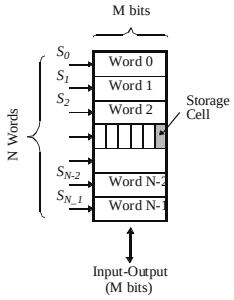
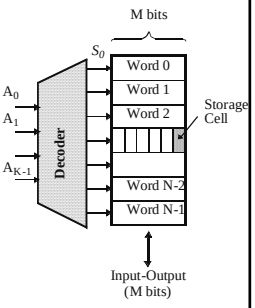
Memory

RWM		NVRWM	ROM
Random Access	**Non-Random Access**		
SRAM DRAM	FIFO LIFO Shift Register CAM	EPROM E^2PROM FLASH	Mask-Programmed Programmable (PROM)

Memory Decoders



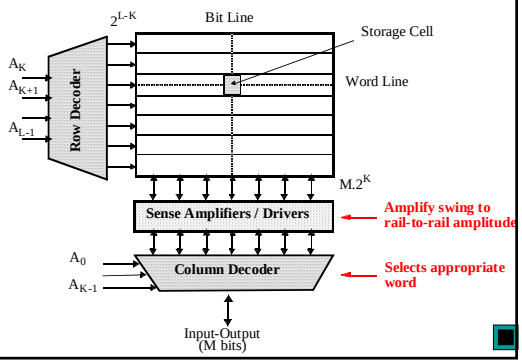
N words => N select signals
Too many select signals



Decoder reduces # of select signals
K = log_2N

Array-Structured Memory

Problem: ASPECT RATIO or HEIGHT >> WIDTH



Array Decoding

- Typically want an aspect ratio that is not too far from square
- How to divide up the row, column address decoding?

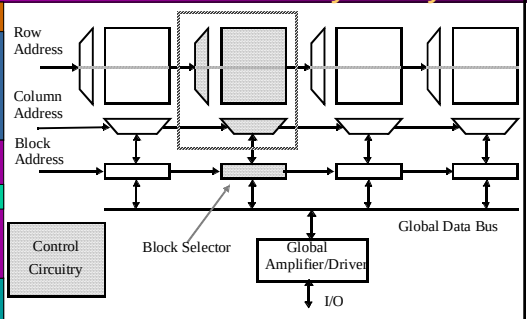
Use an 8K x 32 SRAM = 256 Kb = 2^{18}

2^{18} = 2^9 rows x 2^9 columns

Row decoder is 9 to 512 decoder

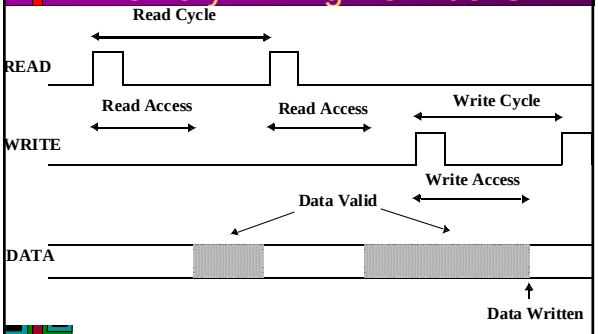
Every 32 (2^5) columns is a 'word', and we only need to decode words. So, column decoder needs to decode 2^4 words, so need a 4 to 16 column decoder.

Hierarchical Memory Arrays

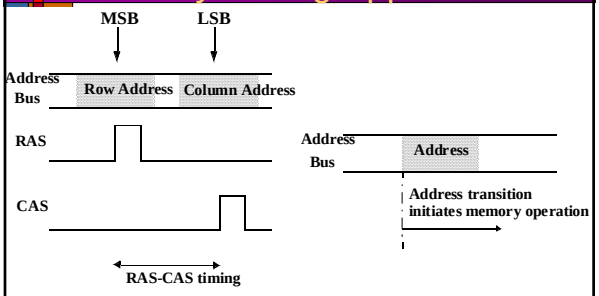


Advantages:
1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Memory Timing Definitions



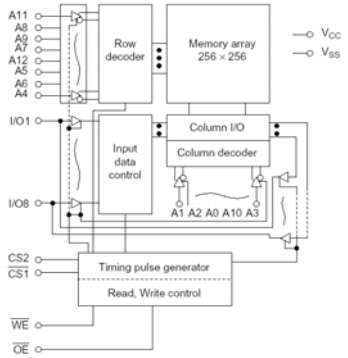
Memory Timing Approaches



DRAM Timing
Multiplexed Adressing

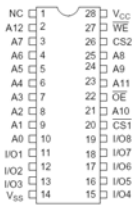
SRAM Timing
Self-timed

Example: HM6264 8kx8 SRAM



HM6264 Interface

HM6264BLP/BLSP/BLFP Series



(Top view)

Pin Description

Pin name	Function	Pin name	Function
A0 to A12	Address input	$\overline{WE}$	Write enable
I/O1 to I/O8	Data input/output	$\overline{OE}$	Output enable
$\overline{CS1}$	Chip select 1	NC	No connection
CS2	Chip select 2	V_{CC}	Power supply
		V_{SS}	Ground

Function Table

Function Table

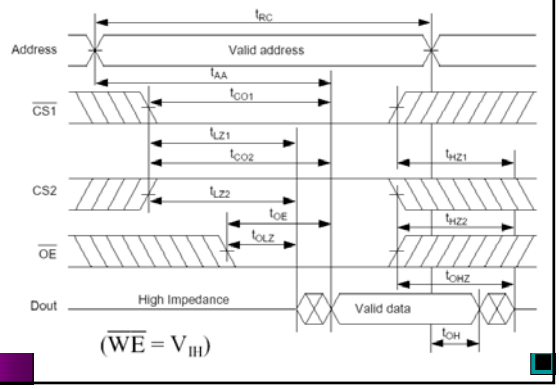
$\overline{WE}$	$\overline{CS1}$	CS2	$\overline{OE}$	Mode	V_{CC} current	I/O pin	Ref. cycle
×	H	×	×	Not selected (power down)	I_{SB}, I_{SB1}	High-Z	—
×	×	L	×	Not selected (power down)	I_{SB}, I_{SB1}	High-Z	—
H	L	H	H	Output disable	I_{CC}	High-Z	—
H	L	H	L	Read	I_{CC}	Dout	Read cycle (1)–(3)
L	L	H	H	Write	I_{CC}	Din	Write cycle (1)
L	L	H	L	Write	I_{CC}	Din	Write cycle (2)

Note: ×: H or L

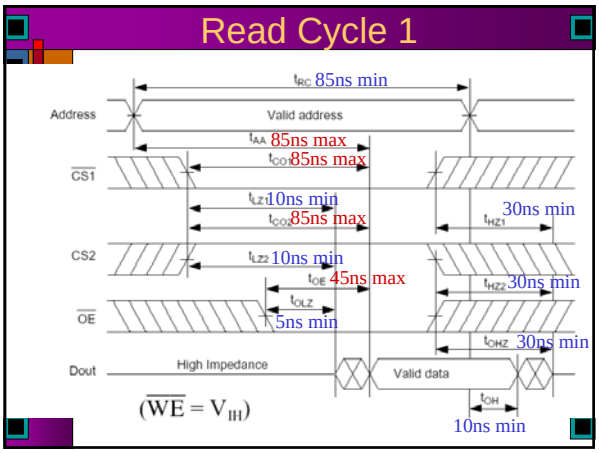
Timing

Parameter		Symbol	HM6264B-8L Min	HM6264B-8L Max	HM6264B-10L Min	HM6264B-10L Max	Unit	Notes
Read cycle time		t_{RC}	85	—	100	—	ns	
Address access time		t_{AA}	—	85	—	100	ns	
Chip select access time	$\overline{CS1}$	t_{CO1}	—	85	—	100	ns	
	CS2	t_{CO2}	—	85	—	100	ns	
Output enable to output valid		t_{OE}	—	45	—	50	ns	
Chip selection to output in low-Z	$\overline{CS1}$	t_{LZ1}	10	—	10	—	ns	2
	CS2	t_{LZ2}	10	—	10	—	ns	2
Output enable to output in low-Z		t_{OLZ}	5	—	5	—	ns	2
Chip deselection in to output in high-Z	$\overline{CS1}$	t_{HZ1}	0	30	0	35	ns	1, 2
	CS2	t_{HZ2}	0	30	0	35	ns	1, 2
Output disable to output in high-Z		t_{OHZ}	0	30	0	35	ns	1, 2
Output hold from address change		t_{OH}	10	—	10	—	ns	

