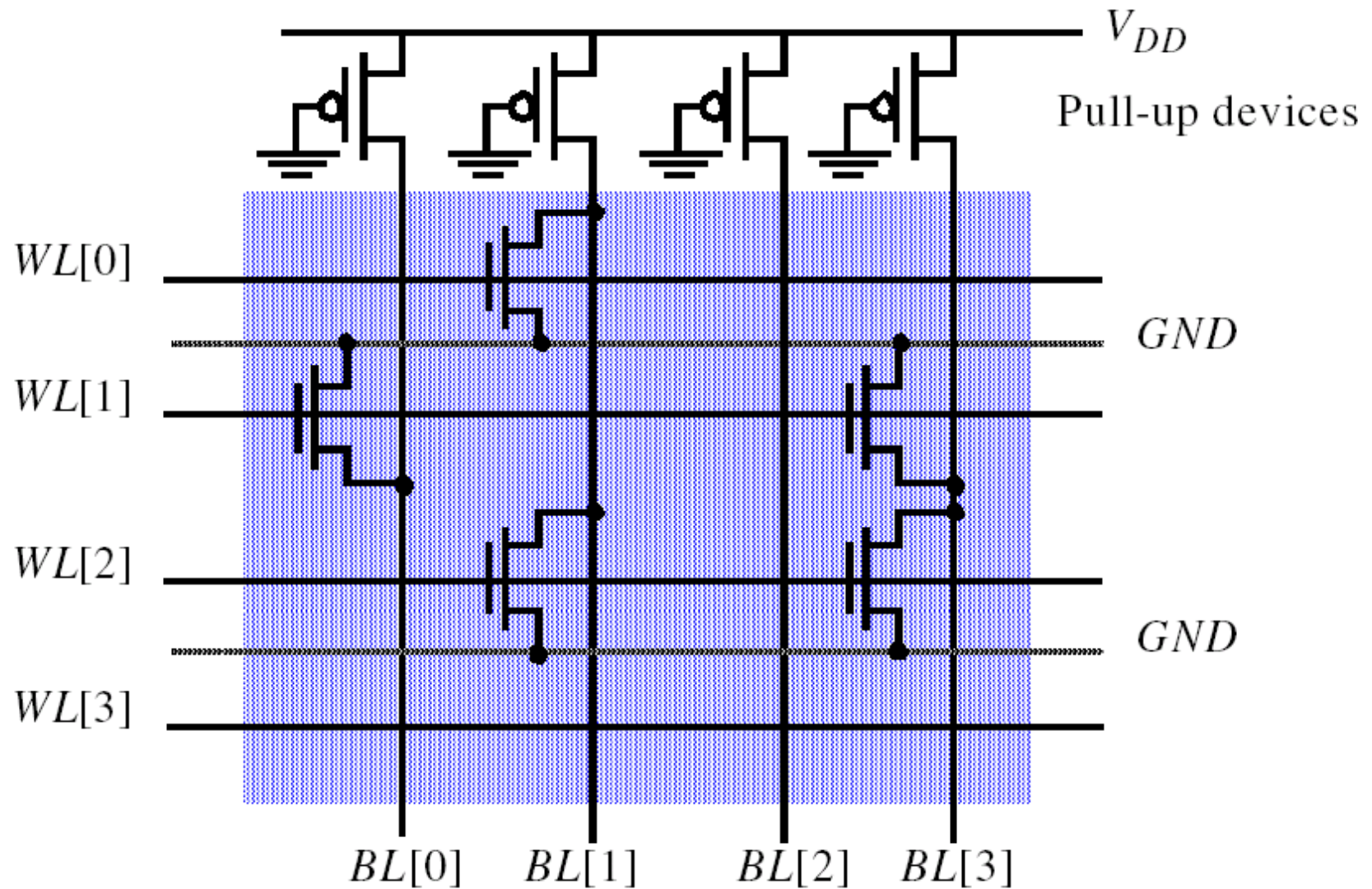
SRAM 2-Port



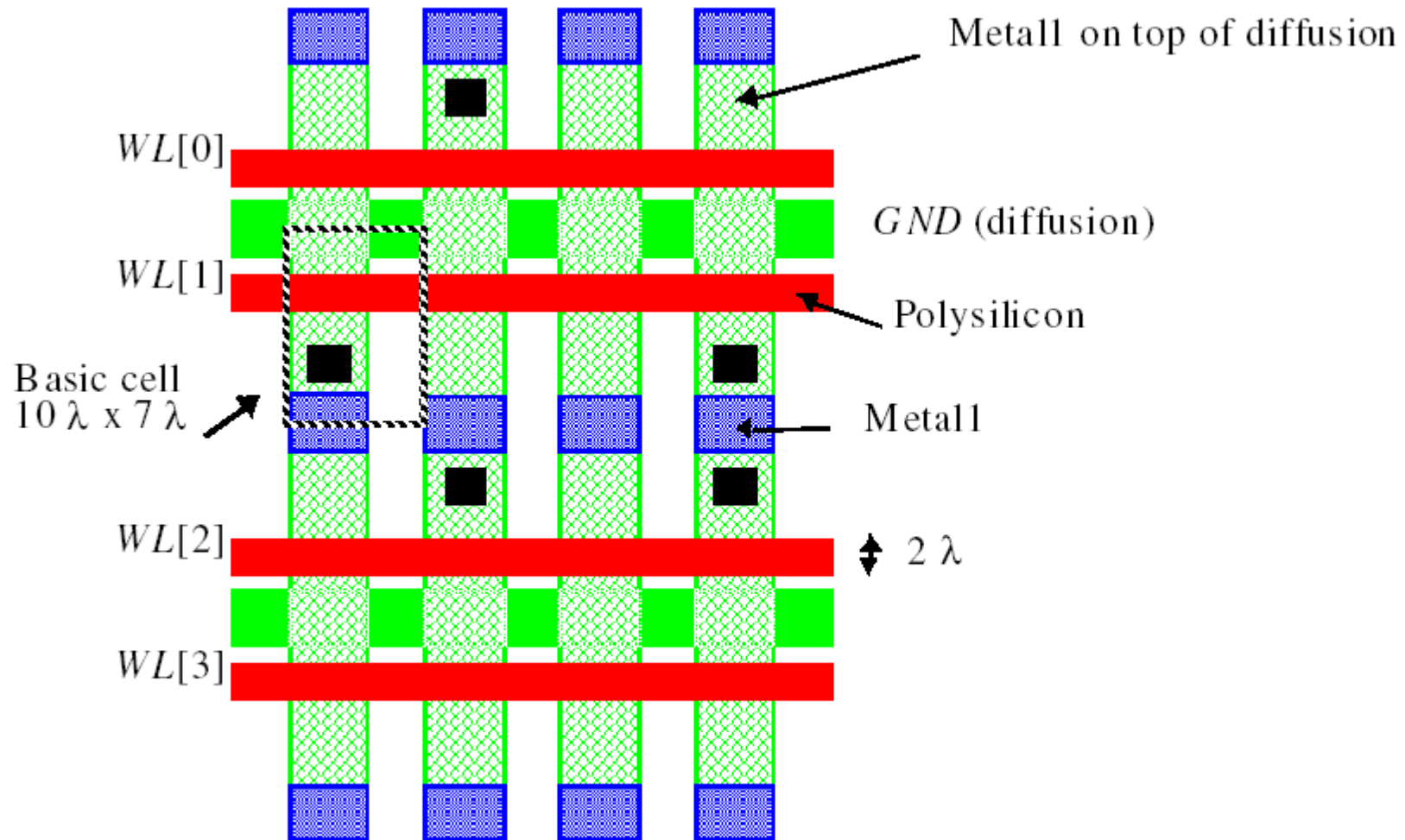
ROM

- SRAM is big, 6-T for a single cell
- If we know at fabrication time what the contents will be, and we don't need to write them what can we do better?
 - Hard wire the values into the memory
- Very dense layout
- Very fast operation
- Two types of ROM
 - Mask programmable
 - Diffusion programmable

