VLSI Memory Design

Shmuel Wimer
Bar Ilan University, School of Engineering

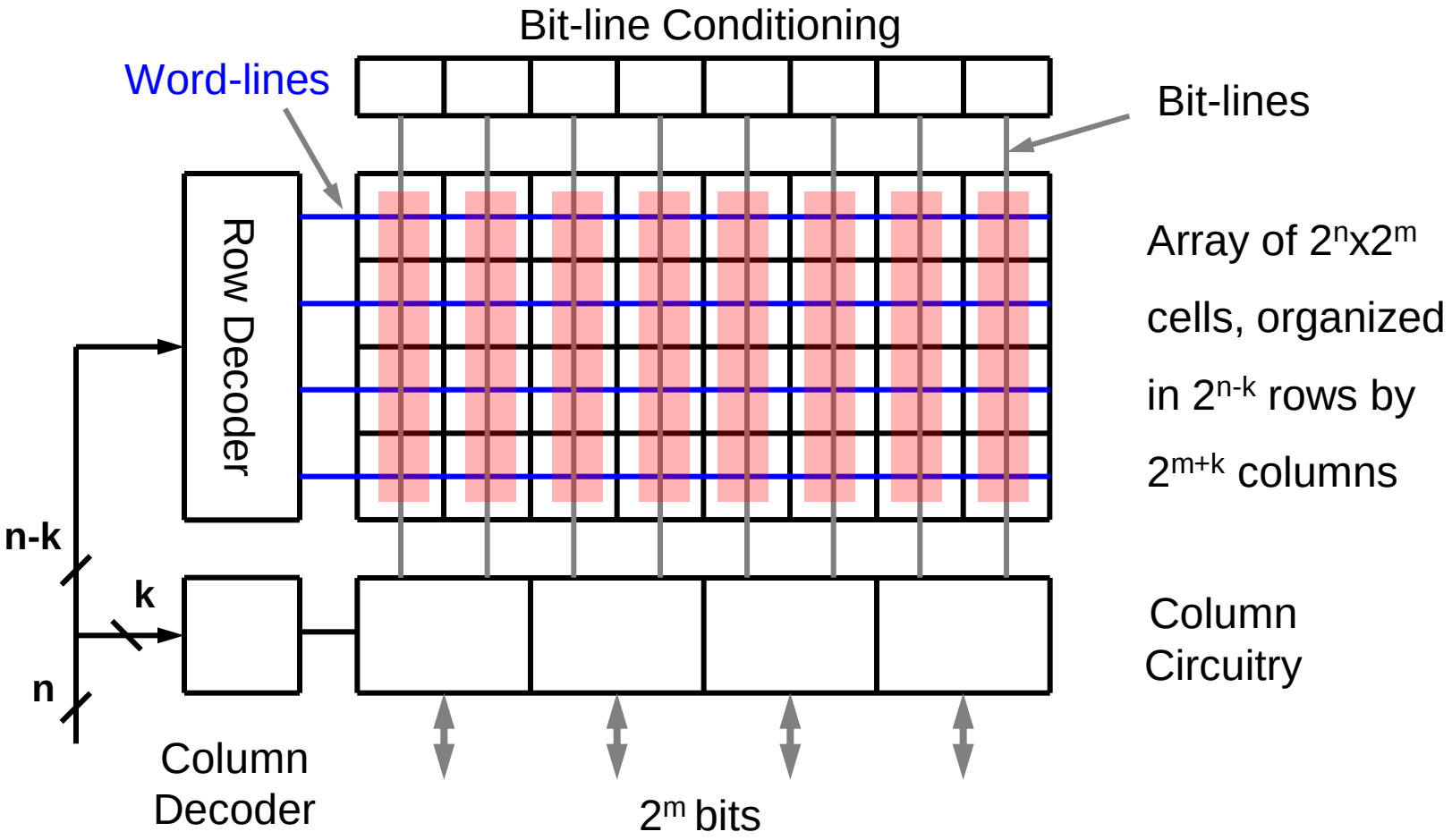
A memory has 2^n words of 2^m bits each. Usually $2^n >> 2^m$, (e.g. 1M Vs. 64) which will result a very tall structure.

