Topic 10
Memory Circuits

Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

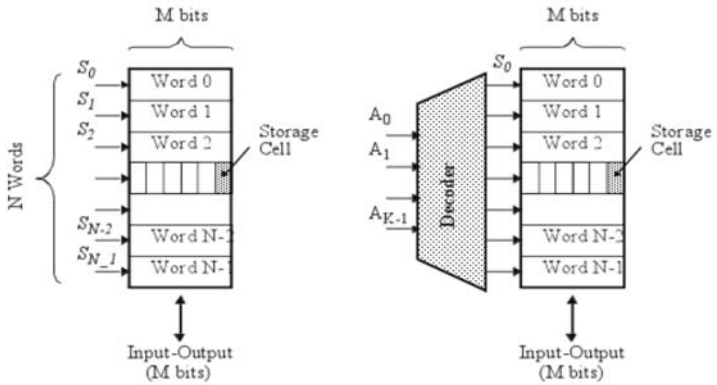
Reading: Weste Ch 8.3.1-8.3.2, Rabaey Ch.10, p.551-595

URL: www.ee.ic.ac.uk/pcheung/
E-mail: p.cheung@ic.ac.uk

Semiconductor Memory Classification

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

Memory Architecture: Decoders

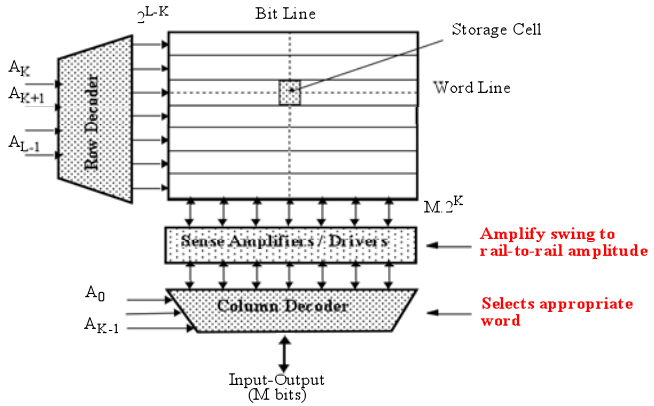


N words => N select signals
Too many select signals

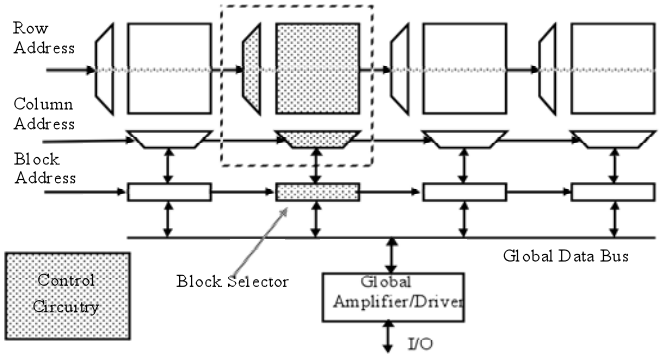
Decoder reduces # of select signals
$K = log_2 N$

Array-Structured Memory Architecture

Problem: ASPECT RATIO or HEIGHT >> WIDTH

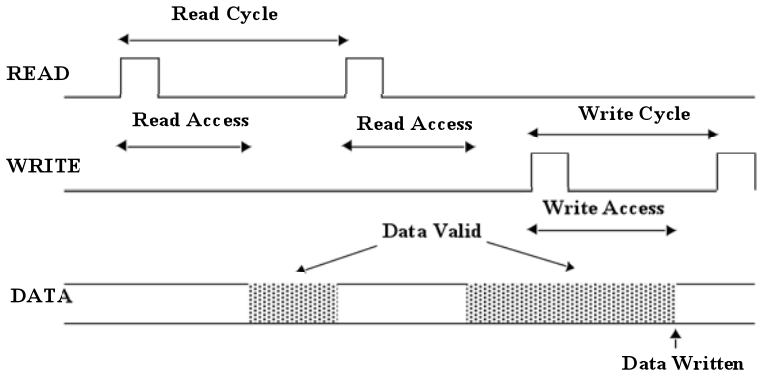


Hierarchical Memory Architecture

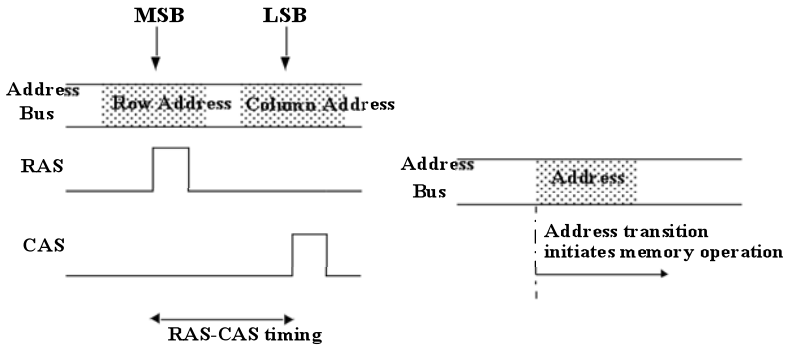


Advantages:
1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Memory Timing: Definitions