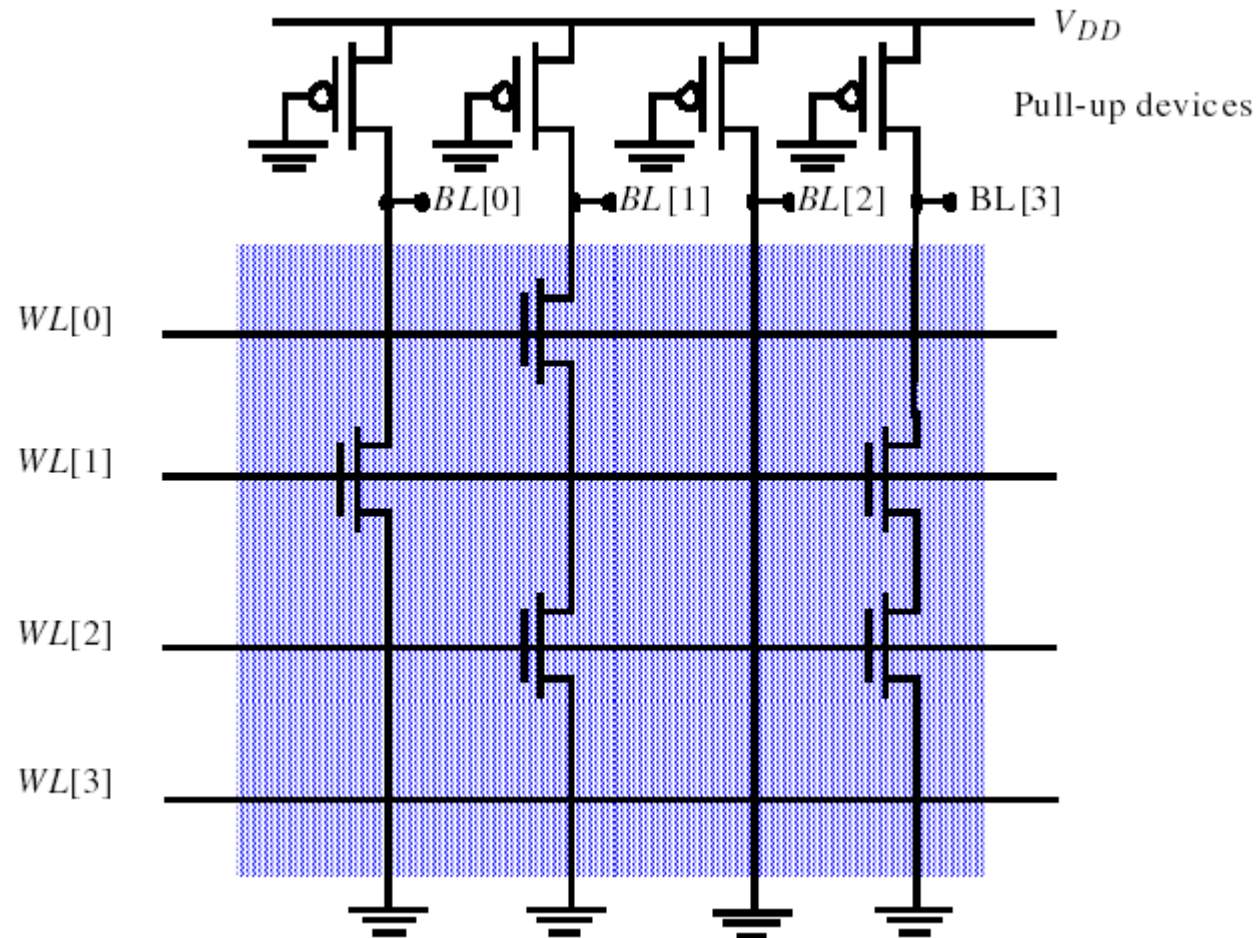
NOR ROM



NOR ROM Layout

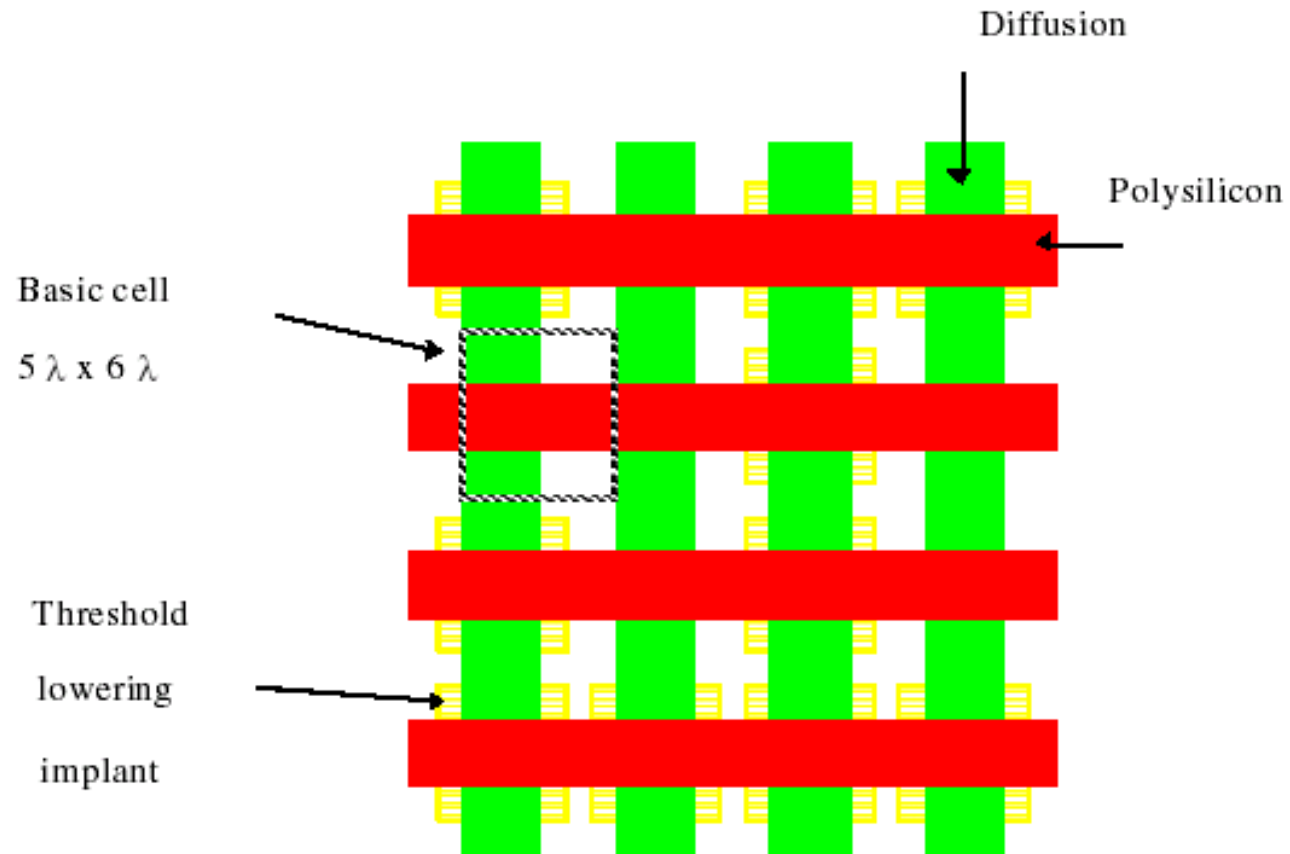


NAND ROM



All WL high except selected row

NAND ROM Layout



Memory Comparison

- All available today

1-Port SRAM	4k bit	8k bit	16k bit	32k bit	64k bit
Area (mm ²)	0.44	0.7	1.24	2.18	3.93
Delay (ns)	2.83	2.96	3.05	3.86	4.68
Power (mA/MHz)	0.15	0.175	0.18	0.226	0.285

2-Port SRAM	4k bit	8k bit	16k bit	32k bit	64k bit
Area	170%	180%	180%	173%	182%
Delay	118%	116%	116%	114%	100%