The array is therefore folded into 2^{n-k} rows, each containing 2^k words, namely, every row contains 2^{m+k} bits.

Consider 8-words of 4-bit memory. We'd like to organize it in 4 lines and 8 columns. The memory is folded into 4-word by 8-bit, so n=3, m=2 and k=1.

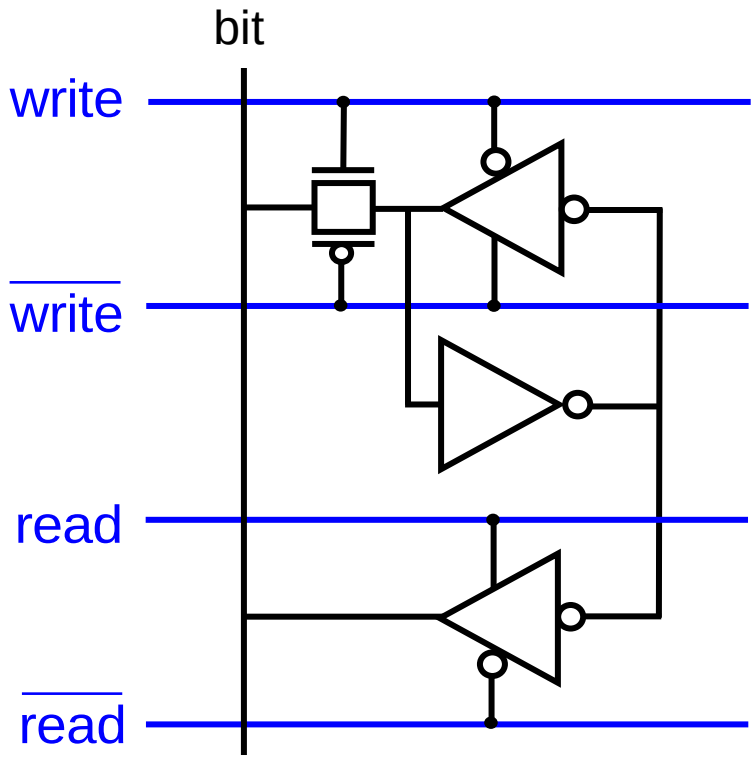
Larger memories are built from smaller sub-arrays to maintain short word and bit lines.