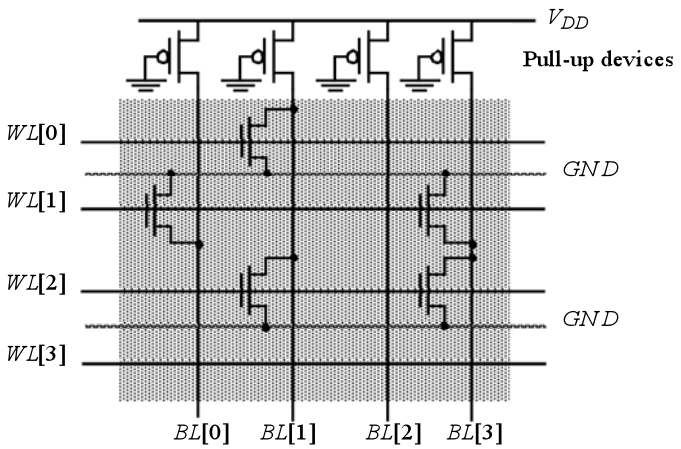
Memory Timing: Approaches



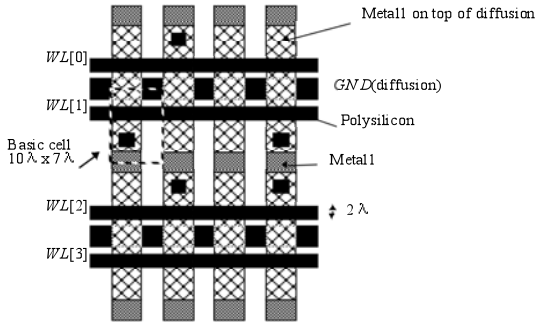
DRAM Timing
Multiplexed Adressing

SRAM Timing
Self-timed

MOS NOR ROM

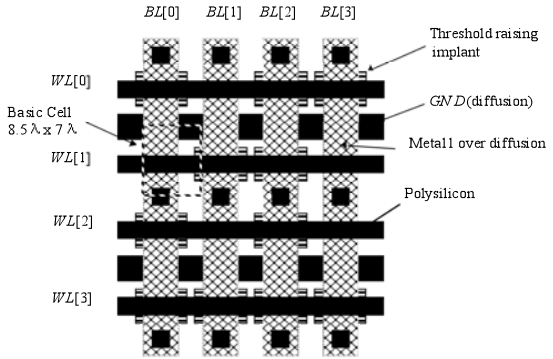


MOS NOR ROM Layout



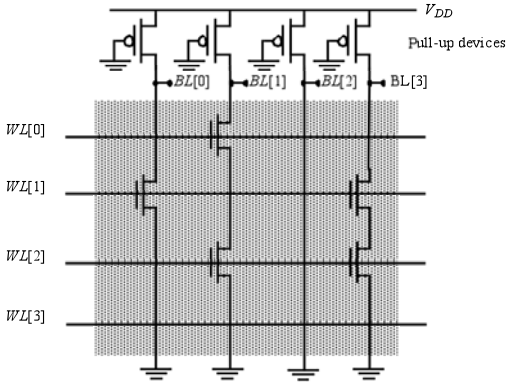
Only 1 layer (contact mask) is used to program memory array
Programming of the memory can be delayed to one of last process steps

MOS NOR ROM Layout



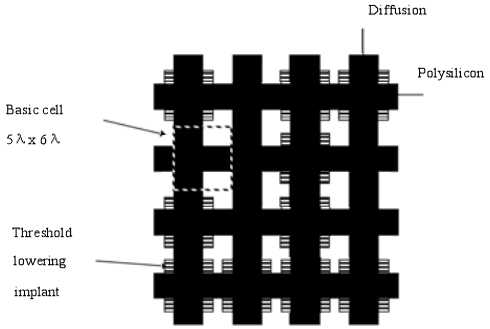
Threshold raising implants disable transistors

MOS NAND ROM



All word lines high by default with exception of selected row

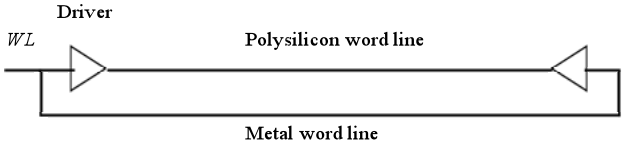
MOS NAND ROM Layout



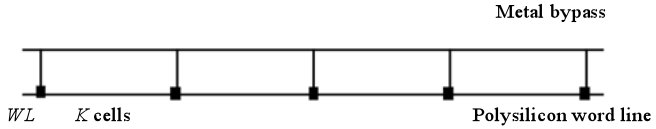
No contact to VDD or GND necessary; drastically reduced cell size

Loss in performance compared to NOR ROM

Decreasing Word Line Delay



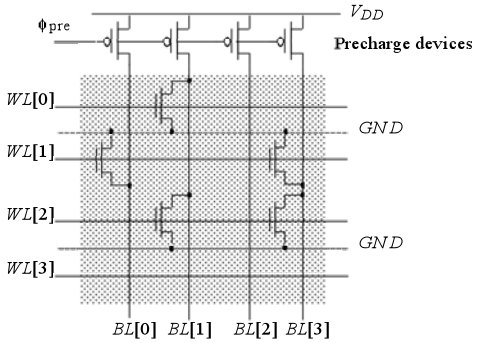
(a) Driving the word line from both sides



(b) Using a metal bypass

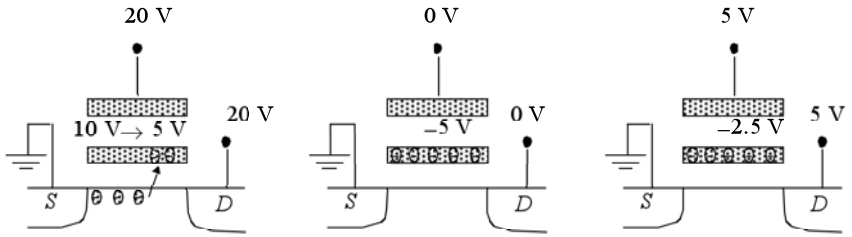
(c) Use silicides

Precharged MOS NOR ROM



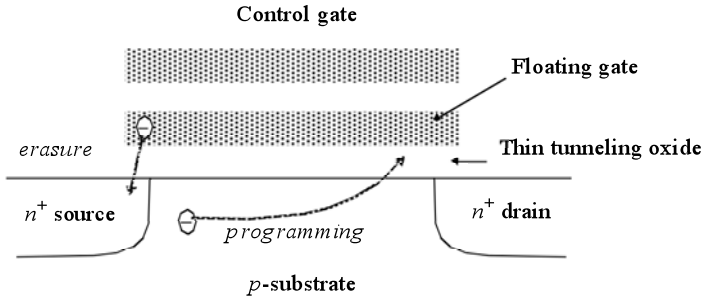
PMOS precharge device can be made as large as necessary, but clock driver becomes harder to design.

Floating-Gate Transistor Programming

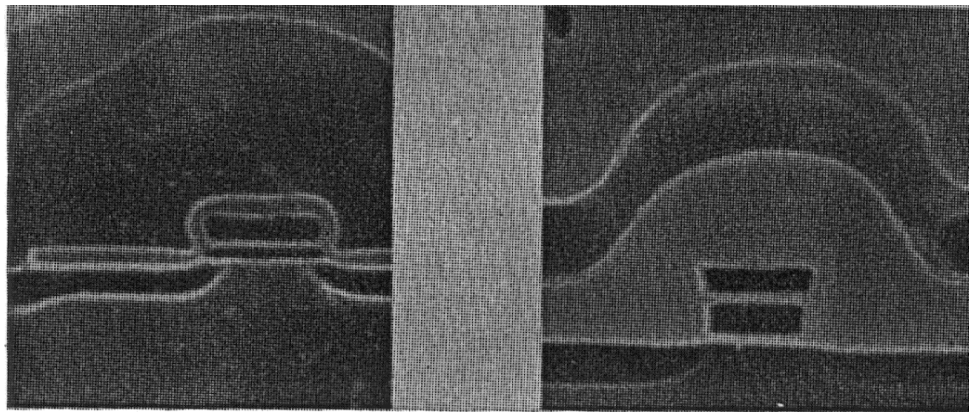


Avalanche injection. **Removing programming voltage leaves charge trapped.** **Programming results in higher V_T.**