Power	159%	146%	155%	160%	141%

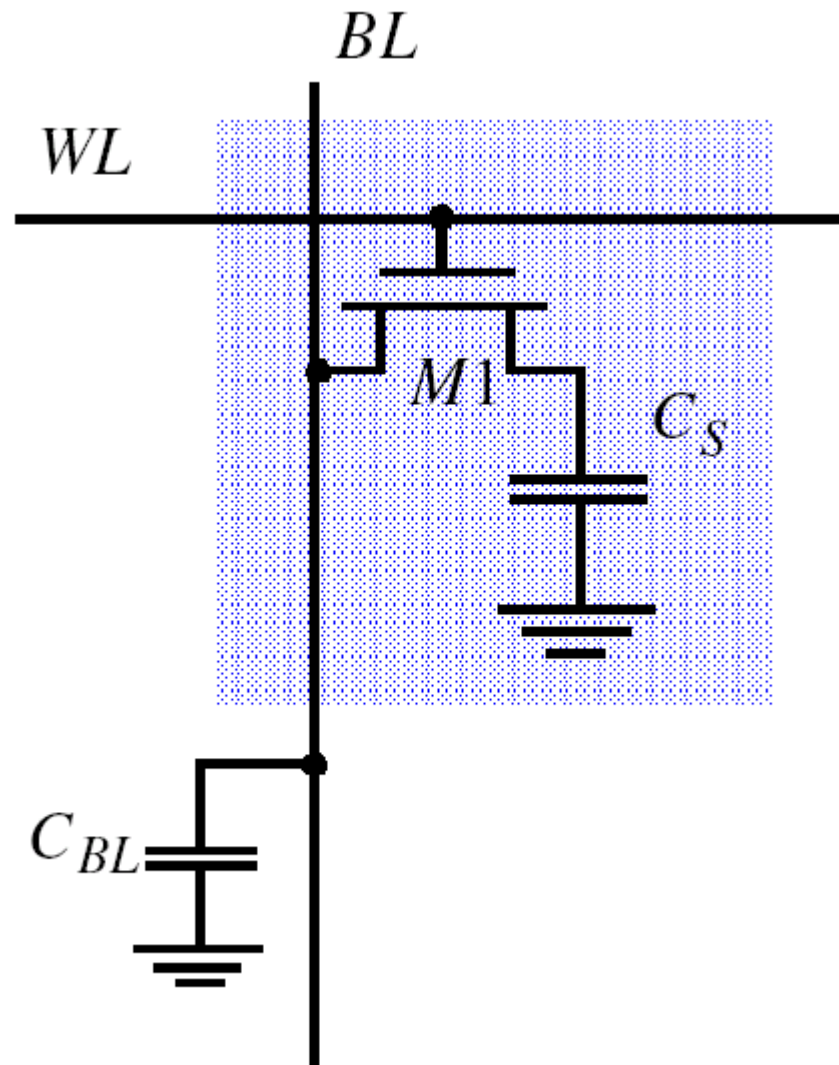
Diffused ROM	4k bit	8k bit	16k bit	32k bit	64k bit
Area	27%	23%	18%	15%	13%
Delay	101%	106%	107%	95%	81%
Power	59%	69%	77%	76%	100%

0.35um CMOS (austria microsystems)

Memory Technology: DRAM

- All we need to do is store a bit
 - then read it out right
- Instead of storing the bit in a transistor
 - Use a capacitor
- DRAMs use a transistor to guard a capacitor
 - Put data in, and disconnect it
 - Reconnect and read it out