General Memory architecture



4-word by 8-bit folded memory

12-Transistor SRAM Cell

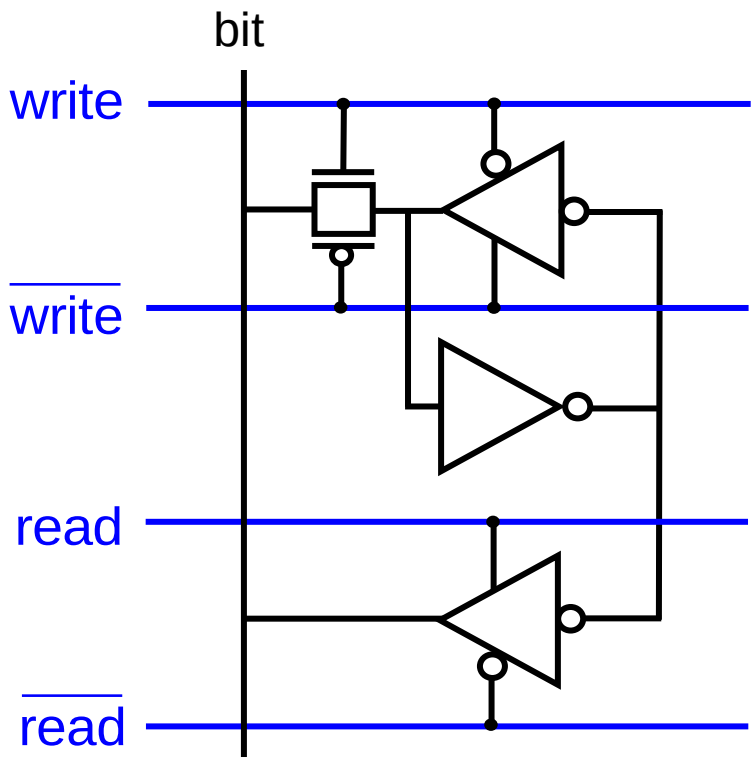


When write=1 the value at the bit is passed to the middle inverter while the upper tri-state inverter is in high Z.

Once write=0 the upper and the center inverters are connected in a positive feedback loop to retain cell's value as long as write=0.

The value of bit-line needs to override the value stored at the cell. It requires careful design of transistor size for proper operation.

12-Transistor SRAM Cell

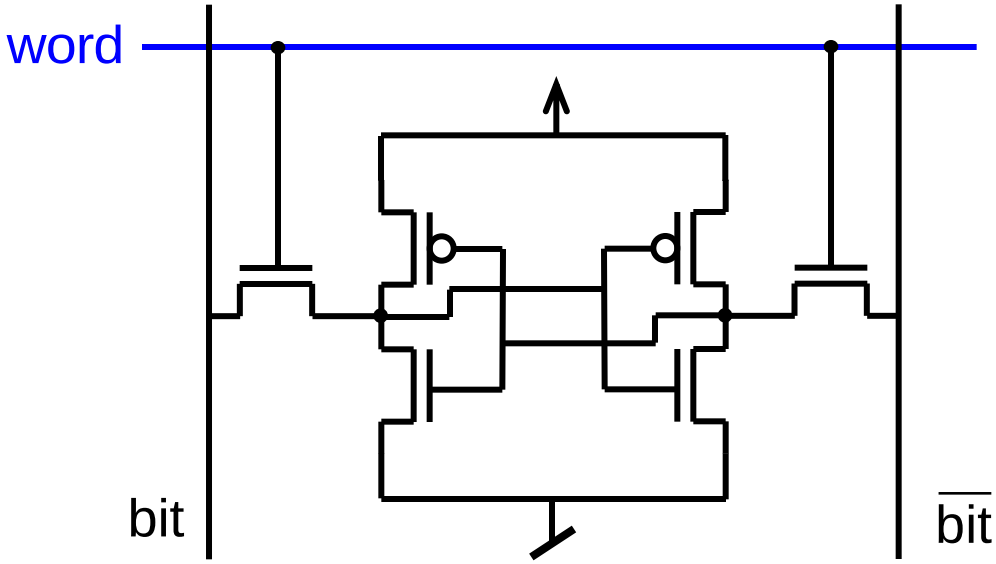


When read=1 the output of the lower tri-state inverter gets connected to the bit so cell's value appears on the bit-line.