Flash EEPROM



Cross-sections of NVM cells



Flash EPROM

Courtesy Intel

Characteristics of State-of-the-art NVM

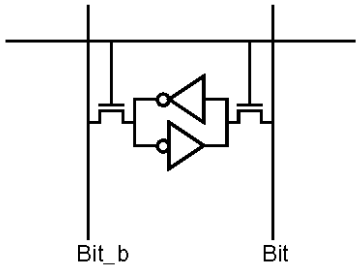
	EPROM [Tomita91]	EEPROM [Terada89, Pashley89]	Flash EEPROM [Jinbo92]
Memory size	16 Mbit (0.6 μm)	1 Mbit (0.8 μm)	16 Mbit (0.6 μm)
Chip size	7.18 x 17.39 mm^2	11.8 x 7.7 mm^2	6.3 x 18.5 mm^2
Cell size	3.8 μm^2	30 μm^2	3.4 μm^2
Access time	62 nsec	120 nsec	58 nsec
Erasure time	minutes	N.A.	4 sec
Programming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

Read-Write Memories (RAM)

- **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**

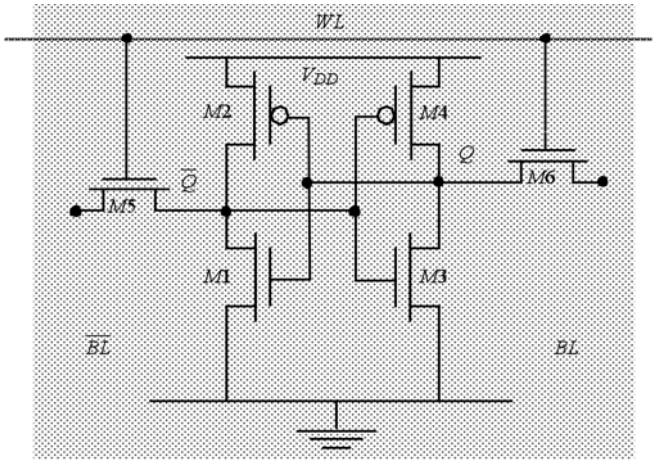
Basic RAM Cell

Uses only six transistors:



Read and write use the same port. There is one wordline and two bit lines. The bit lines carry the data. The cell is small since it has a small number of wires.

6-transistor CMOS SRAM Cell



SRAM Read/Write

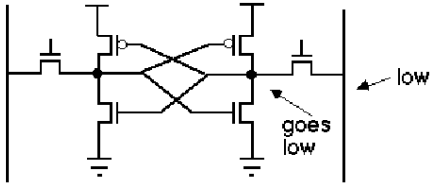
The key issue in an 6T SRAM is how to distinguish between read and writes. There is only one wordline, so it must be high for both reads and writes. The key is to use the fact there are two bitlines.

Read:

- Both Bit and $\overline{\text{Bit}}$ must start high. A high value on the bitline does not change the value in the cell, so the cell will pulls one of the lines low

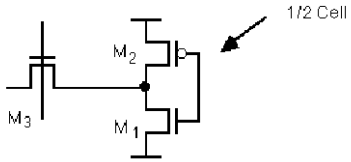
Write:

- One (Bit or $\overline{\text{Bit}}$) is forced low, the other is high
- This low value overpowers the pMOS in the inverter, and this will write the cell.



RAM Cell Design

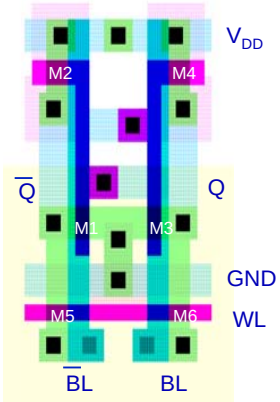
For the cell to work correctly a zero on the bit line must over power the pMOS pull up, but a one on the bit line must not over power the pull down (otherwise reads would not work)



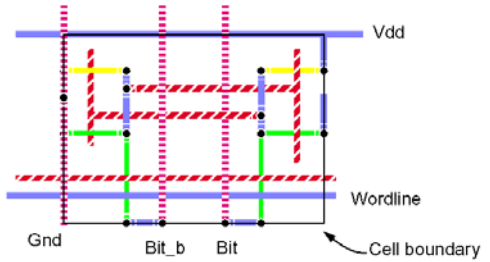
For the pull down M3 is passing a zero, so for it to overpower the pMOS it must be at least as wide (preferably 1.5x as wide). This gives a 2-3:1 current ratio between the nMOS and the pMOS.

For pull up M3 is passing a one so it is somewhat weaker. Still M3 should be 1.5 to 2x smaller than M1 to make sure a read does not disturb the value of the cell.

6T-SRAM — Layout



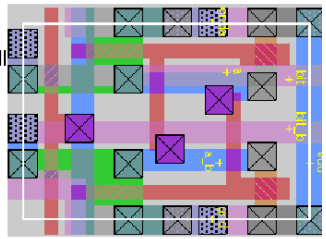
There are many clever SRAM layouts. This is a common one:

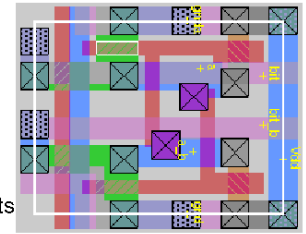


This layout is fairly dense, since the most of the contacts (bitline, Vdd, Gnd) are shared. Also the a clever cross-coupling method is used.