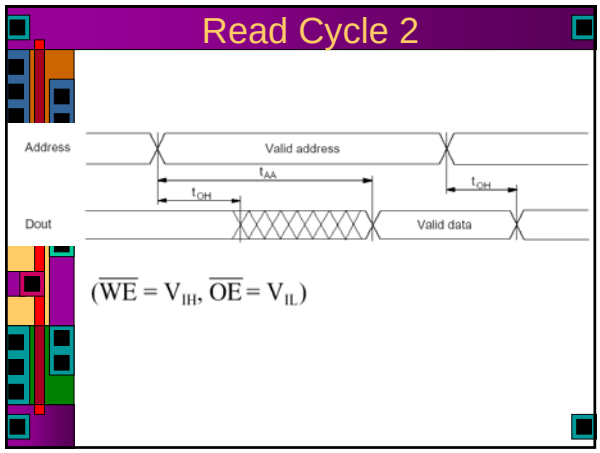
Read Cycle 1



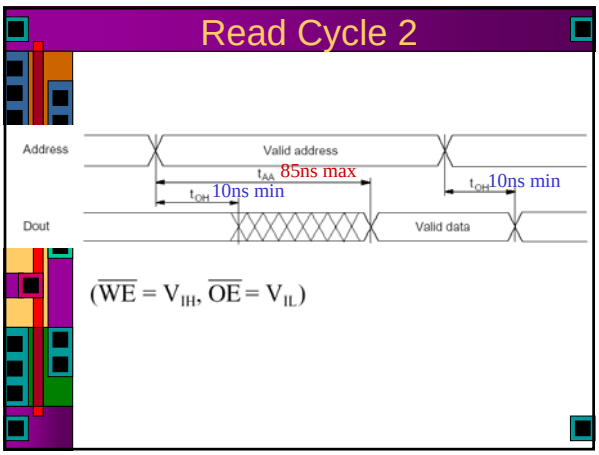
Read Cycle 1



Read Cycle 2



Read Cycle 2

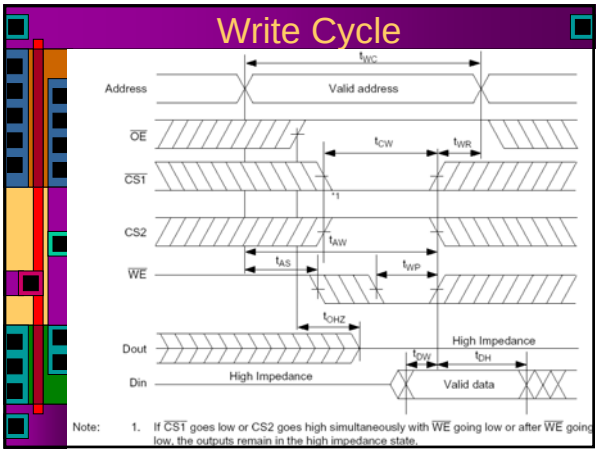


Write Timing

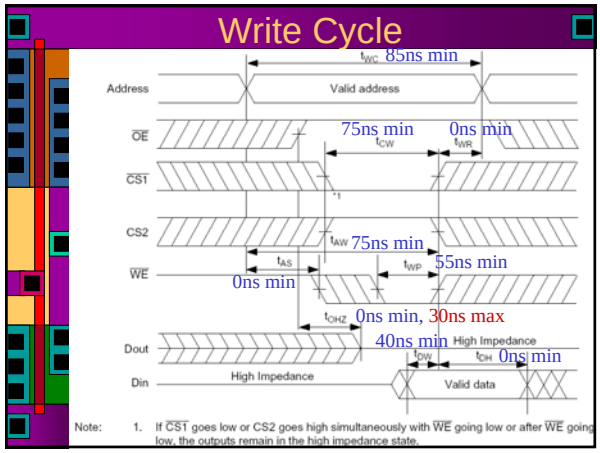
Parameter	Symbol	HM6264B-8L Min	HM6264B-8L Max	HM6264B-10L Min	HM6264B-10L Max	Unit	Notes
Write cycle time	t_{WC}	85	—	100	—	ns	
Chip selection to end of write	t_{CW}	75	—	80	—	ns	2
Address setup time	t_{AS}	0	—	0	—	ns	3
Address valid to end of write	t_{AW}	75	—	80	—	ns	
Write pulse width	t_{WP}	55	—	60	—	ns	1, 6
Write recovery time	t_{WR}	0	—	0	—	ns	4
$\overline{WE}$ to output in high-Z	t_{WHZ}	0	30	0	35	ns	5
Data to write time overlap	t_{DW}	40	—	40	—	ns	
Data hold from write time	t_{DH}	0	—	0	—	ns	
Output active from end of write	t_{OW}	5	—	5	—	ns	
Output disable to output in high-Z	t_{OHZ}	0	30	0	35	ns	5

Notes: 1. A write occurs during the overlap of a low $\overline{CS1}$, and high CS2, and a high $\overline{WE}$. A write begins at the latest transition among $\overline{CS1}$ going low,CS2 going high and $\overline{WE}$ going low. A write ends at the earliest transition among $\overline{CS1}$ going high CS2 going low and $\overline{WE}$ going high. Time t_{WP} is measured from the beginning of write to the end of write.

Write Cycle



Write Cycle



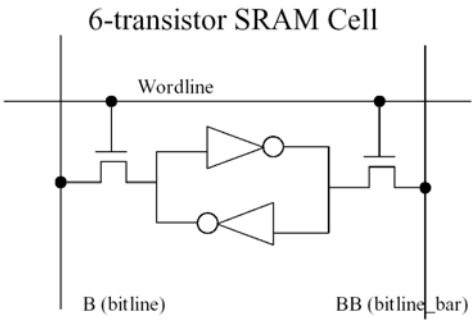
What Does All This Mean

- For a read:
  - If you assert CS1, CS2, address, and OE all at the same time, it will be max 85ns before valid data are available at chip outputs
- For a write:
  - You can assert CS1, CS2, address, data, and WE all at the same time if you want to
  - You need to wait 55ns from WE edge, or 75ns from CS1/CS2 edge for write to have happened

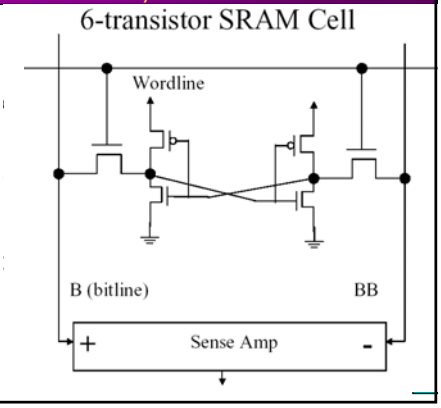
R/W Memories In General

- **STATIC (SRAM)**
  - **Data stored as long as supply is applied**
  - **Large (6 transistors/cell)**
  - **Fast**
  - **Differential**
- **DYNAMIC (DRAM)**
  - **Periodic refresh required**
  - **Small (1-3 transistors/cell)**
  - **Slower**
  - **Single Ended**

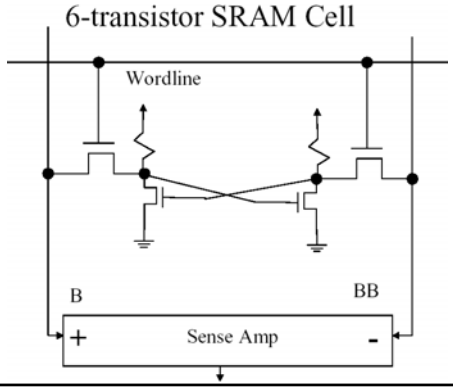
SRAM Circuits



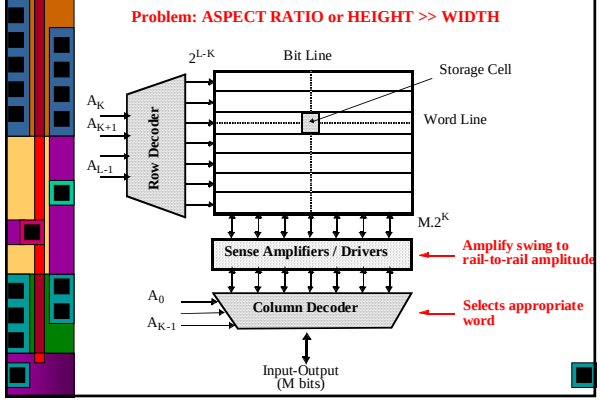
SRAM Cell, Transistors



SRAM, Resistive Pullups

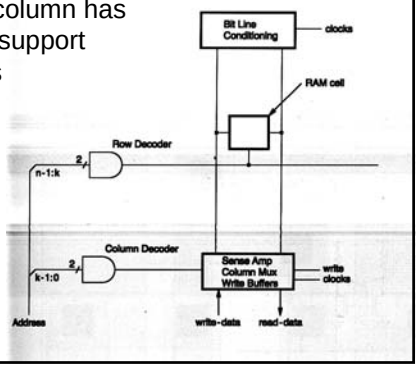


Array-Structured Memory

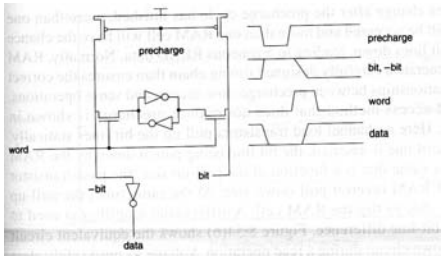


Memory Column

- Each column has all the support circuits

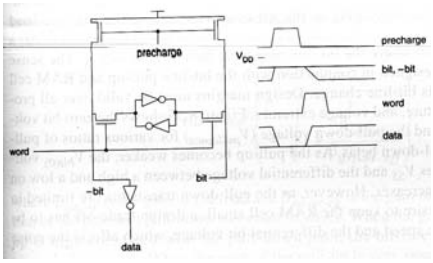


Reading the Bit



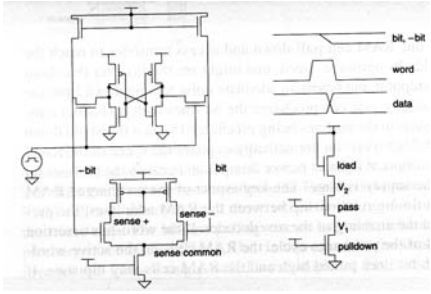
- Single-ended read using an inverter
- Dynamic pre-charge on the bit lines
  - P-types pull bit lines high