DRAM Cell



DRAM Details

- Pros

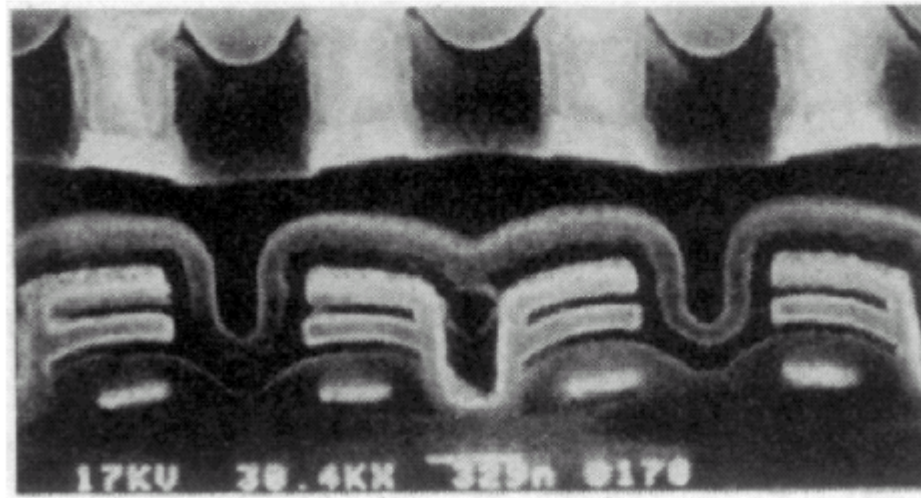
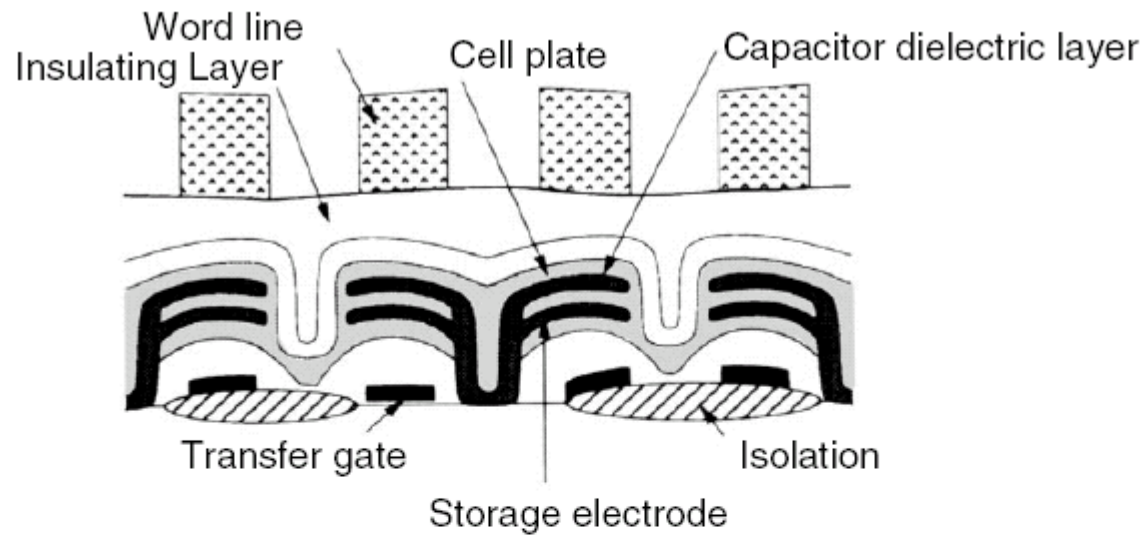
Tiny design: 1 Transistor + 1 capacitor

- One one word line and one bit line
- In reality these to are even built on top of one another

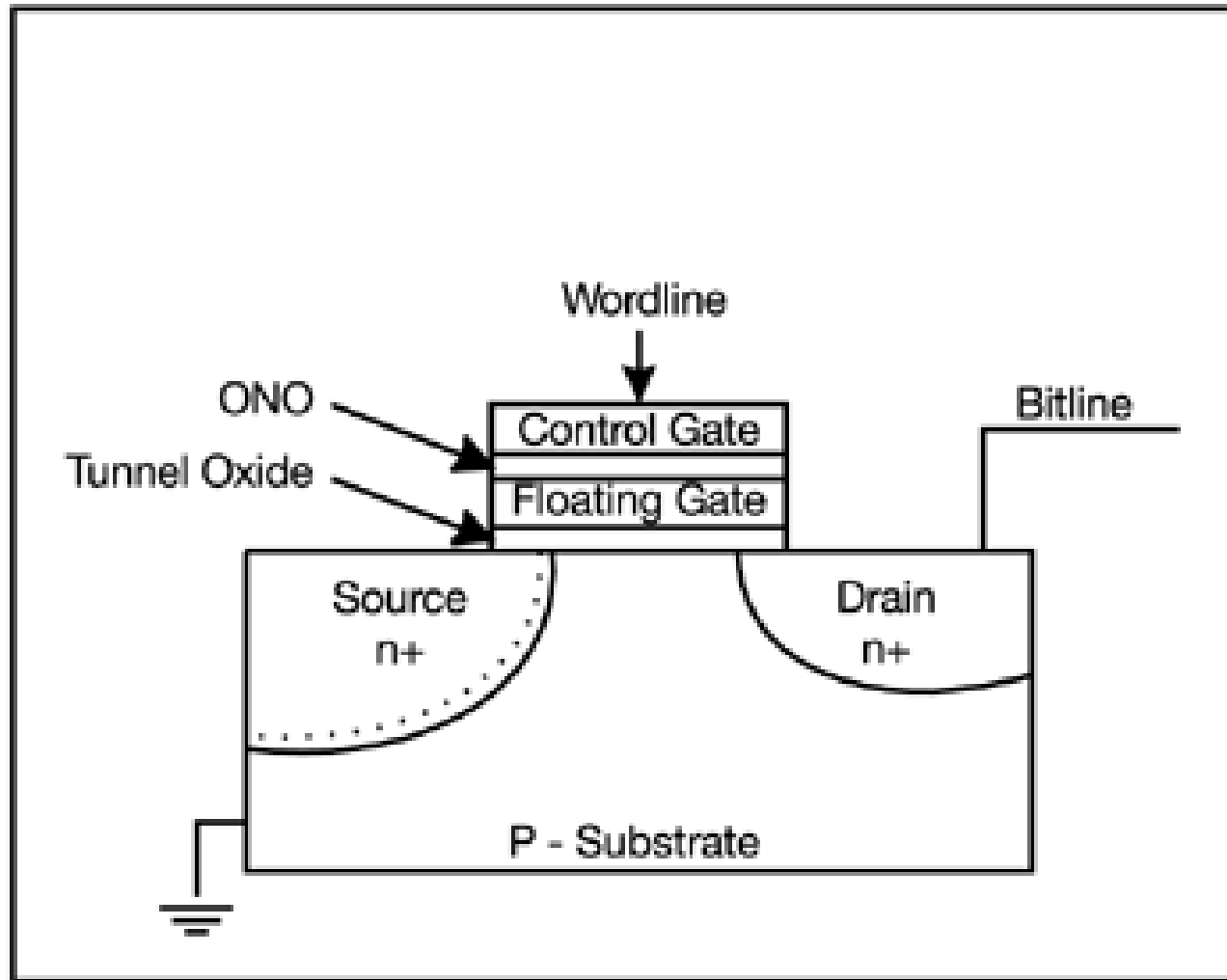
- Cons

- Capacitor Design: Very small, high cap, 3D structures
- Sense Amp Design: Small voltage ($\sim 100e$)
- Destructive Read: Spill the entire contents onto bit-line
- Refresh: Puts the Dynamic in Dynamic RAM
- Speed: Slower (tiny voltages, built for density)