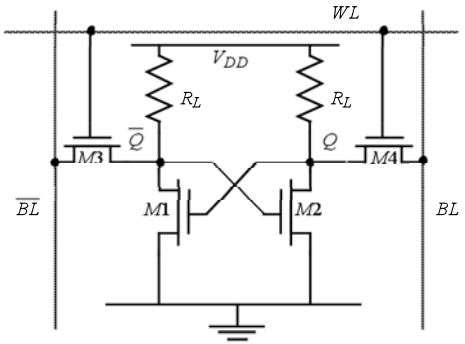
More Cell Layout

- A conservative cell:
  - It has substrate and well connects in each cell
  - It has a wordline poly contact in each cell
  - pMOS transistors are very weak (3:3)
  - nMOS pulldown is 8:2
  - All the boundaries are shared
  - 41 x 28, about 1/4 the size of latch cell
- A slightly smaller cell
  - Only nwell contact in cell
  - pMOS transistors are very weak (3:3)
  - nMOS pulldown is 6:2
  - 36 x 28
- White box is the repeat box. Cells overlap contacts





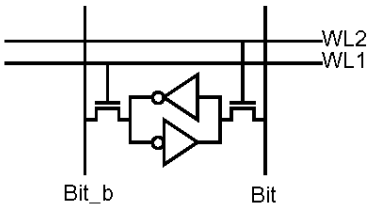
Resistance-load SRAM Cell

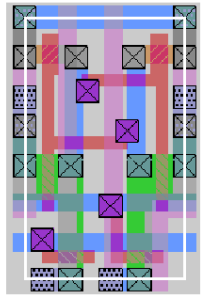


Static power dissipation -- Want R_L large
Bit lines precharged to V_{DD} to address t_p problem

Dual Port RAM

Split wordline so there are two wordlines, one for each pass transistor.

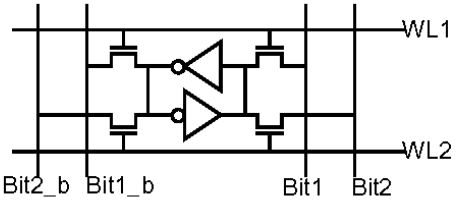




- Nearly the same size as SRAM, 46 x 28
- Can read two different cells in one cycle or perform one write.
  - Raise WL1 on Register 5, and WL2 on Register 7. Register 5 value will be on bit_b (complemented, of course), and Register 7 will be on bit. Since you need both bit and bit_b to write the cell, you can only do one write per cycle.

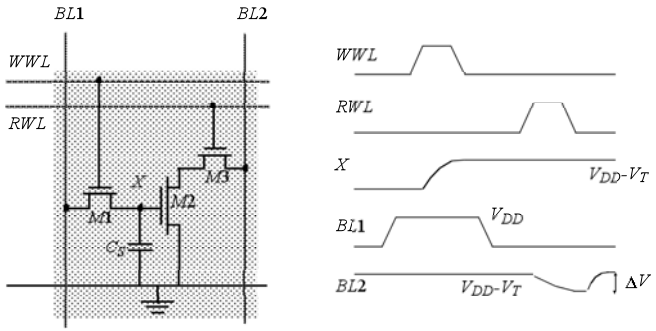
Multiport RAM Cell

Can build true multi-ported memory cell, by adding more bitline pairs and wordlines to a cell.



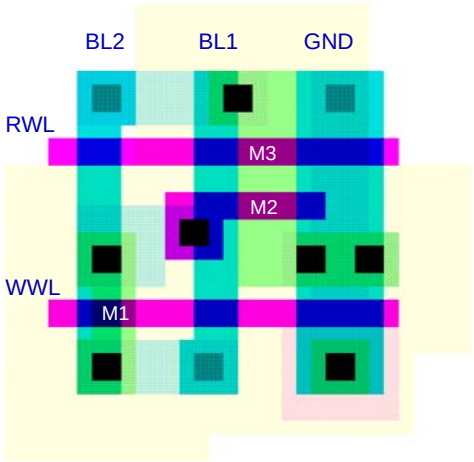
Shown in the figure is a true dual port cell. You can read or write on each port every cycle. Since it has more bitlines than the previous cell, it is much larger in area.

3-Transistor DRAM Cell

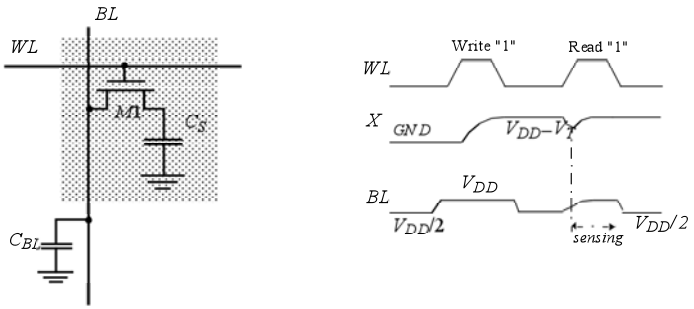


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a "1" = V_{WWL}-V_{Tn}

3T-DRAM — Layout



1-Transistor DRAM Cell



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.

DRAM Cell Observations

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

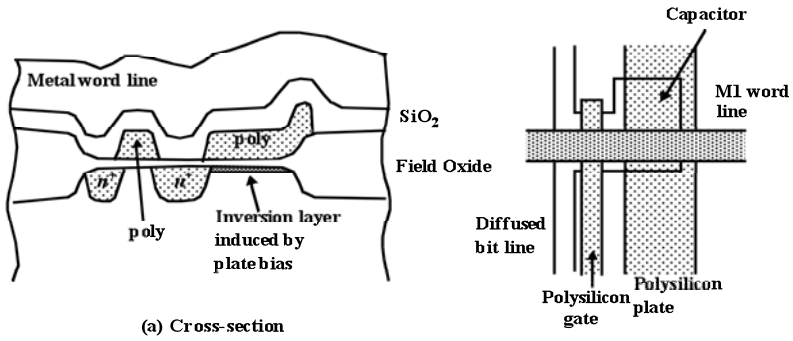
DRAM memory cells are single ended in contrast to SRAM cells.

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}.