Reading the Bit 2



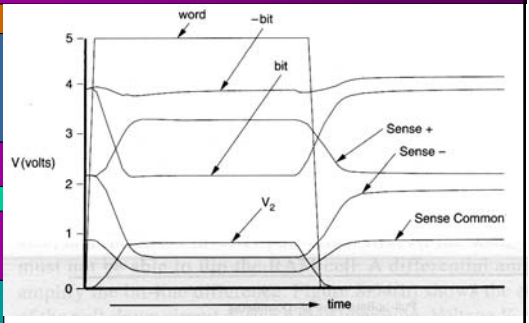
- Single-ended read using an inverter
- Dynamic pre-charge on the bit lines
  - Note the N-types used as pull-ups

Reading the Bit 3

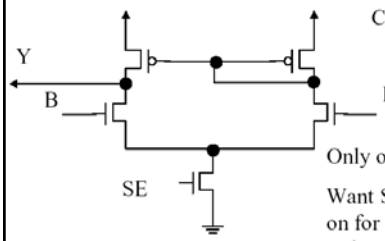


- Differential read using sense amp
- Static N-type pullup on the bit lines

Read Waveforms



Sense Amp



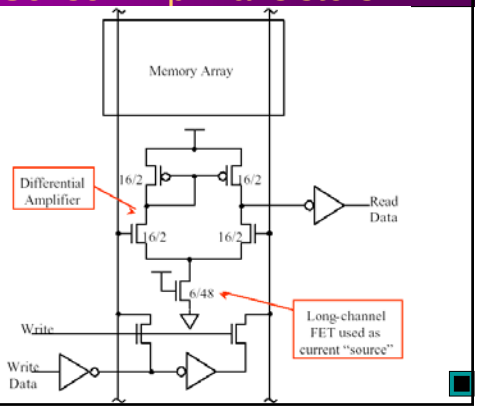
Current-mirror SA

Only one bit line will swing.

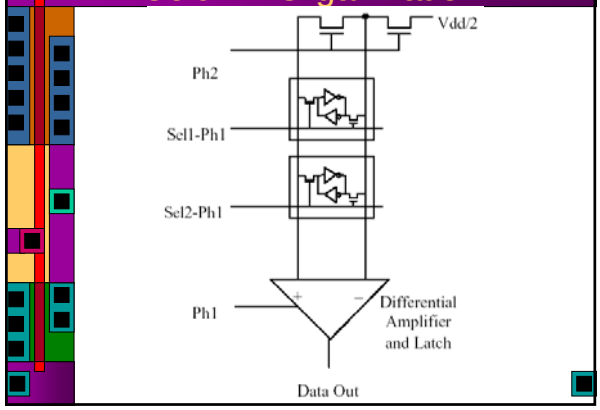
Want Sense Amplifier turned on for short amount of time in order to save power.

Job of SA is to sense bit line swing, amplify to full swing output.

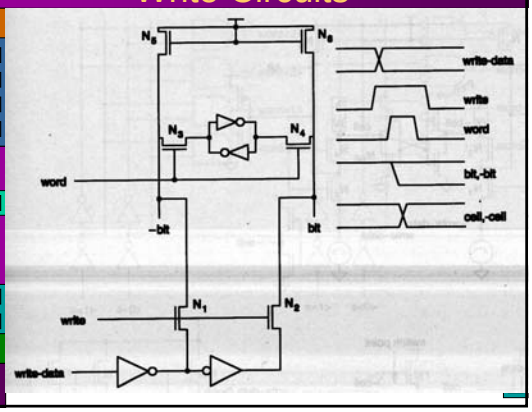
Sense Amp Transistors



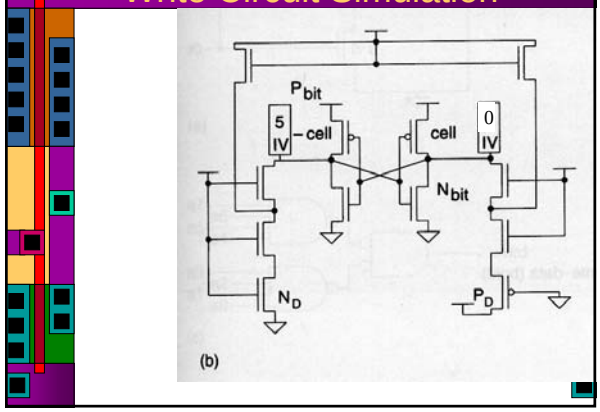
Column Organization



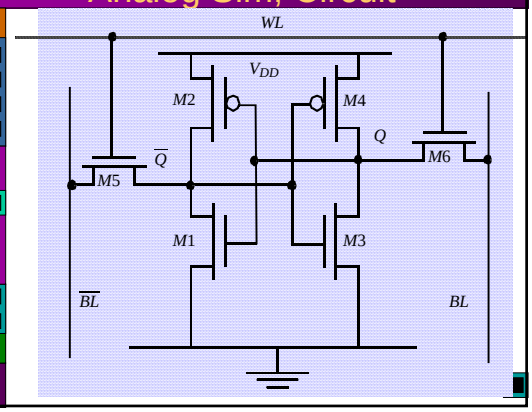
Write Circuits



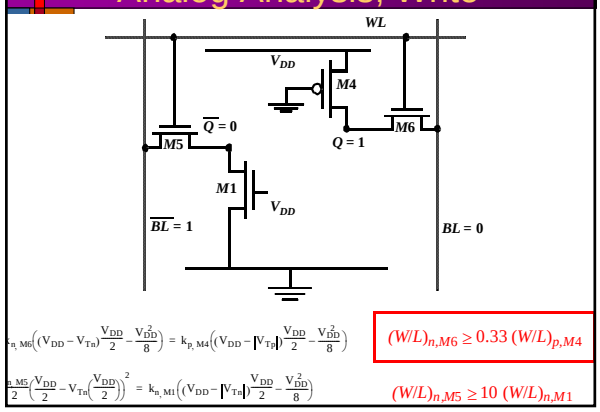
Write Circuit Simulation



Analog Sim, Circuit



Analog Analysis, Write



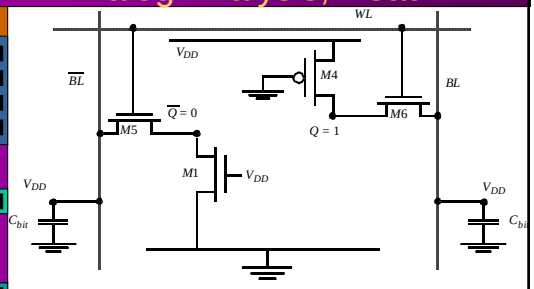
$$k_{n,M6}\left((V_{DD}-V_{Tn})\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right) = k_{p,M4}\left((V_{DD}-|V_{Tp}|)\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right)$$

$$(W/L)_{n,M6} \geq 0.33\,(W/L)_{p,M4}$$

$$\frac{k_{n,M5}}{2}\left(\frac{V_{DD}}{2}-V_{Tn}\left(\frac{V_{DD}}{2}\right)\right)^2 = k_{n,M1}\left((V_{DD}-|V_{Tn}|)\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right)$$

$$(W/L)_{n,M5} \geq 10\,(W/L)_{n,M1}$$

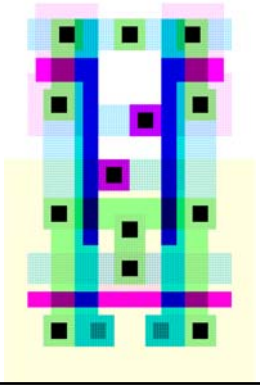
Analog Analysis, Read



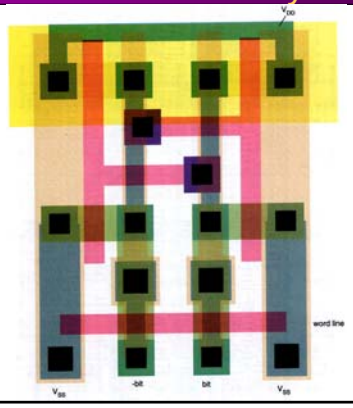
$$\frac{k_{n,M5}}{2}\left(\frac{V_{DD}}{2} - V_{Tn}\left(\frac{V_{DD}}{2}\right)\right)^2 = k_{n,M1}\left((V_{DD} - |V_{Tp}|)\frac{V_{DD}}{2} - \frac{V_{DD}^2}{8}\right)$$

$(W/L)_{n,M5} \leq 10\,(W/L)_{n,M1}$

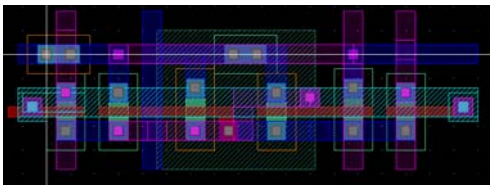
6T SRAM Layout



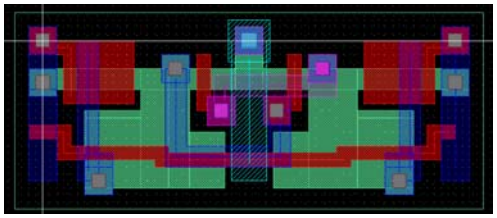
Another 6T SRAM Layout



SRAM bit from makemem (v1)