DRAM structures



Flash



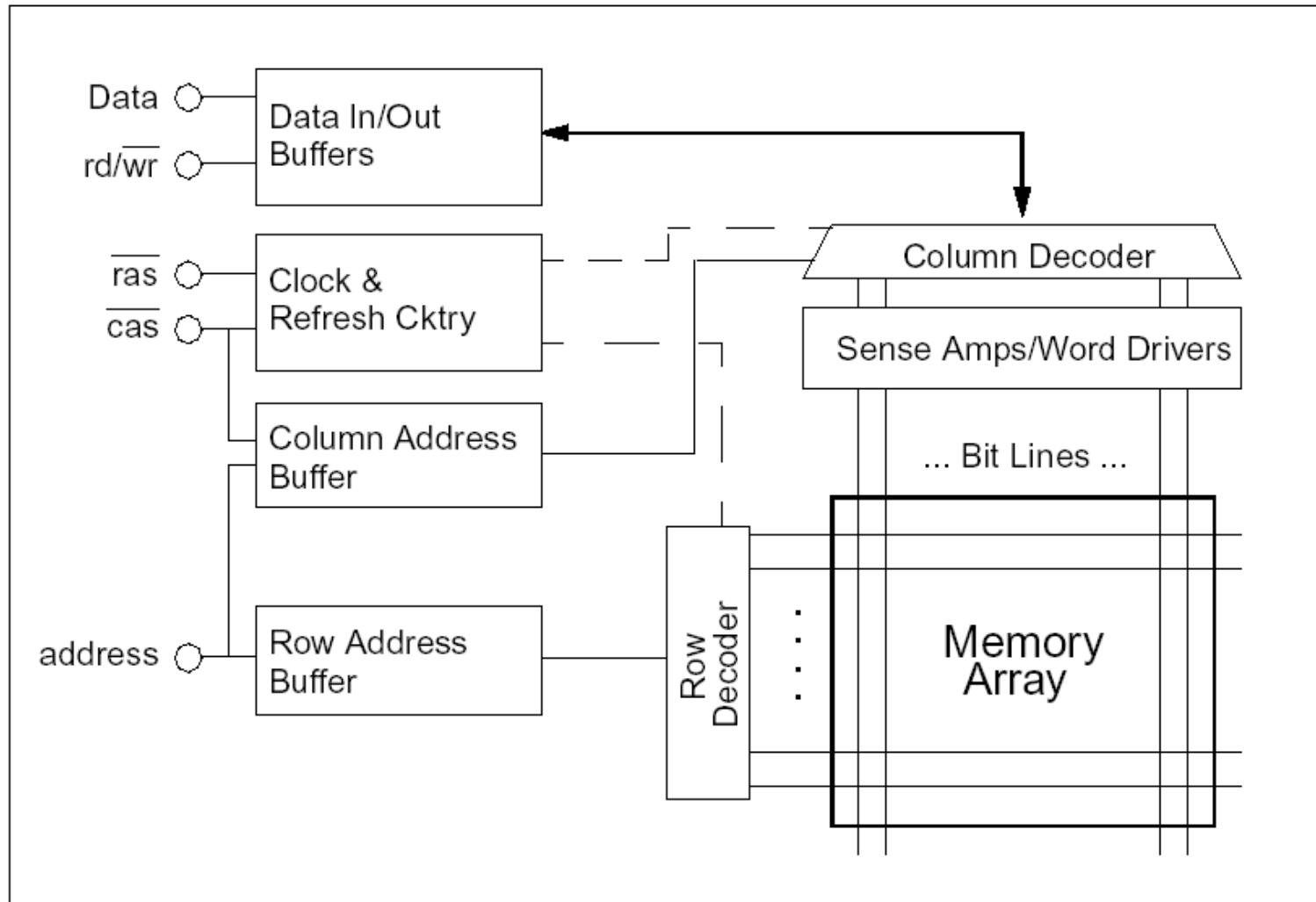
Flash Details

- Pros:
 - High density
 - Writable and Readable
 - Store multiple bits per cell (multi-valued)
 - Non-volatile
 - Increasingly available for SOC
- Cons:
 - Slow (90ns)
 - Field erase
 - Have to erase a whole section of the chip
 - Erase sets the bits to '1', writing to them can turn them to '0'

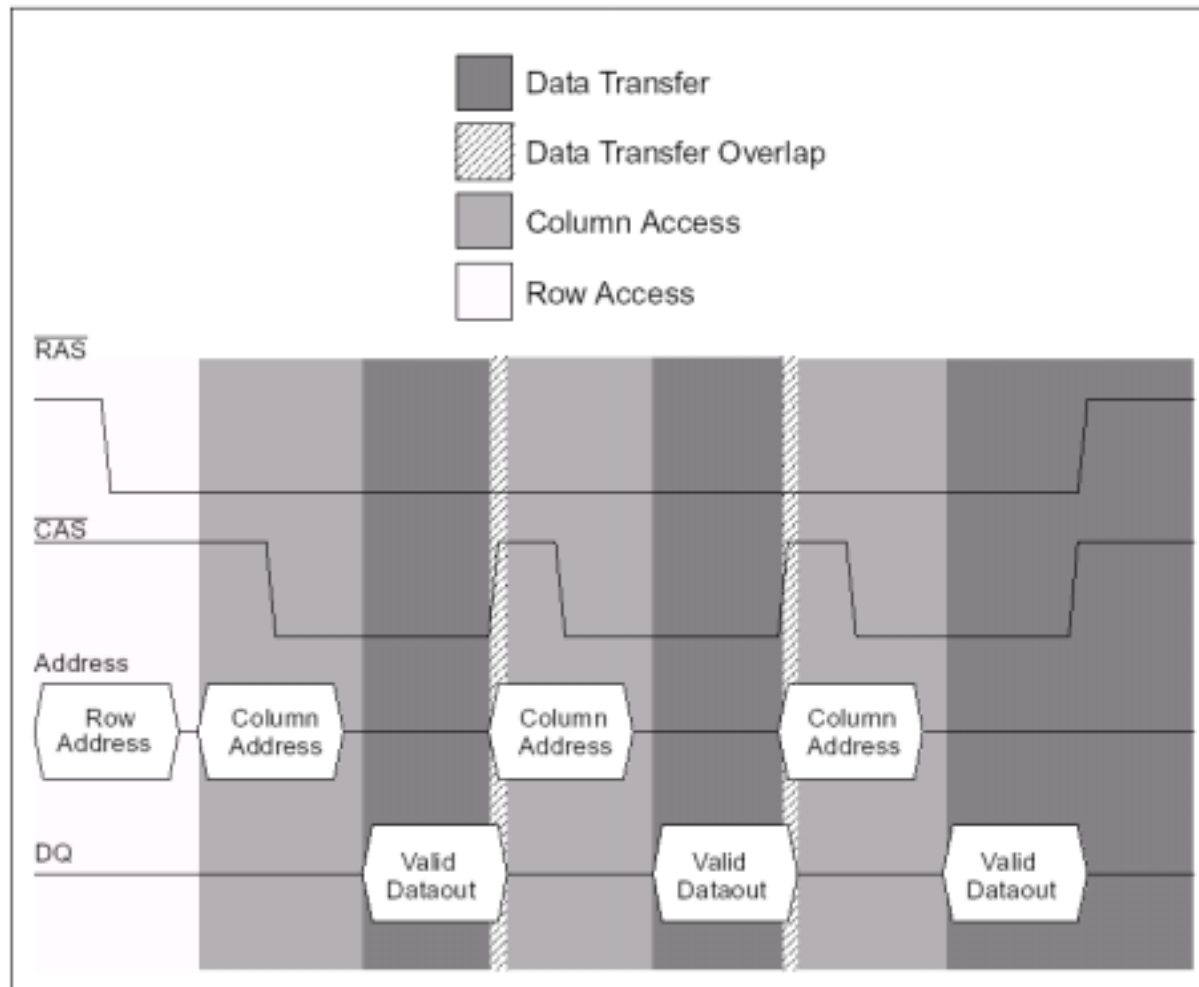
DRAM Memory Architectures

- Even if the memory arrays are the same
 - The interface and interconnect can be different
- Discuss 4 types of DRAM architectures
 - FPM, EDO, SDRAM, Rambus
- All are built on the same basic memory tile
 - Difference lies in how data is accessed
 - And how the bus is designed
- All need to send a row address and a column address