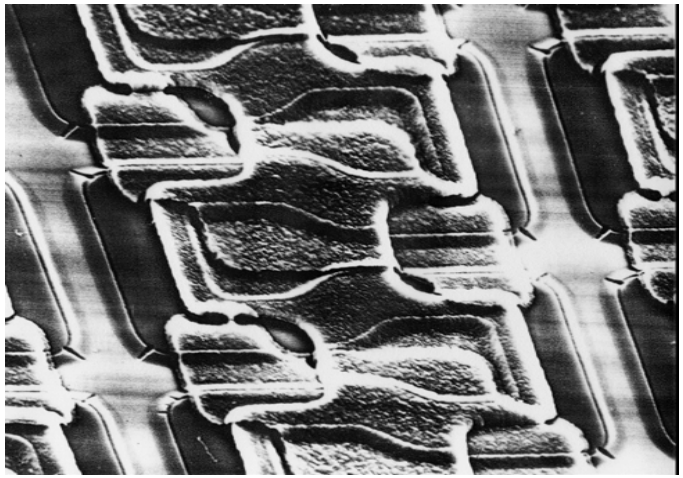
1-T DRAM Cell



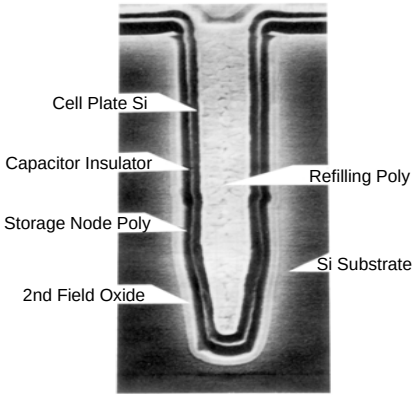
Used Polysilicon-Diffusion Capacitance

Expensive in Area

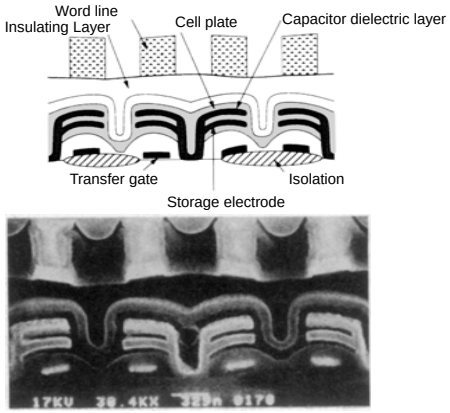
SEM of poly-diffusion capacitor 1T-DRAM



Advanced 1T DRAM Cells

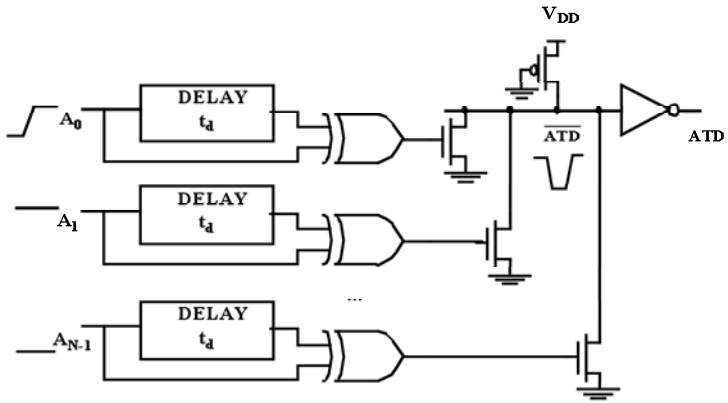


Trench Cell



Stacked-capacitor Cell

Address Transition Detection



Row Decoders

Collection of 2^M complex logic gates
Organized in regular and dense fashion

(N)AND Decoder

$$WL_0 = A_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

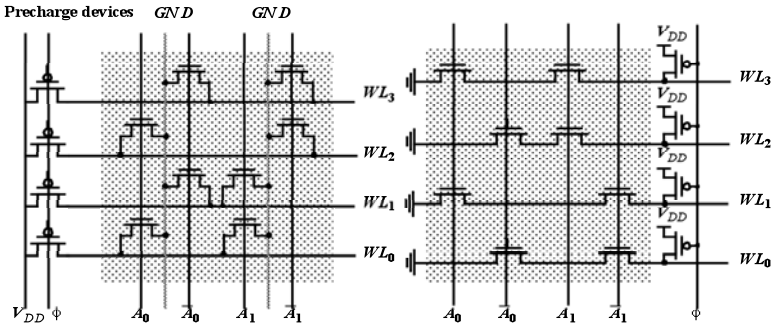
$$WL_{511} = \bar{A}_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

NOR Decoder

$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

$$WL_{511} = \overline{A_0 + \bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \bar{A}_4 + \bar{A}_5 + \bar{A}_6 + \bar{A}_7 + \bar{A}_8 + \bar{A}_9}$$

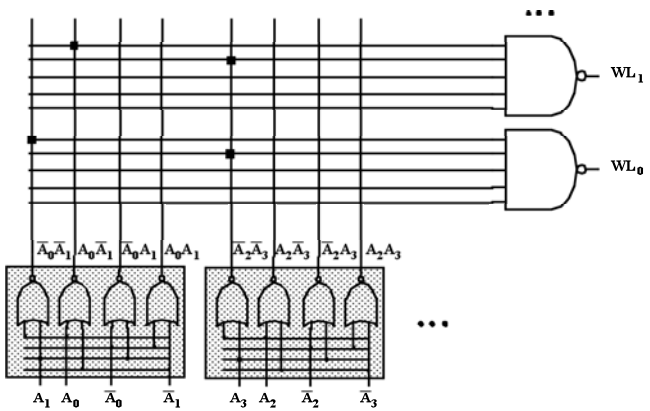
Dynamic Decoders



Dynamic 2-to-4 NOR decoder 2-to-4 MOS dynamic NAND Decoder

Propagation delay is primary concern

A NAND decoder using 2-input pre-decoders

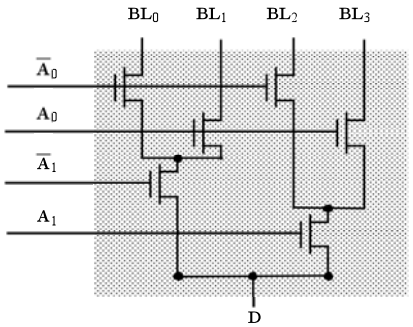


Splitting decoder into two or more logic layers produces a faster and cheaper implementation

4 input pass-transistor based column decoder

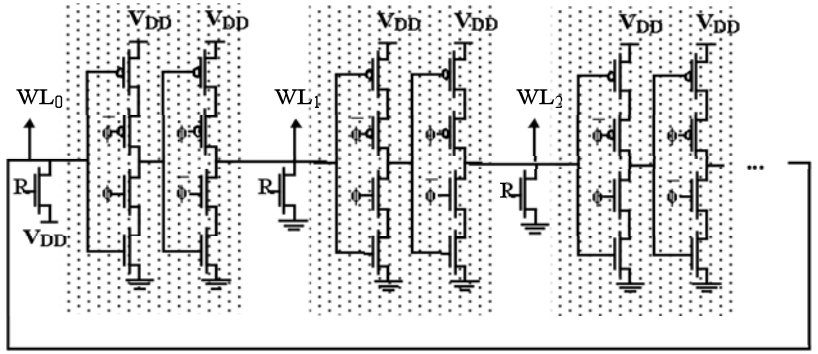
Advantage: speed (t_{pd} does not add to overall memory access time)
only 1 extra transistor in signal path
Disadvantage: large transistor count

4-to-1 tree based column decoder



Number of devices drastically reduced
Delay increases quadratically with # of sections; prohibitive for large decoders
Solutions: **buffers**
progressive sizing
combination of tree and pass transistor approaches

Decoder for circular shift-register



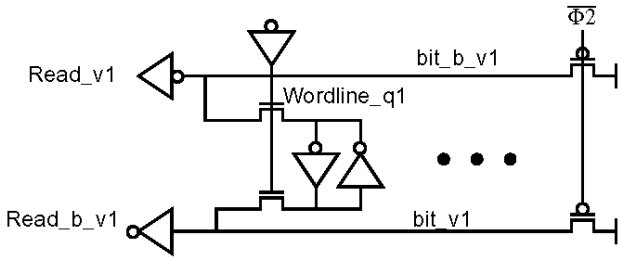
Bitline I/O Circuit - Read

For reads both bitlines must be high, for write you need to drive the bitlines to the correct value.