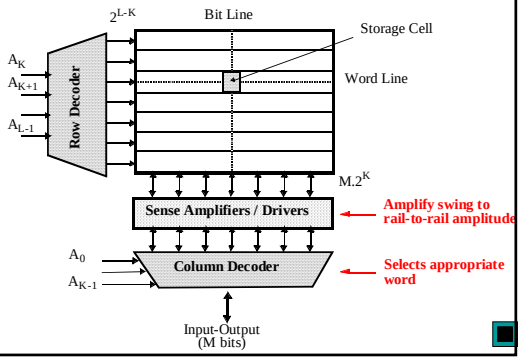


SRAM bit from makemem (v2)

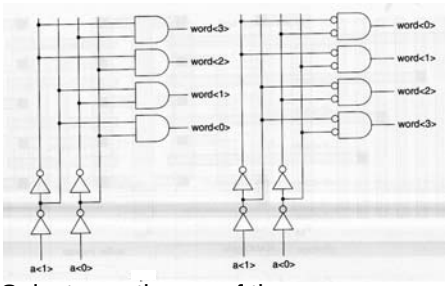


Array-Structured Memory

Problem: ASPECT RATIO or HEIGHT >> WIDTH

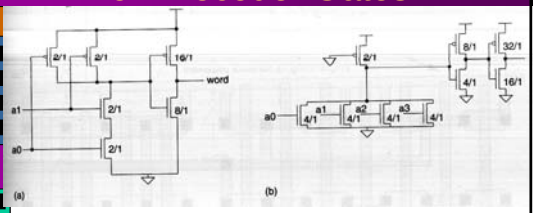


Row Decoders



- Select exactly one of the memory rows
  - Simple versions are just gates

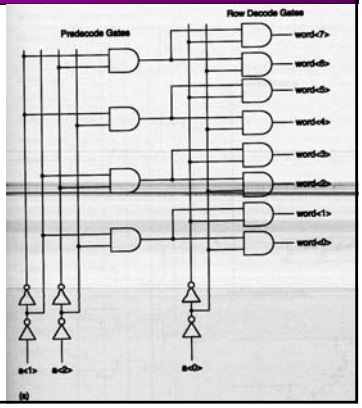
Row Decoder Gates



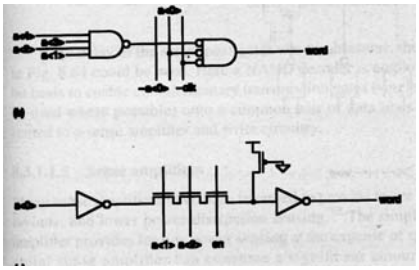
- Standard gates
- Or, pseudo-nmos gates with static pull up
  - Easier to make large fan-in NOR

Pre-decode Row Decoder

- Multiple levels of decoding can be more efficient layout



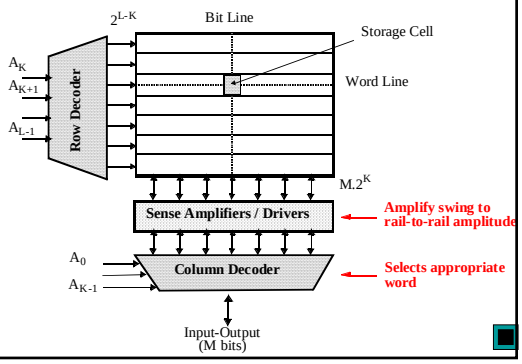
Pre-decode Row Decoder



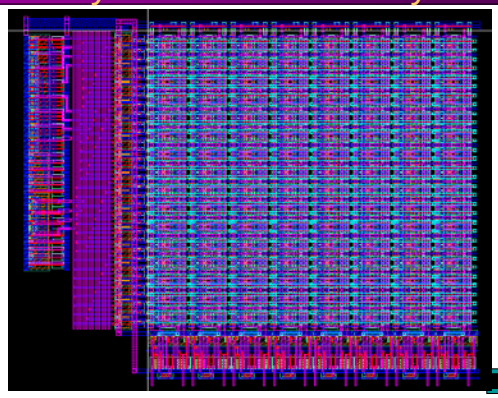
- Other circuit tricks for building row decoders…

Array-Structured Memory

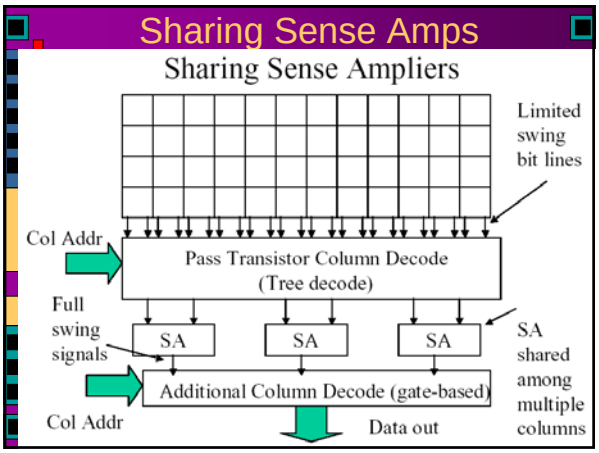
Problem: ASPECT RATIO or HEIGHT >> WIDTH



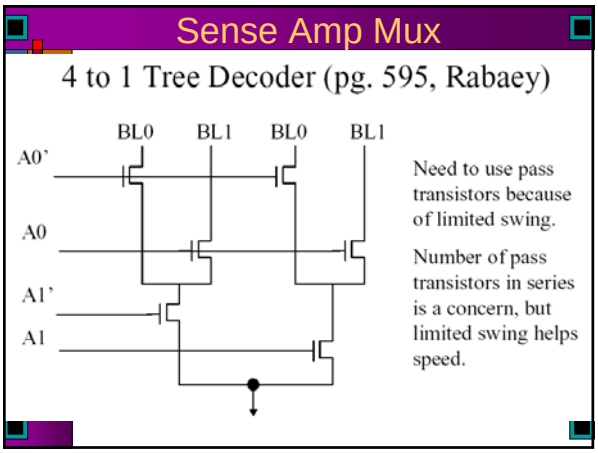
Array-Structured Memory



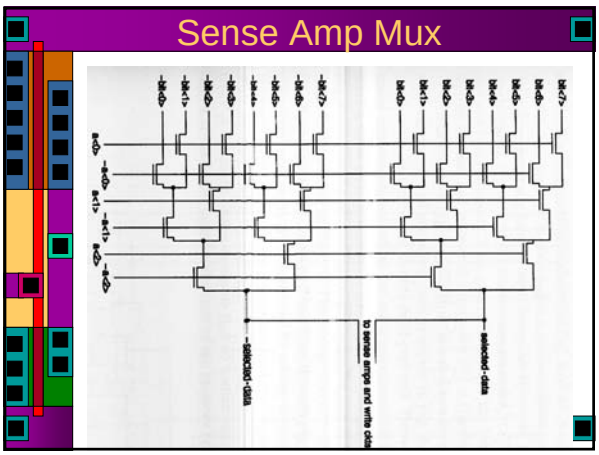
Sharing Sense Amps



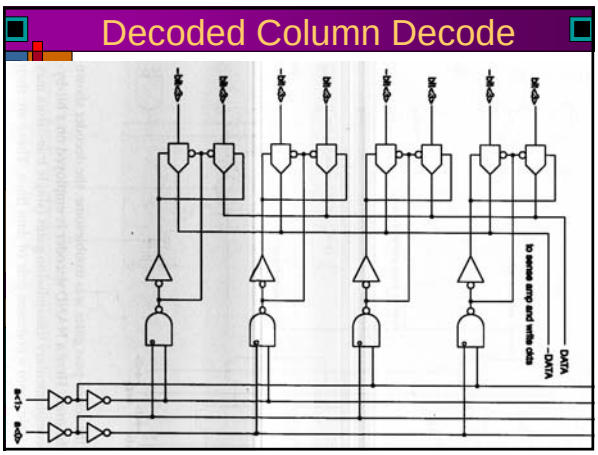
Sense Amp Mux



Sense Amp Mux



Decoded Column Decode

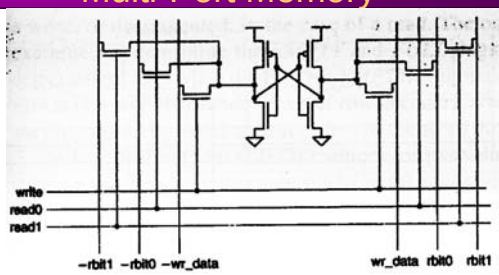


Improving Speed, Power

Critical path runs through row decode, word line assertion

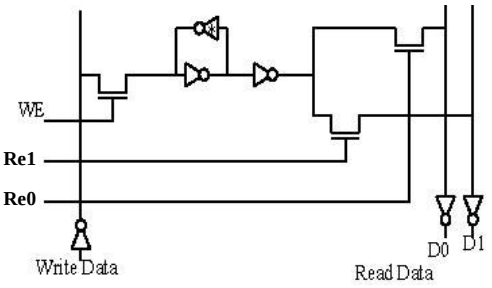
- Need smaller decoding, less word line capacitance in order to improve speed.
- Break a large array into smaller sub-arrays, and use hierarchical decoding to select a sub array
  - PowerPC 32K x 8 cache broken into 32 blocks, each 1K x 8
  - Cypress 1Mb Dual Port broken into 32 blocks, each 32 K bits ($2^5 \times 2^5 \times 2^{10} = 2^{20}$). Each blocks is 512 rows x 64 columns
  - Mitsubishi SRAM (Rabaey text). 32 blocks of 128K bits (1024 rows x 128 columns)
- Only one sub-array will be activated, saves power!!!!