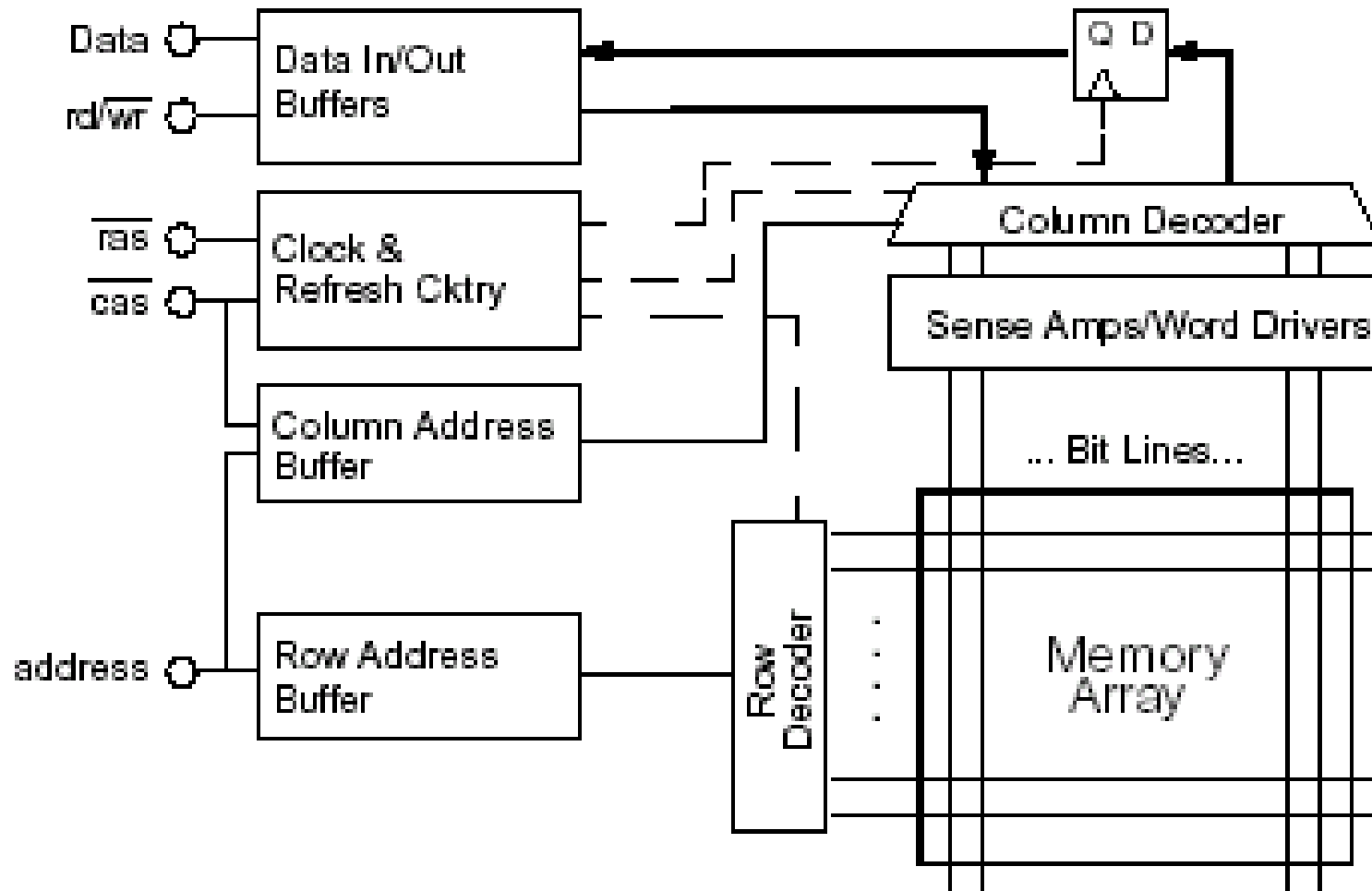
FPM



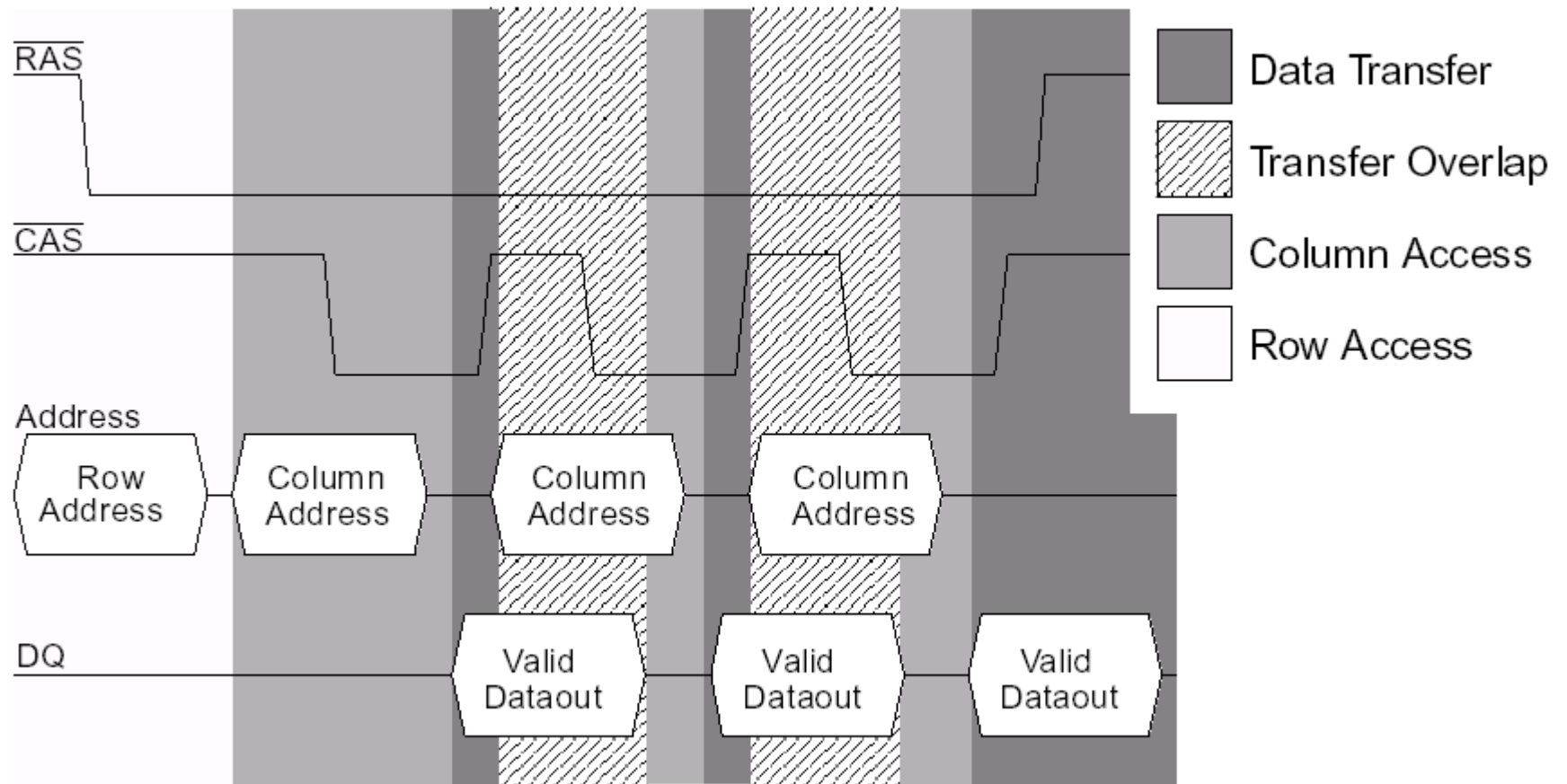
FPM Timing



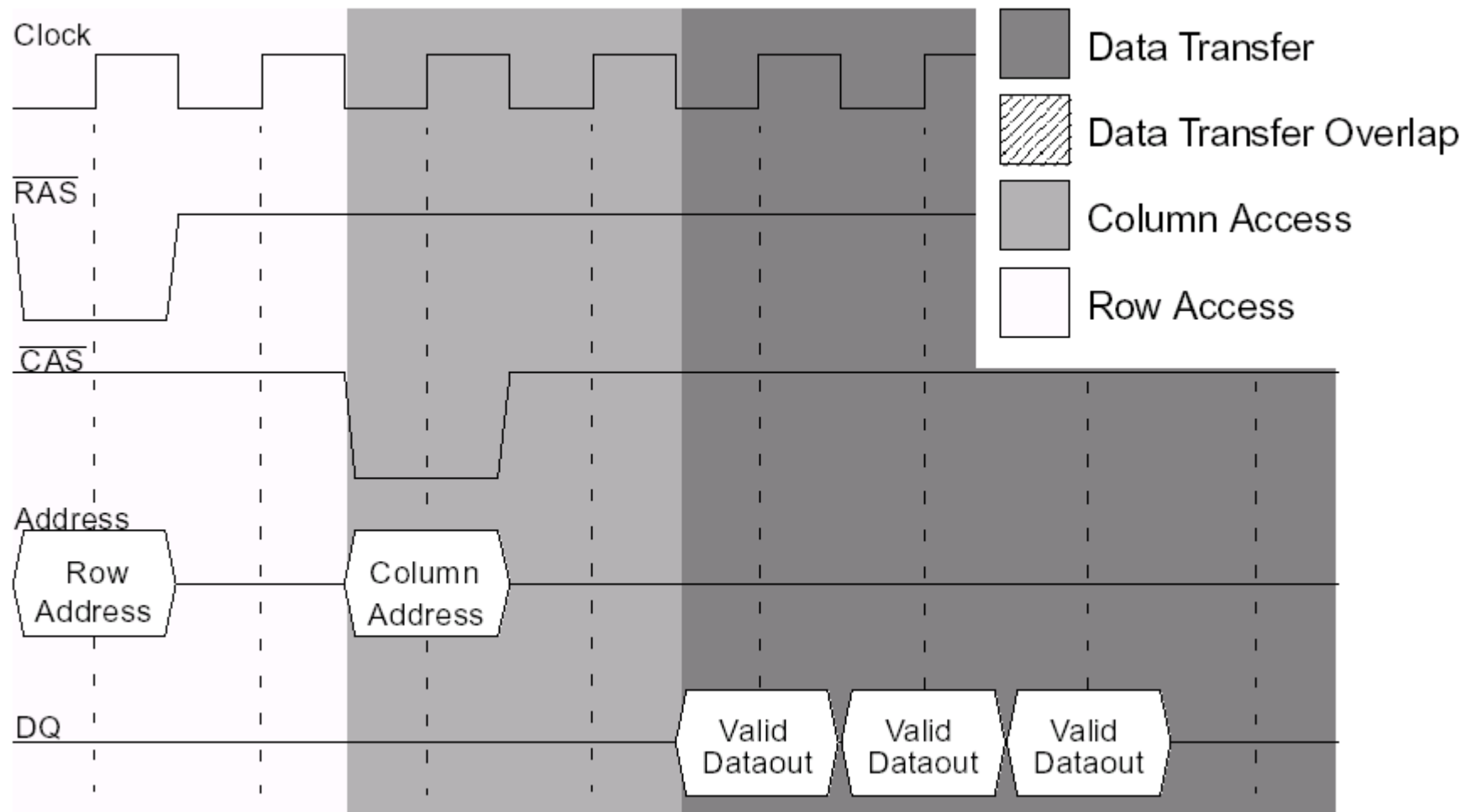
EDO



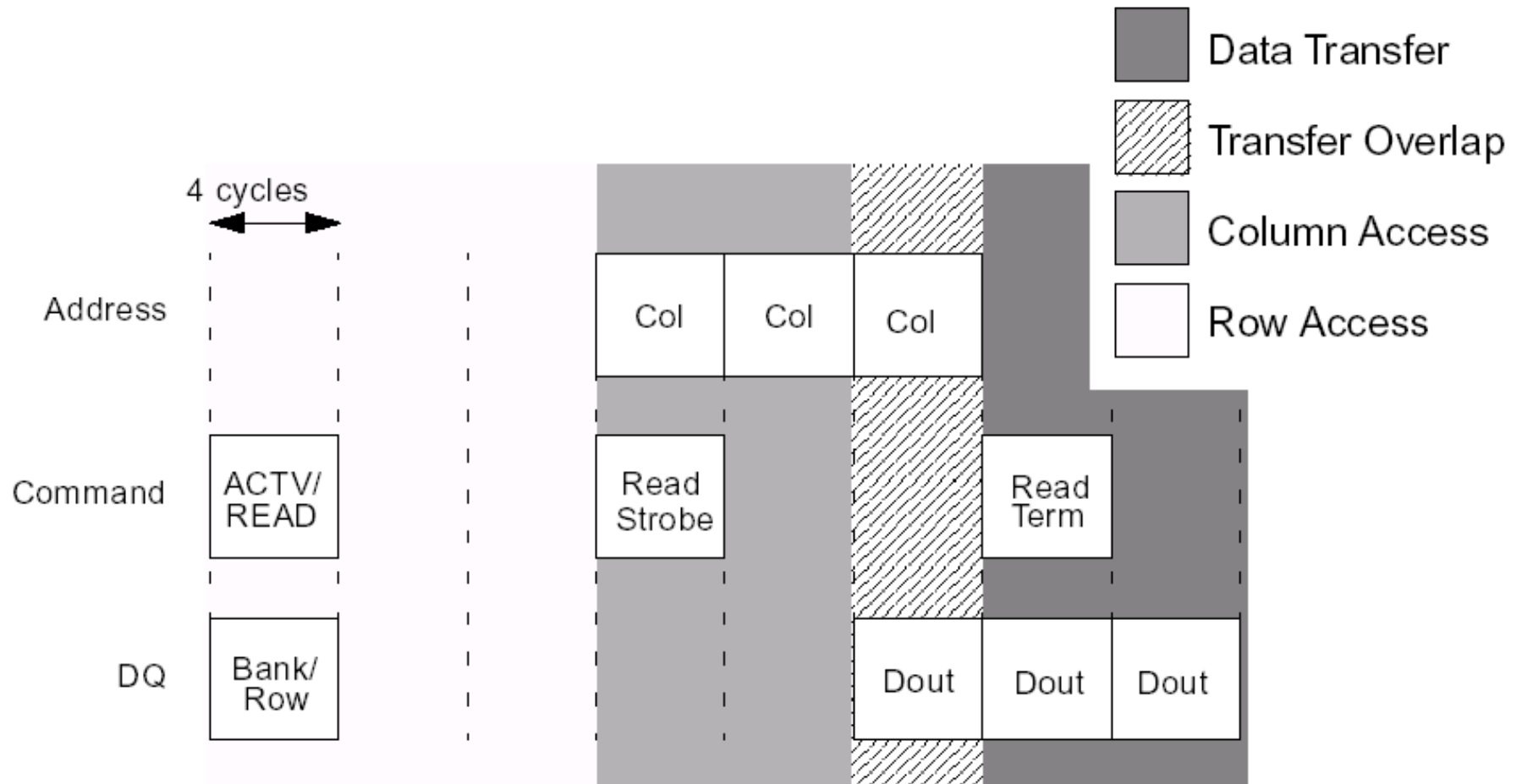
EDO Timing



SDRAM Timing

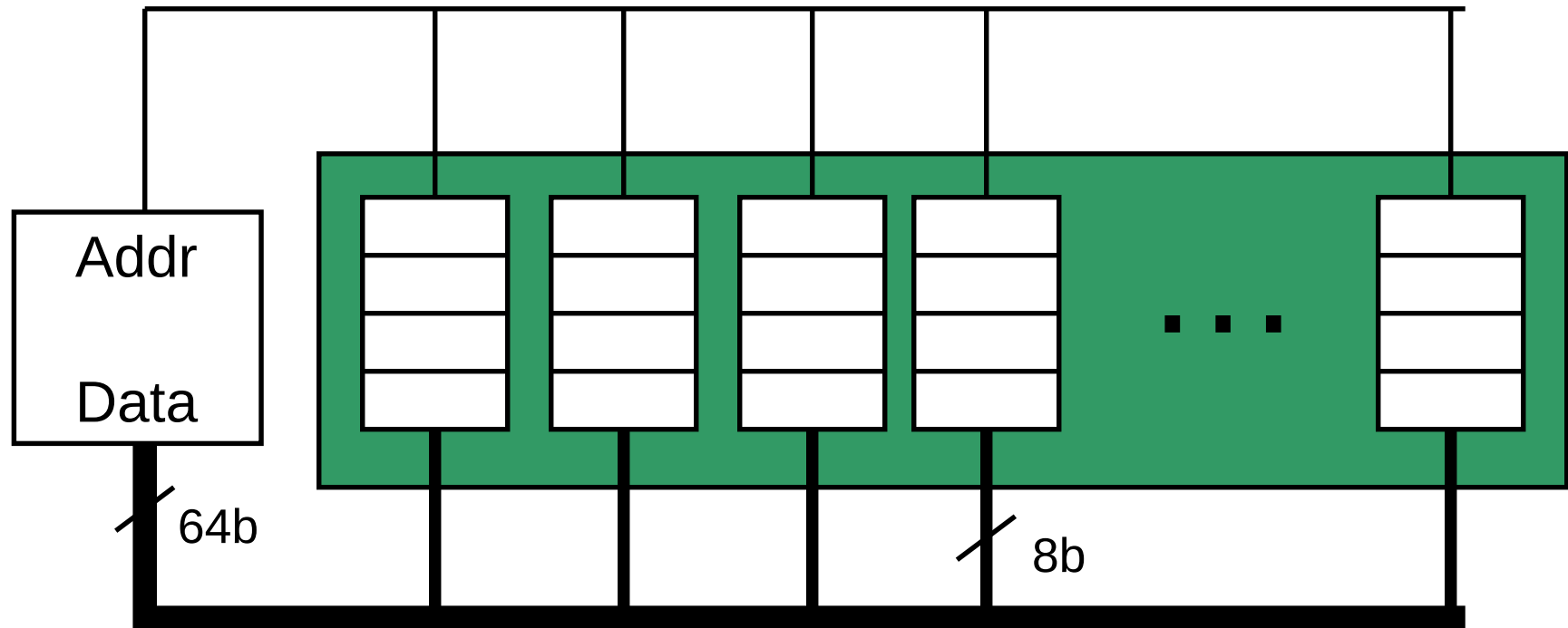