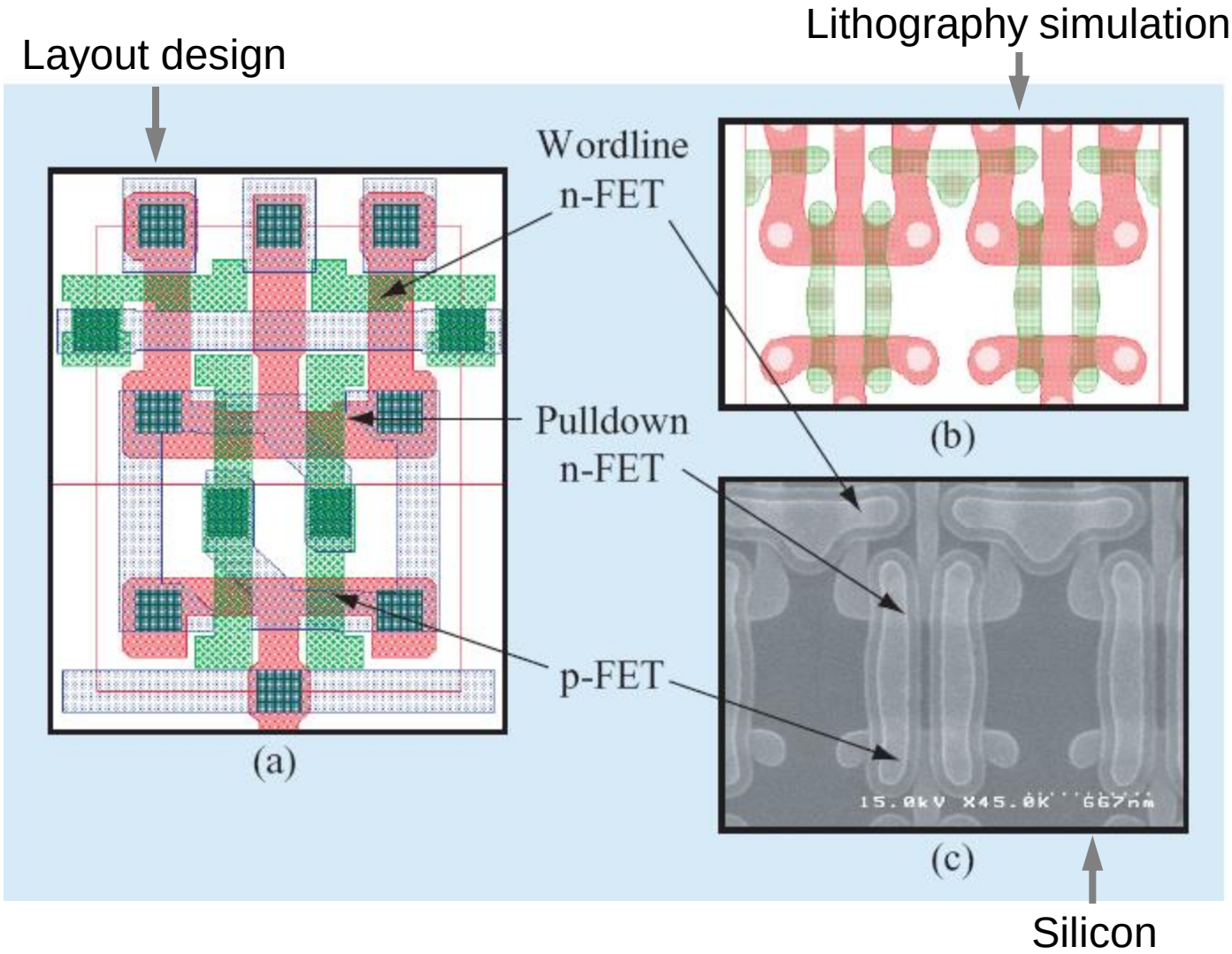
The bit-line is first pre-charged to one, so only if the value stored at cell is zero the bit-line is pulled down.

Though robust, 12-transistor cell consumes large area. Since it dominates the SRAM area, a 6-transistor is proposed, where some of the expense is charged on the peripheral circuits.