- Bitlines need to be precharged, or use a pseudo nMOS load

We will use a precharged structure:

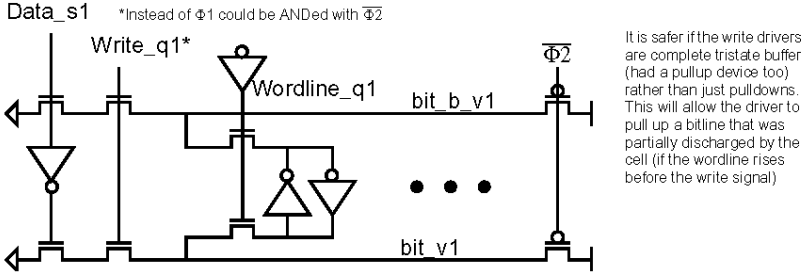
- To avoid a conflict during precharge, make wordline a qualified clock.



- Bitlines are like the outputs of normal precharge gates - _v signals

Bitline I/O Circuit - Write

Similar to read, but need to drive the bit lines too.



It is safer if the write drivers are complete tristate buffer (had a pullup device too) rather than just pulldowns. This will allow the driver to pull up a bitline that was partially discharged by the cell (if the wordline rises before the write signal)

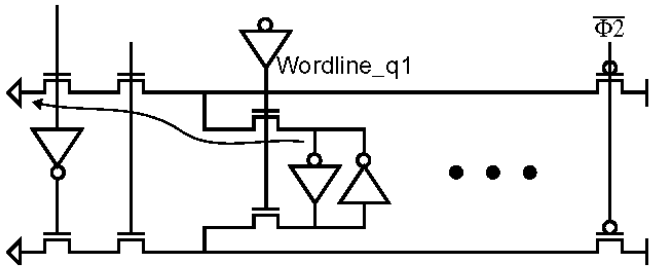
Notice that since the memory cell is a storage element, its enable (the wordline) needs to be a _q signal. That will ensure that the clock falls latching in the data BEFORE the data has a chance to change. The wordline is really the clock to the latch (memory) cell.

Need to isolate the write driver so it does not fight with the precharge (power issue), which is why the write signal is qualified.

Write Drivers

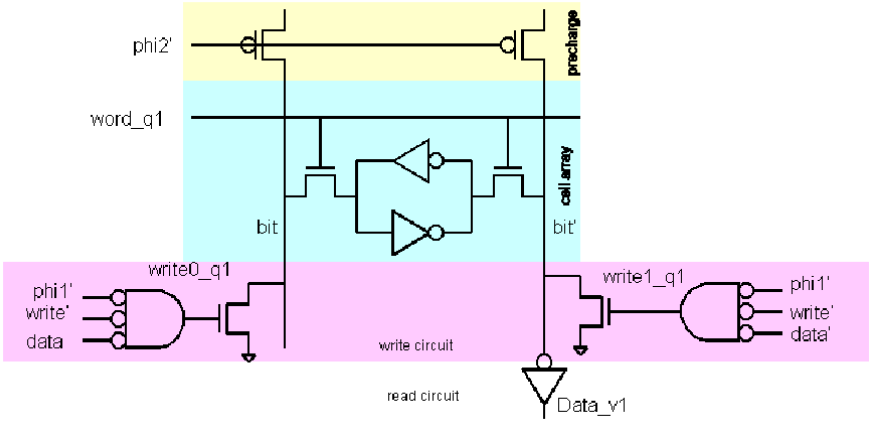
Also need to worry about the series resistance of the driver

- The resistance of the 3 series nMOS transistors must still be 2x less than the resistance of the pMOS in the cell



- If the pass device in the cell is 4:2, and the pMOS load is 3:4, then each nMOS in the driver must be at least 8:2. If the pMOS was 3:2, it would be hard to get the cell to write in irsim.

Alternative Write Circuit



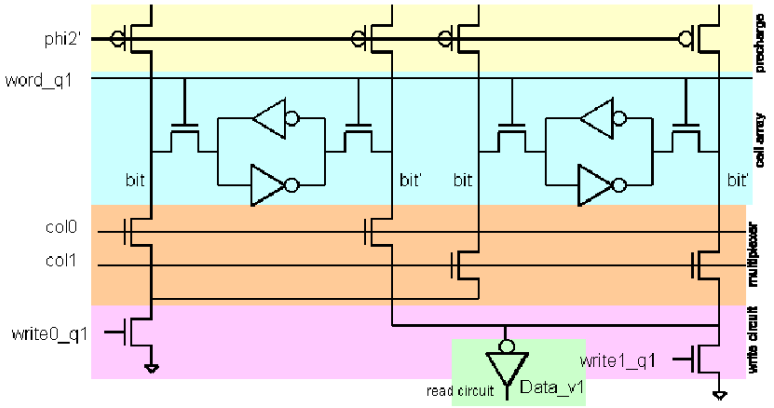
Bitline Multiplexing

The bitline pitch is pretty small (about 28λ) and there is a lot of stuff that is needed for the bitline (read and write circuits). Often many bitlines are muxed together, and one set of IO circuits is used for these bitlines

Two basic options:

- Share mux between read and write circuits
  - + Least amount of logic needed on bit pitch
  - - Adds another series device to write drivers, need to use single write device
- Use different mux for read and write
  - + No series devices in write path
    - Write mux can be qualified with Write & Clock
  - - Adds some more muxes to bit logic

Bitline Mux - Option 1



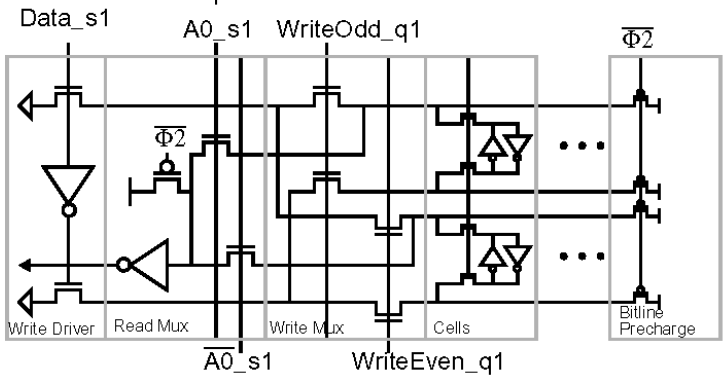
Should have a precharge on the output of the mux too, since otherwise the output will have a degraded high level.