Multi-Port Memory



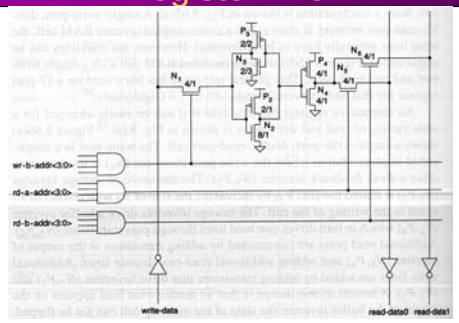
- Very common to require multiple read ports
  - Think about a register file, for example

Multi-Port Register



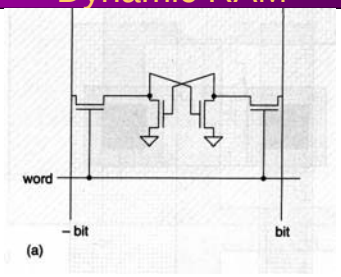
- Slightly larger cell, but with single-ended read – makes a great register file

Register File



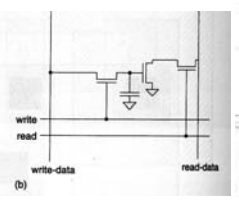
- Slightly larger cell, but with single-ended read – makes a great register file

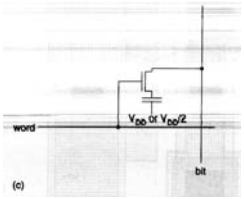
Dynamic RAM



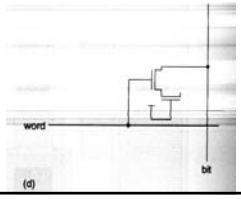
- Get rid of the pull-ups!
  - Store info on capacitors
  - Means that stored information leaks away

Dynamic RAM…

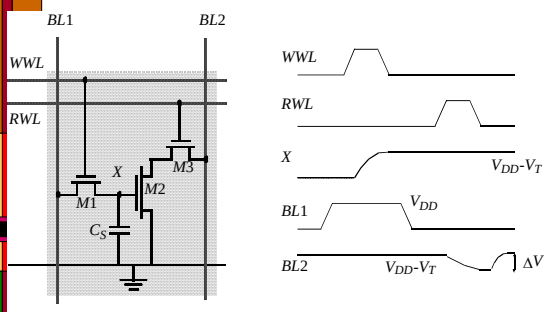




- Once you agree to use a capacitor for charge storage there are other ways to build this…

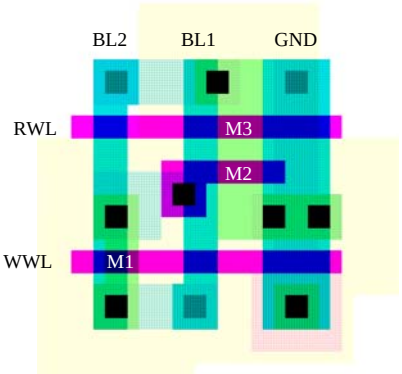


3T DRAM Circuit

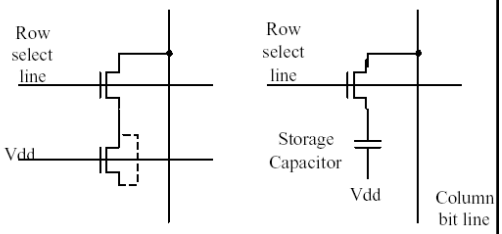


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a "1" = V_{WWL}-V_{Tn}

3T DRAM Layout



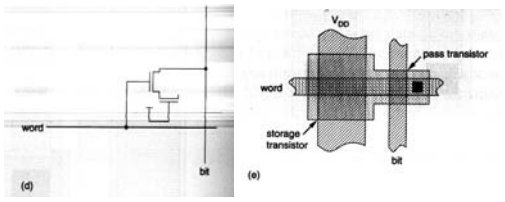
1 T DRAM Circuit



2 Transistor DRAM Cell Equivalent Circuit

$$C_{storage} = C_{drain} + C_{gate} + C_{source}$$

2-T (1-T) DRAM layout



- Note the increased gate size of the storage transistor
  - Increases the capacitance

1T DRAM Observations

1T DRAM requires a sense amplifier for each bit line, due to charge redistribution read-out.

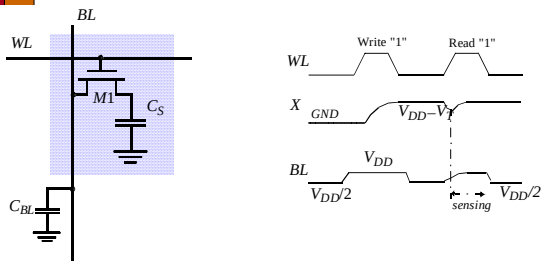
DRAM memory cells are single ended in contrast to SRAM cells.

The read-out of the 1T DRAM cell is destructive; read and refresh operations are necessary for correct operation.

Unlike 3T cell, 1T cell requires presence of an extra capacitance that must be explicitly included in the design.

When writing a "1" into a DRAM cell, a threshold voltage is lost. This charge loss can be circumvented by bootstrapping the word lines to a higher value than V_{DD}.

1T DRAM Read/Write

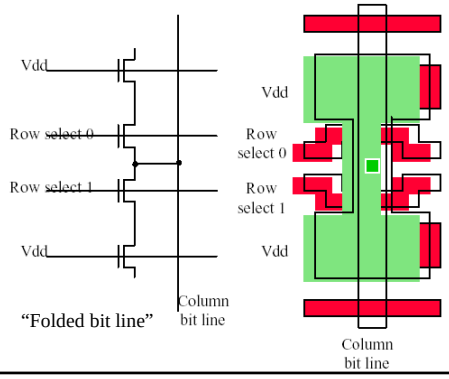


Write: C_S is charged or discharged by asserting WL and BL.
Read: Charge redistribution takes places between bit line and storage capacitance

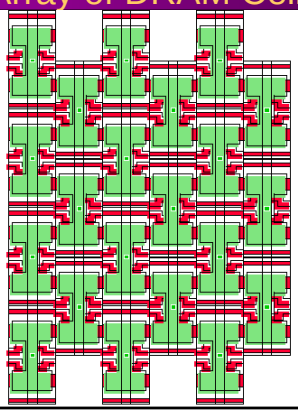
$$\Delta V = V_{BL} - V_{PRE} = (V_{BIT} - V_{PRE})\frac{C_S}{C_S + C_{BL}}$$

Voltage swing is small; typically around 250 mV.

1T DRAM Cell

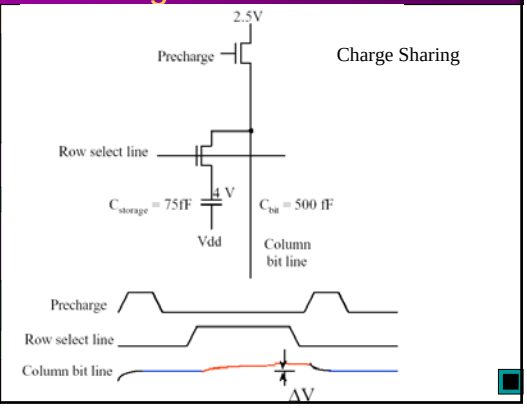


Array of DRAM Cells

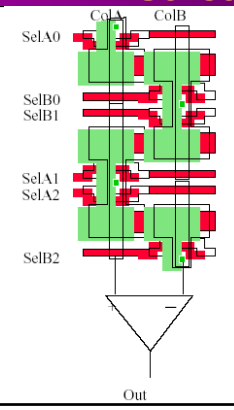


"Folded Bit Line"

Reading a 1T DRAM Cell



DRAM Sense Amp

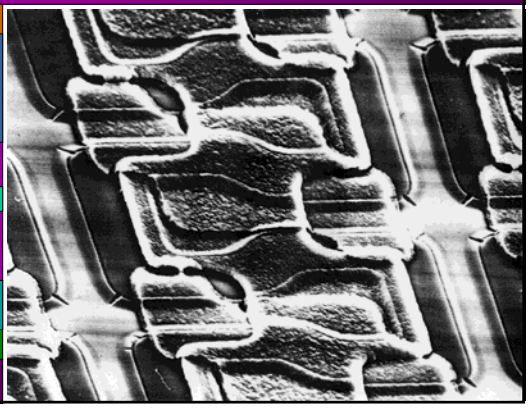


Column bit lines are precharged to Vdd/2.

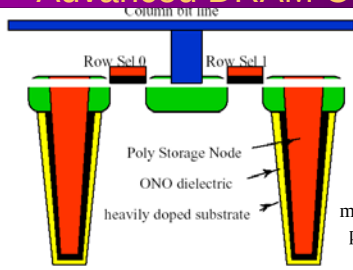
Near proximity of column lines gives excellent common noise rejection from coupled signals.