6-Transistor SRAM Cell

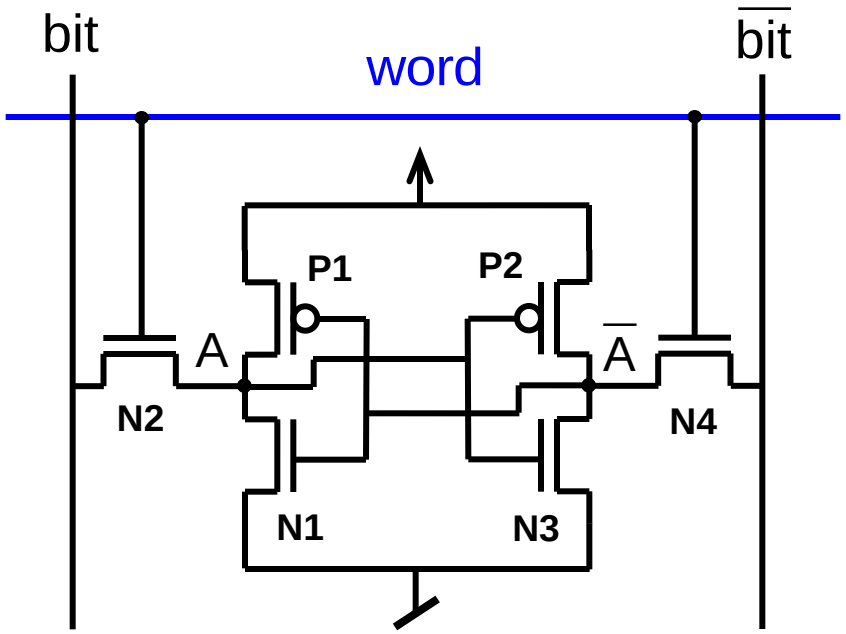


Layout of IBM 0.18u SRAM cell



Read Write Operations

SRAM operation is divided into two phases called Φ1 and Φ2, which can be obtained by clk and its complement.



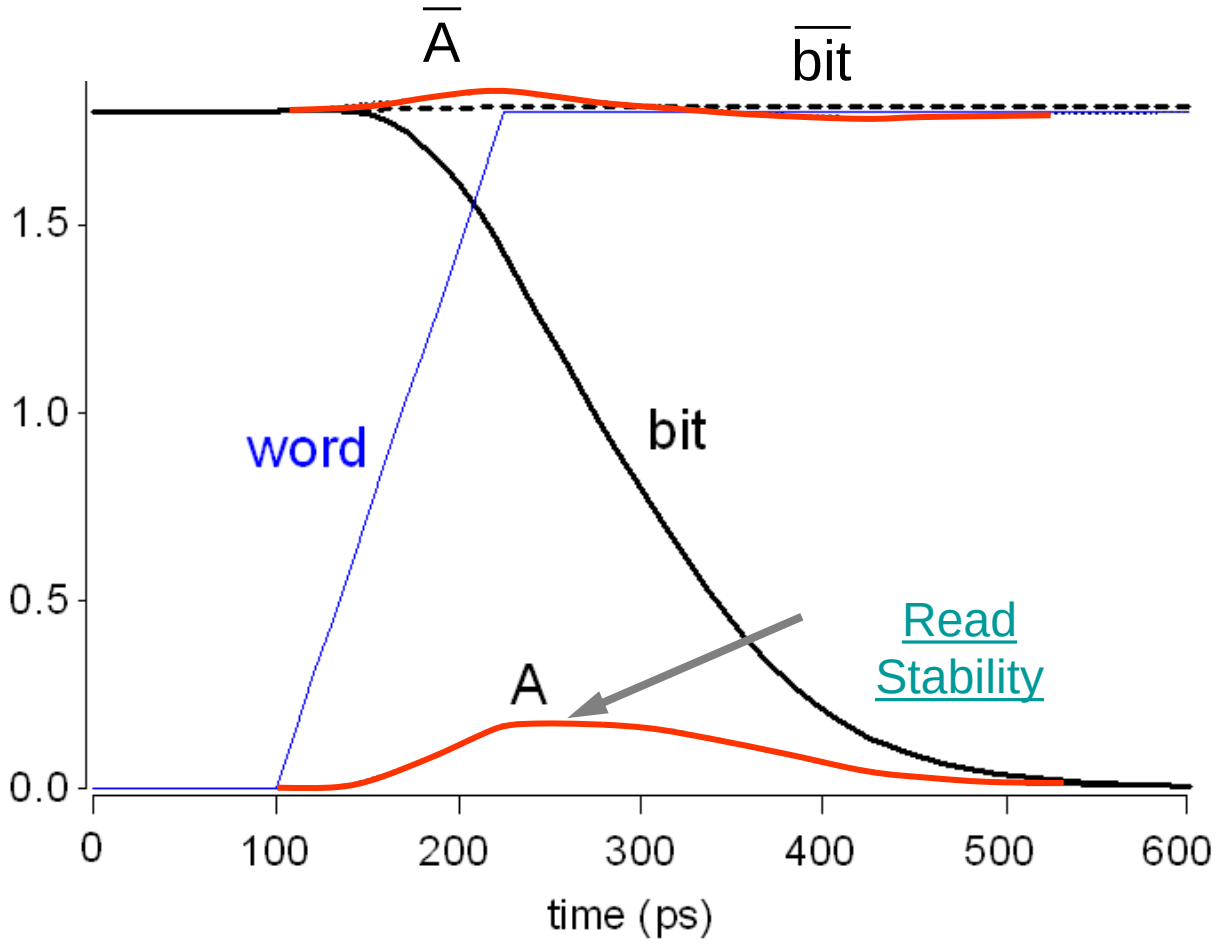
Pre-charge both bit-lines high.