Bitline Mux - Option 2

Uses separate mux for read and write

Notice that the read mux is precharged

- You don't need to use pMOS devices in the mux



Sense Amplifiers

- Since Cdiff ~ Cgate, the diffusion contacts of access devices cause large cap on the bitlines, for large arrays
- Bitline cap becomes an issues around 32 cells/bitline

  Example:

  If the diffusion contacts are shared (adjacent cells), 128 cells @4fF/2cells = 256fF + wire cap. This would lead to access times of around 3ns.

Can take advantage of the differential nature of the bitlines

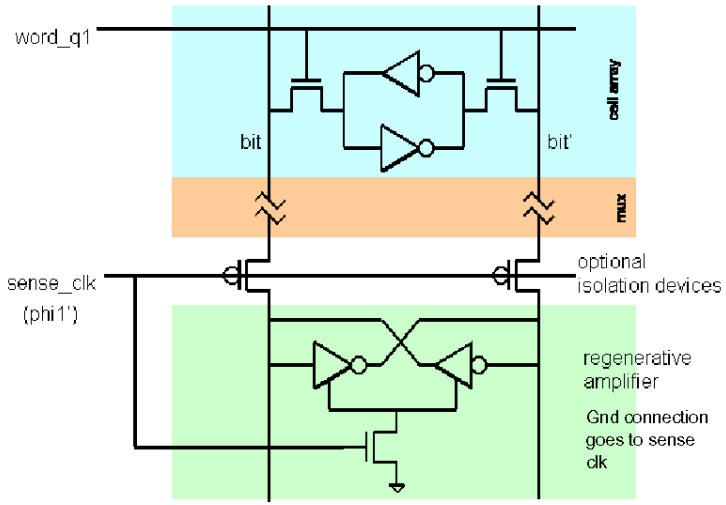
- Decrease delay by sensing smaller signals

  Noise margin is ok, most noise is common mode

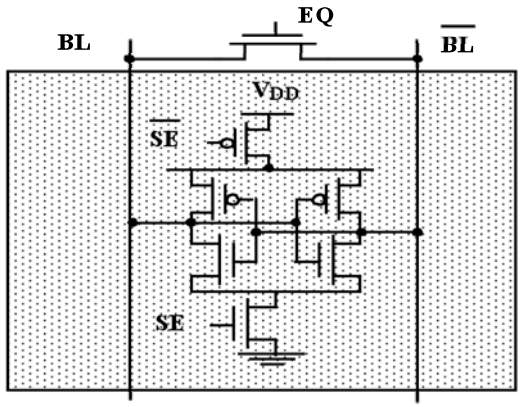
  Build a differential amplifier (sense amp)

  (Not needed in this class)

Sense Amp Circuit

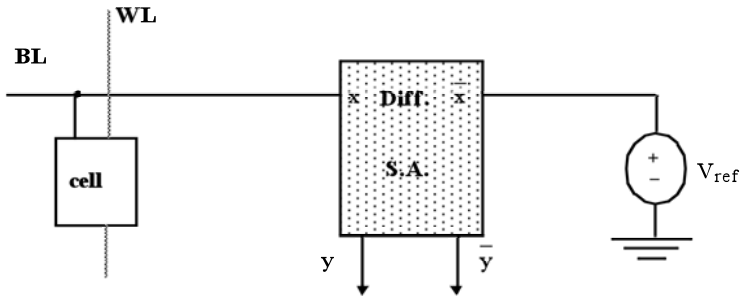


Latch-Based Sense Amplifier



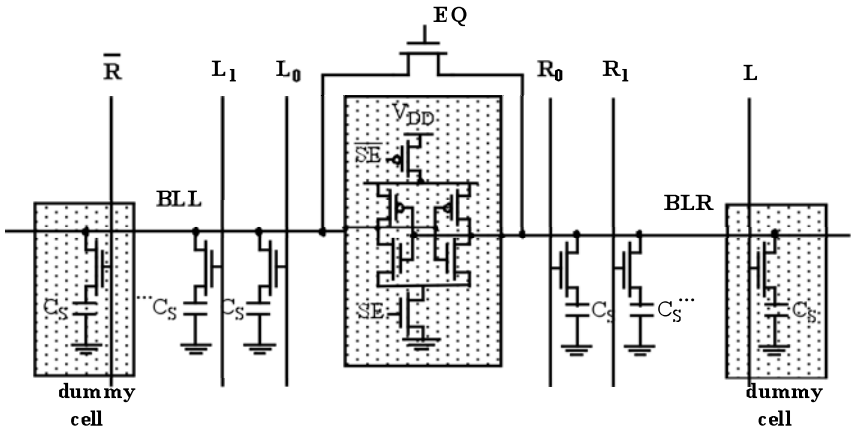
Initialized in its meta-stable point with EQ
Once adequate voltage gap created, sense amp enabled with SE
Positive feedback quickly forces output to a stable operating point.

Single-to-Differential Conversion

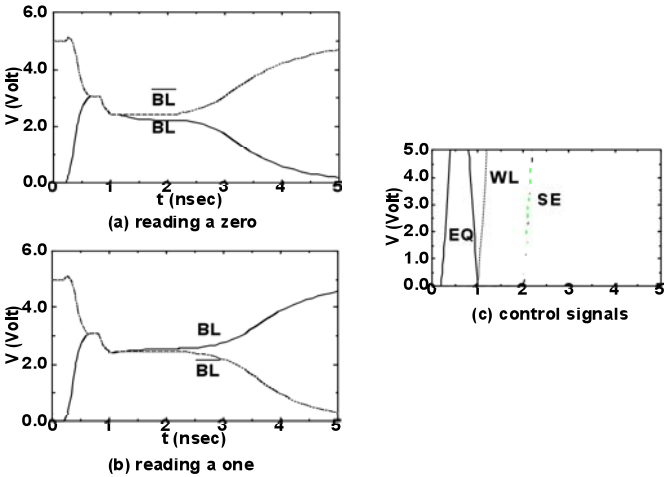


How to make good V_{ref}?