Sense amp must discriminate less than 100 mV voltage difference in the column lines

Photo of 1T DRAM



Advanced DRAM Cells



Try to get more capacitance per unit area…

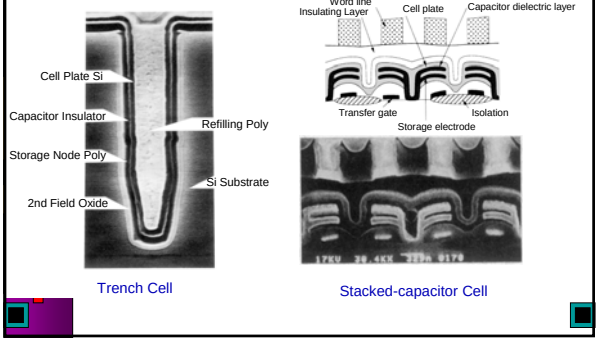
Trench Capacitor

ONO = Oxide Nitrate Oxide

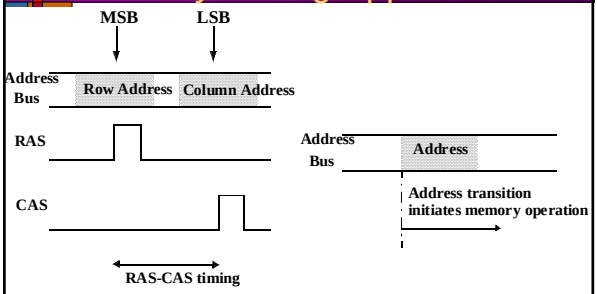
Special IC process for making DRAM

Smaller than 2 transistor cell . . . higher density memory

Examples of Advanced DRAMs



Memory Timing Approaches

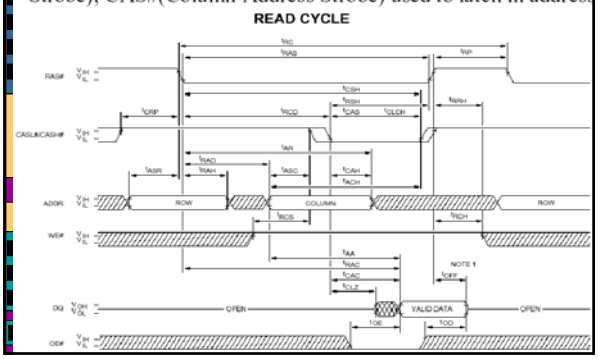


DRAM Timing
Multiplexed Adressing

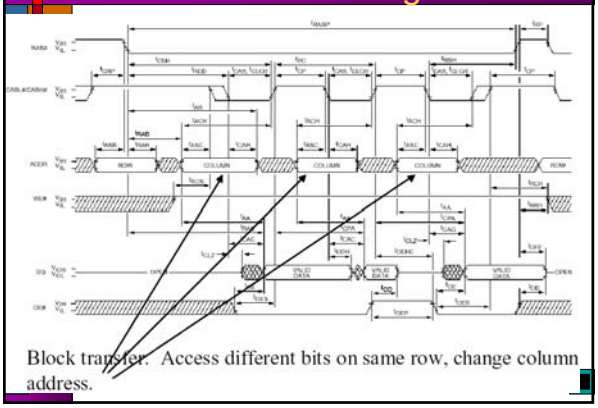
SRAM Timing
Self-timed

DRAM Interface

Multiplexed Address bus (Row, Column). RAS# (Row Address Strobe), CAS#(Column Address Strobe) used to latch in address



Extended Data Out Page Mode



Block transfer. Access different bits on same row, change column address.

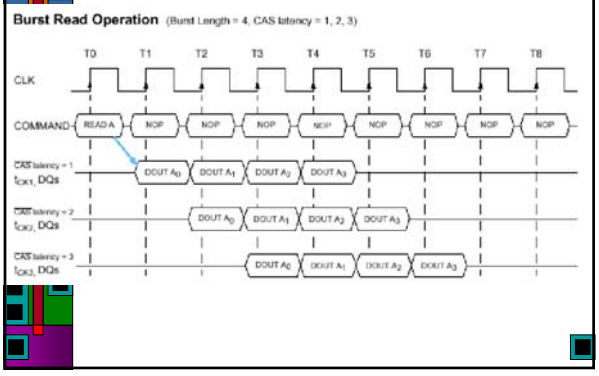
Comments on Timing

- Typical times are Tras = 60 ns (RAS pulse width), Trc = 100 ns
  - Extra time on Read cycle (RAS high) is needed to recharge bitlines
- Block mode transfers (Page mode transfers) read bits from same row
  - Only change column address
  - Time to first bit on row = 50ns, time to successive bits = 25 ns (we have access to all bits on this row, just need to mux them out).

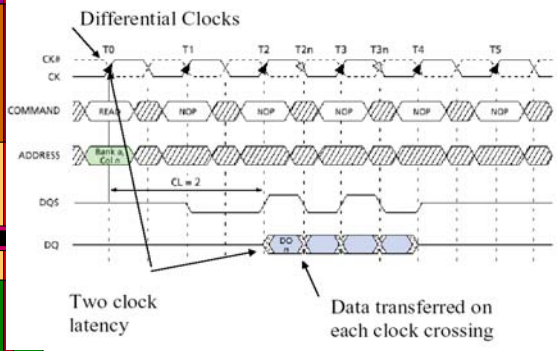
Architectural Issues

- Need to support block transfers efficiently since DRAM used as main memory and reads/writes due to cache fills
- Add a clock to DRAM interface (SDRAM, DDR-SDRAM) to support burst mode operations for cache fills
  - Pentium burst mode is 2-1-1-1 (two clocks for first data, 1 clock for each sucessive data, address only provided for first data, internal counter on RAM used for address generation).
  - Pentium Pipelined burst mode is: 2-1-1-1; 1*-1-1-1; 1*-1-1-1; .... Sucessive cycles pick up where the last cycle left off.

SDRAM - Use CAS for Bursts



DDR SDRAM



- Double Data Rate

DRAM Timing

- Clock Frequency – 133 Mhz, 100 Mhz
- Two clock latency to first data (20 ns for 100 Mhz clock)
  - SDRAM - 10 ns per location afterwards. For byte-wide, 100 MB/sec transfer rate. 400 MB/sec on 32-bit bus
  - DDR-SDRAM - 5 ns per location afterwards. For byte-wide, 200 MB/sec transfer rate. On 32-bit bus, 800 MB/sec transfer rate.

RAMBUS DRAM (RDRAM)

- DRAM with a high speed interface
- 400 Mhz differential clock, data transferred on each edge
- Reduced swing signaling about a reference voltage
  - Termination voltage is 1.5 V
  - Reference Voltage is 1.0 V
  - Signals swing +/- 200 mv about reference voltage
  - All traces are transmission lines

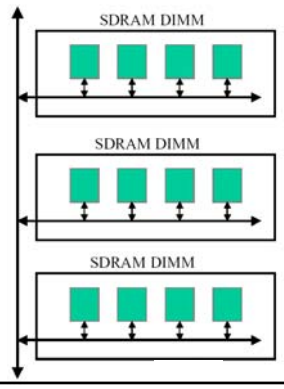
RDRAM Bandwidth

- External bus is 18 bits wide (2 bytes + 2 parity bits)
- External clock cycle is 400 Mhz, but data is clocked on each edge
  - Actually, external clock is a differential pair and data is sampled at each crossing
- Total Bandwidth is 1.6 GBytes/s
  - 2 bytes * 400 Mhz * 2 edges => 1.6 Gbytes
  - Initial configurations are 4 M x 18 (72 Mbits)

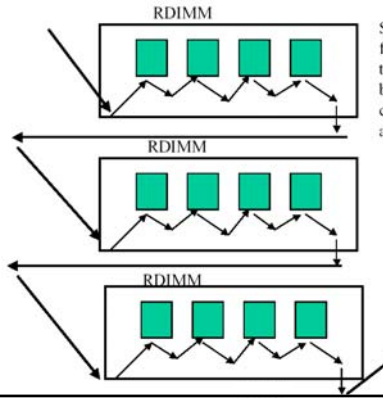
Maximum Bandwidth

- Note that maximum bandwidth with one RDRAM controller is 1.6GB/s.
  - Only one RDRAM chip can be active at a time on RDRAM bus.
  - More RDRAM chips increase capacity, not bandwidth.
    - With normal DRAM and SDRAM, can increase bandwidth by just adding more DRAM chips in parallel from same DRAM controller
  - To double the bandwidth, would need two separate RDRAM controllers

Normal Bus for DRAM DIMMs



RDRAM Bus



Signaling Technology for RDRAM basically the same as PentiumII bus. RDIMMs must be connected serially to avoid stubs.

Deep Pipelining - High Latency

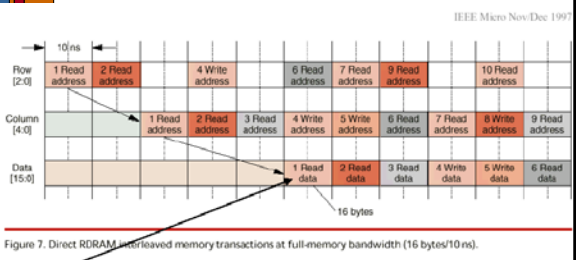


Figure 7. Direct RDRAM interleaved memory transactions at full-memory bandwidth (16 bytes/10 ns).