Turn on word-line.

One of the bit-lines must be pulled-down.

Since bit-line was high, the 0 node will go positive for a short time, but must not go too high to avoid cell switch.

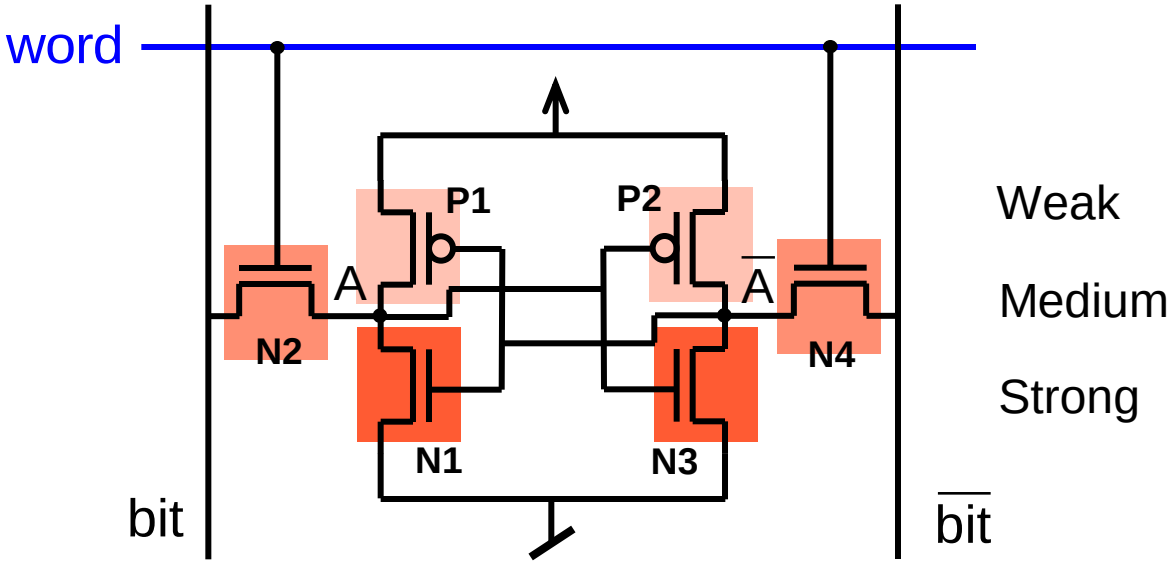
This is called ***read stability***.