Rambus



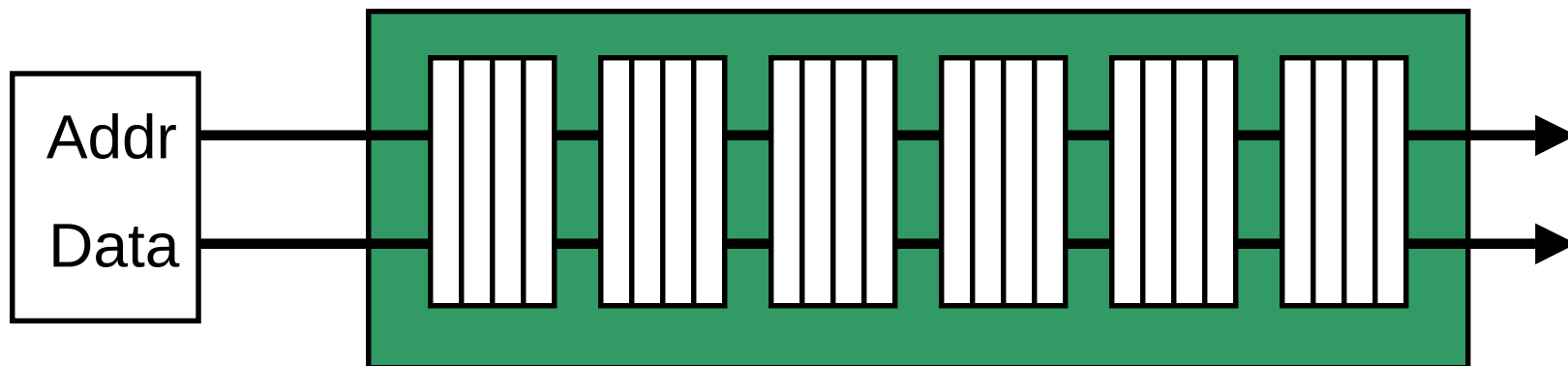
SDRAM

- Large bus (64 bits)
- 4 Banks per device, 4 banks total
 - Banks are across the devices



Rambus

- Address and Data are uniformly loaded
- 4 banks per device, 32 banks total (additive)
- Reduces bank conflict probability



Summary

- Now you should know (at least at a high level)
 - How a SRAM works
 - How a ROM works
 - How a DRAM works
 - How Flash memory works
 - How to pick between them for your designs
 - What the differences are between
 - FPM, EDO, SDRAM, and Rambus