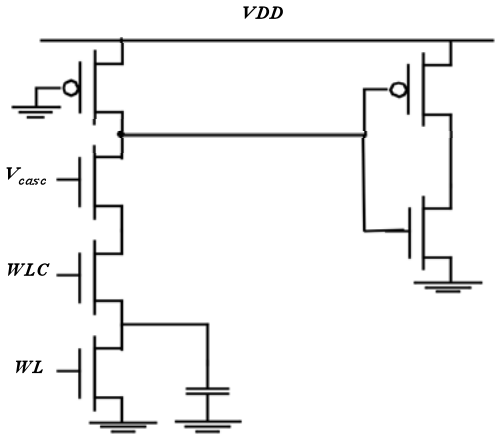
Open bitline architecture



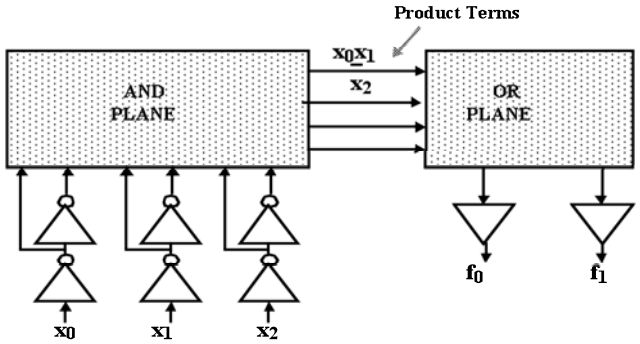
DRAM Read Process with Dummy Cell



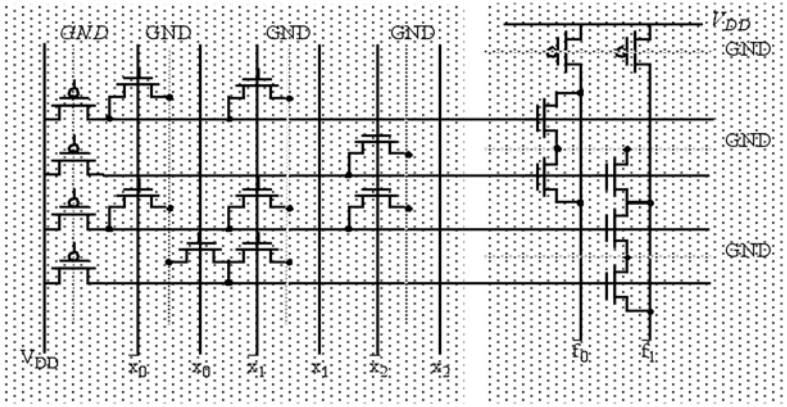
Single-Ended Cascode Amplifier



Programmable Logic Array

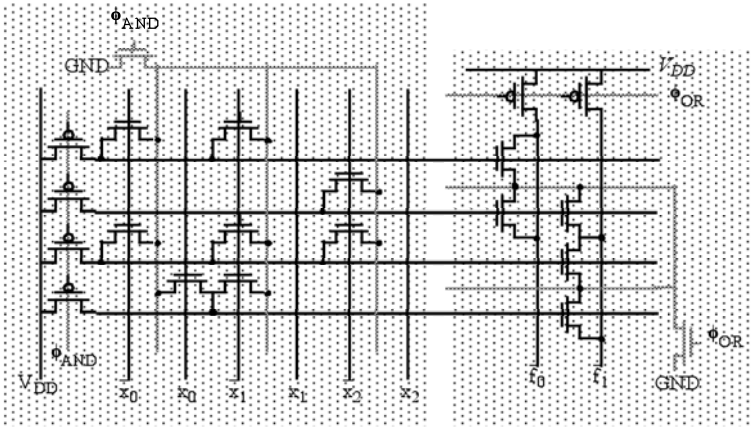


Pseudo-Static PLA



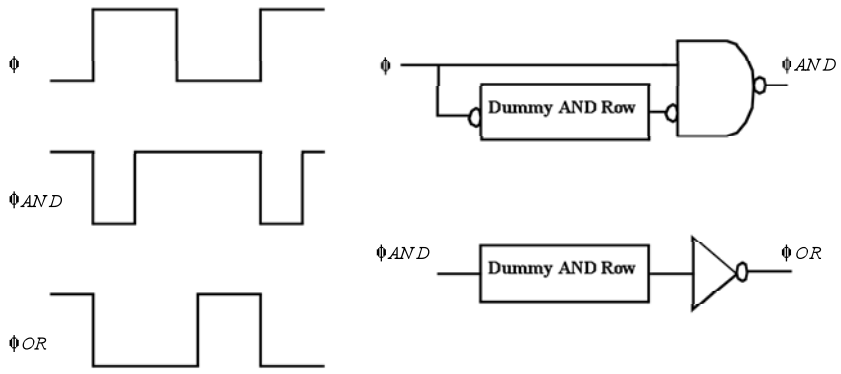
AND-PLANE OR-PLANE

Dynamic PLA



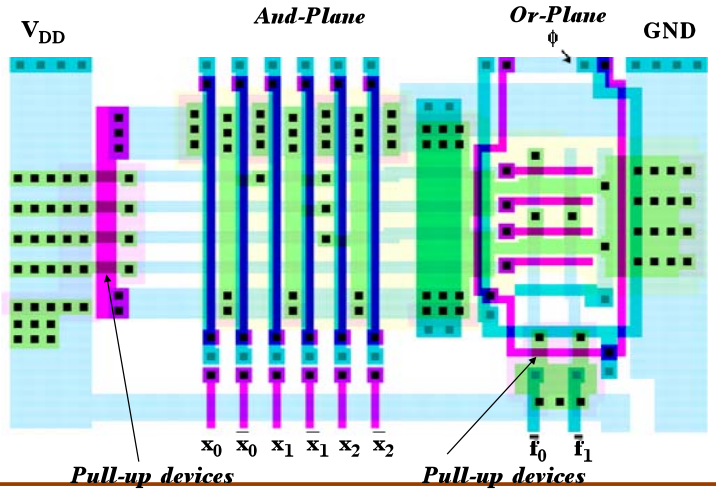
AND-PLANE OR-PLANE

Clock Signal Generation for self-timed dynamic PLA



(a) Clock signals (b) Timing generation circuitry.

PLA Layout



PLA versus 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)