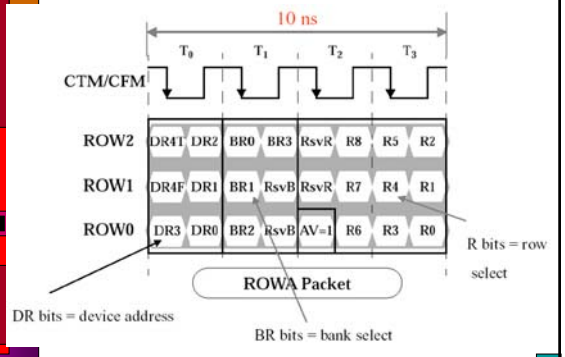
16 bytes transferred because 4 clocks * 2 edges * 2 bytes/transfer

(external bus is 16 or 18 bits wide). 20 clock latency, 20 ns from column address)

RDRAM Addressing

- 3-Bit Row bus used to give commands to RDRAM
- ROW Activate command used for read
  - 4 clocks transfers 8 groups of 3 bits over Row bus due to dual edge clocking (24 bits total)
  - 24 bits in Row Activate command split between device address (6 bits), bank select (4 bits), row select (9 bits), and reserved bits
- There are no chip select lines, internal register holds device address
  - All chips monitor bus - if bus device address matches internal id, then chip is selected.

Row Activate Command



RDRAM System Arch

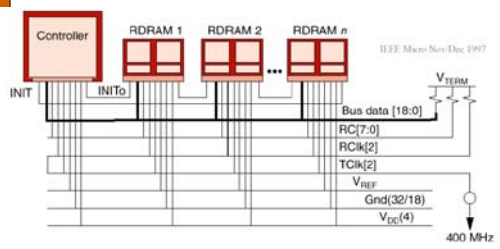
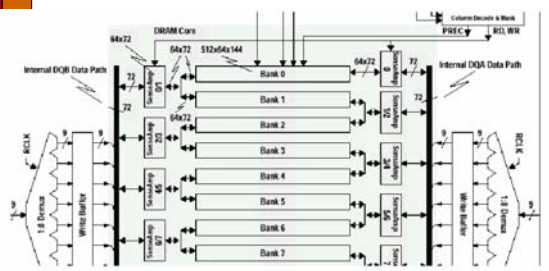


Figure 3. Direct RDRAM system.

18 bit wide external data bus which expands into 128 bit wide datapath internal to chip

RDRAM Internal Arch



Portion of internal architecture (4M x 16 or 4M x 18)

16 banks of 512 rows of 64 dualocts (1 dualoct = 16 bytes = 128 bits)

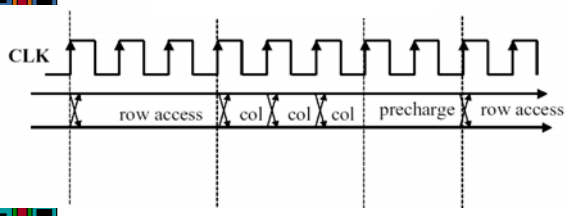
2^4 (banks) * 2^9 (rows) * 2^6 (dualocts) * 2^7 (one dualoct) = 2^{26} (64 Mbit)

A dualoct is the smallest addressable unit.

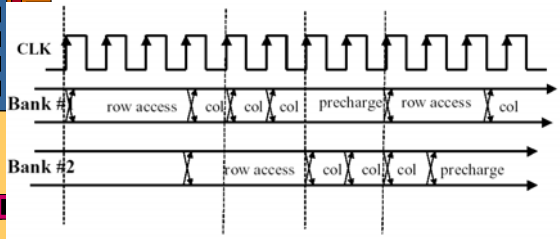
Regular DRAM

- Multiple Banks are key to high throughput
- As one DRAM bank is recovering from read operation, next bank is being accessed
- Essentially on-chip memory interleaving
- Goal is to hide latency and bitline precharge time (recovery time)
  - Latency is access to first byte, critical path through row-decode and word line assertion
  - Bitline Precharge time (recovery time to next access) depends on number of bits in a column (number of rows)

Single Bank DRAM



Multi-Bank DRAM



Number of banks required to hire all row latency and precharge time depends on ratio of latency+precharge to column access time.

Peak Bandwidth

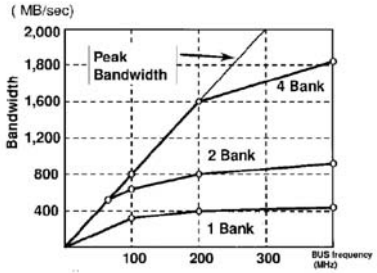
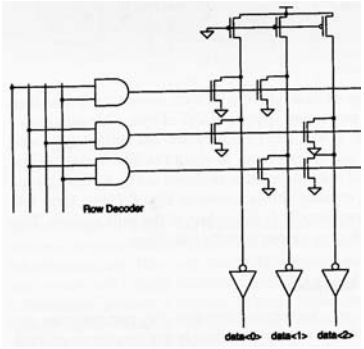


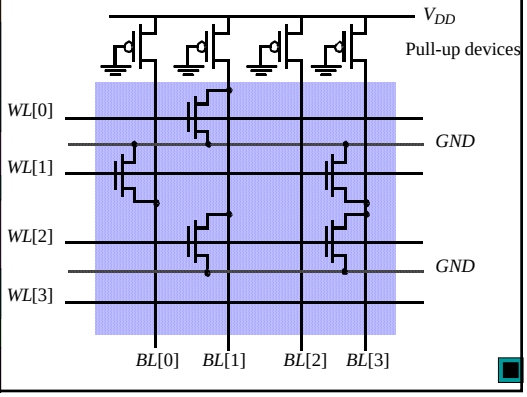
Fig. 11. Multibank system bandwidth calculations.

"High-Speed Dram Architecture Development", H. Ikeda and H. Inukai, IJSSC VOl 34, No 5, May 1999.

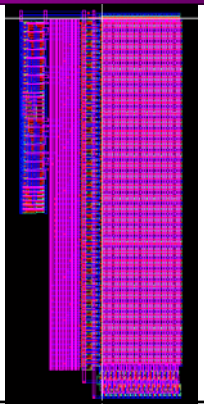
ROM



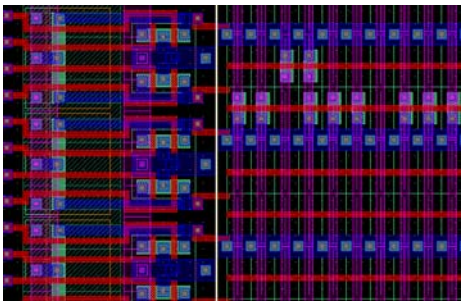
ROM



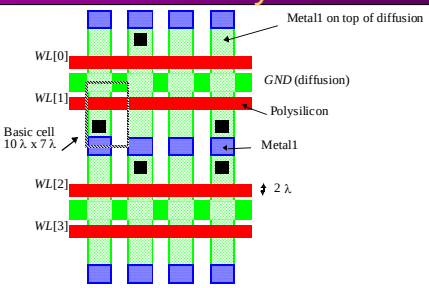
ROM



ROM

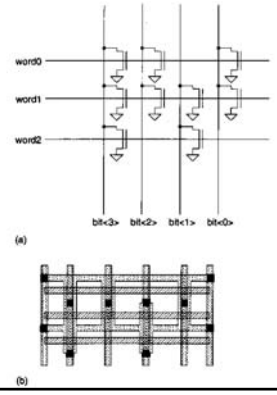


ROM Layout

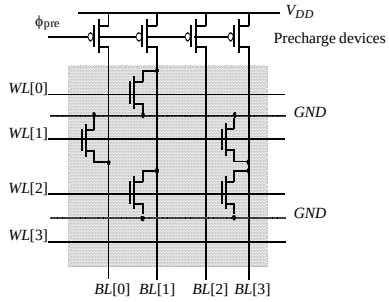


Only 1 layer (contact mask) is used to program memory array
Programming of the memory can be delayed to one of last process steps

ROM Layout

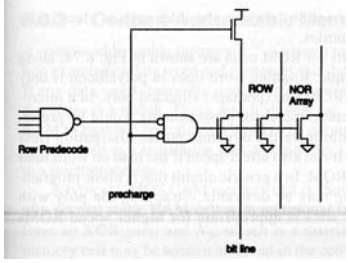


Precharged ROM

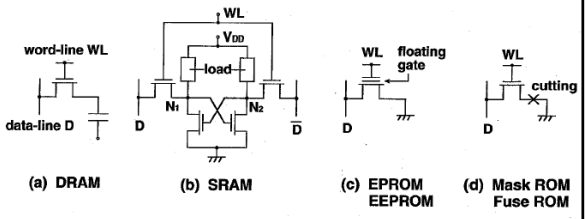


PMOS precharge device can be made as large as necessary, but clock driver becomes harder to design.

Precharged ROM



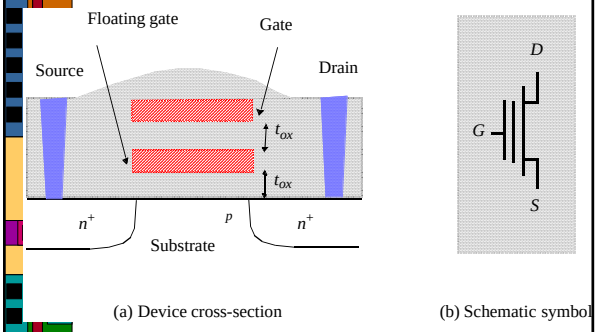
Other Memory Cells



Non-Volatile ROM

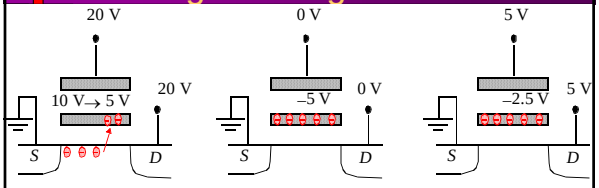
- EPROM
  - Erasable Programmable ROM
- EEPROM
  - Electrically Erasable Programmable ROM
- Flash EEPROM
  - Electrically Erasable Programmable ROM that is erased in large chunks
- All these devices rely on trapping charge on a floating gate

EPROM



(a) Device cross-section (b) Schematic symbol

Programming EPROM



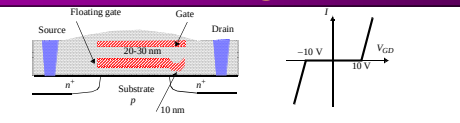
Avalanche injection. **Removing programming voltage leaves charge trapped.** **Programming results in higher V_T.**

- Higher Vth (around 7v) means that 5v Vgs no longer turns on the transistor
- SiO2 is an excellent insulator
  - Trapped charge can stay for years

Erasing an EPROM

- Erase by shining UV light through window in the package
  - UV radiation makes oxide slightly conductive
  - Erasure is slow - from seconds to minutes depending on UV intensity
  - Also the erase/program cycles are limited (around 1000), mainly as a result of the UV erasing
- But, EPROMs are simple and dense

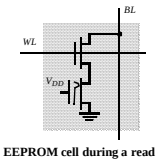
EEPROM



(a) Flotox transistor

(b) Fowler-Nordheim *I-V* characteristic

Floating Gate Tunneling Oxide transistor



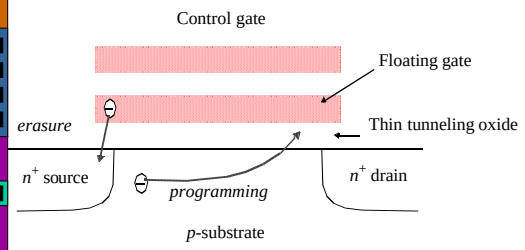
(c) EEPROM cell during a read operation

- Thin oxide allows erasing in-system
  - Fowler-Nordheim Tunneling

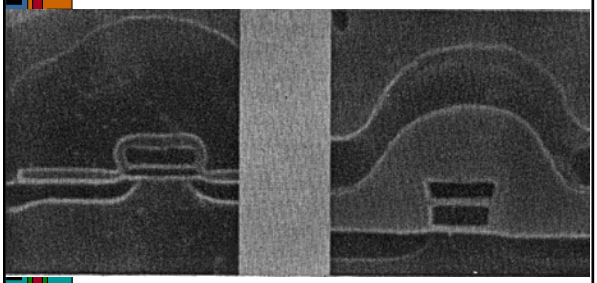
EEPROM

- Two transistors instead of one
  - The second keeps you from removing too much charge during erasure
- Bigger and not as dense as EPROM
- But, more erase/program cycles
  - On the order of 10^5
  - Eventually you get permanently trapped charge in the SiO2

Flash EEPROM



- Essentially the same as EEPROM
  - But, large regions erased at once
  - Means you can monitor the voltages and don't need the extra access transistor

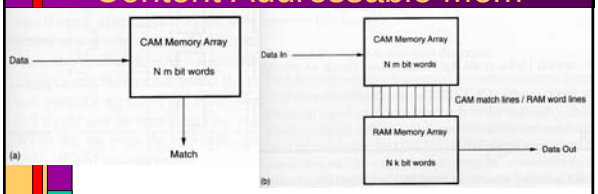


Flash EEPROM

Realistic PROM Devices

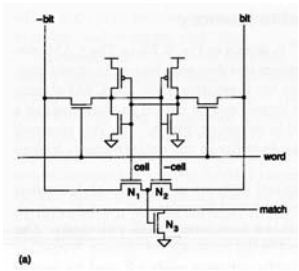
	EPROM [Tomita91]	EEPROM [Terada89, Pashley89]	Flash EEPROM [Jinbo92]
Memory size	16 Mbit (0.6 μm)	1 Mbit (0.8 μm)	16 Mbit (0.6 μm)
Chip size	7.18 x 17.39 mm^2	11.8 x 7.7 mm^2	6.3 x 18.5 mm^2
Cell size	3.8 μm^2	30 μm^2	3.4 μm^2
Access time	62 nsec	120 nsec	58 nsec
Erasure time	minutes	N.A.	4 sec
Programming time/word	5 μsec	8 msec/word, 4 sec /chip	5 μsec
Erase/Write cycles [Pashley89]	100	10^5	10^3-10^5

Content Addressable Mem



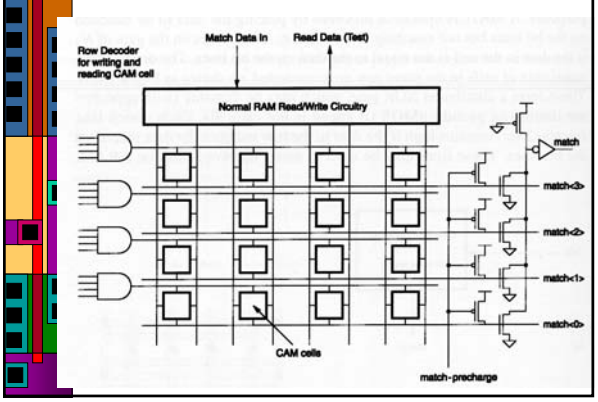
- Asks the question: Are there are any locations that hold this value?
  - Used for tag memories in associative caches
  - Or translation lookaside buffers
  - Or other pattern matching applications

Content Addressable Mem

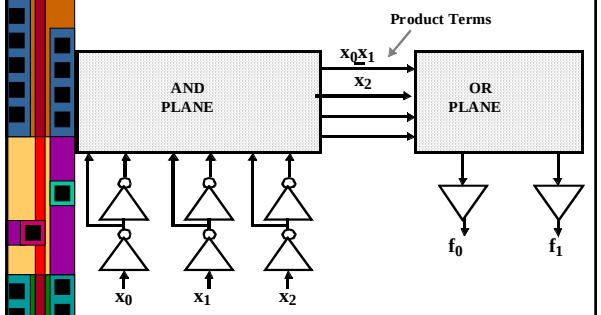


- Add the Match line
  - Essentially a distributed NOR gate

Content Addressable Mem



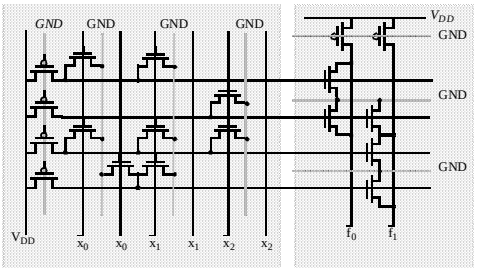
Programmable Logic Array



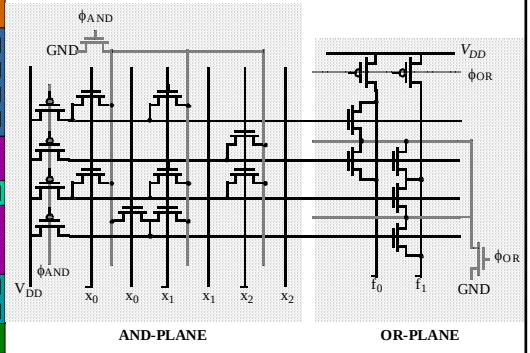
PLA

- Still useful for random combinational logic
  - Standard cell ASIC tools may be replacing them
- They can generate dense AND-OR circuits

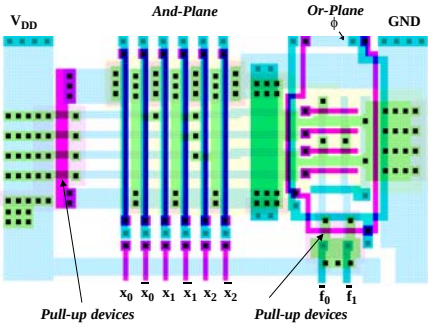
Pseudo-Static PLA Circuit



Dynamic PLA



PLA Layout



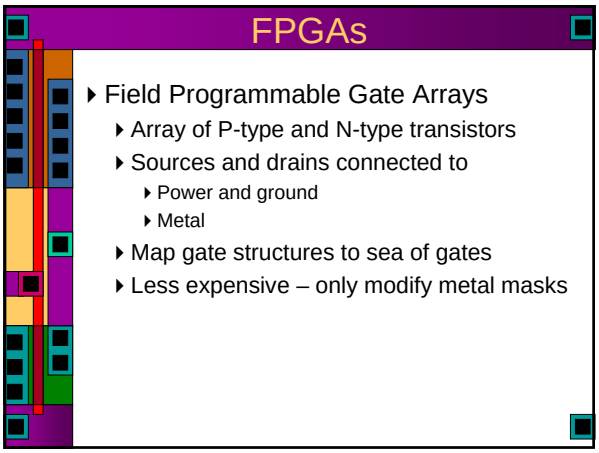
PLA vs. ROM

Programmable Logic Array
structured approach to random logic
"two level logic implementation"
NOR-NOR (product of sums)
NAND-NAND (sum of products)

IDENTICAL TO ROM!

Main difference
ROM: fully populated
PLA: one element per minterm

Note: Importance of PLA's has drastically reduced
1. **slow**
2. **better software techniques (mutli-level logic synthesis)**


FPGAs
Field Programmable Gate Arrays
Array of P-type and N-type transistors
Sources and drains connected to
Power and ground
Metal
Map gate structures to sea of gates
Less expensive – only modify metal masks