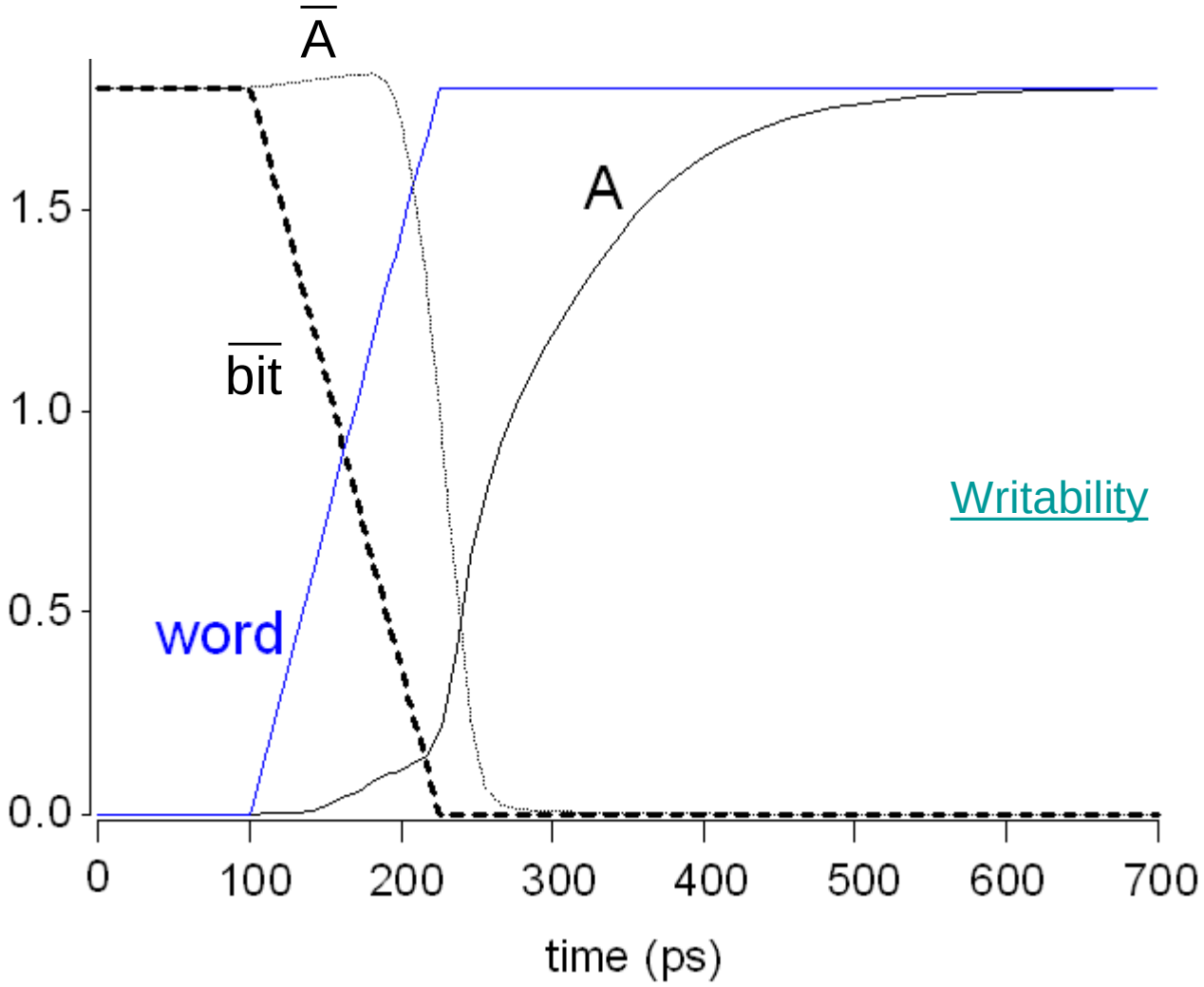


A must remain below threshold, otherwise cell may flip.
Therefore N1>>N2.

Let A=0 and assume that we write 1 into cell. In that case bit is pre-charged high and its complement should be pulled down.

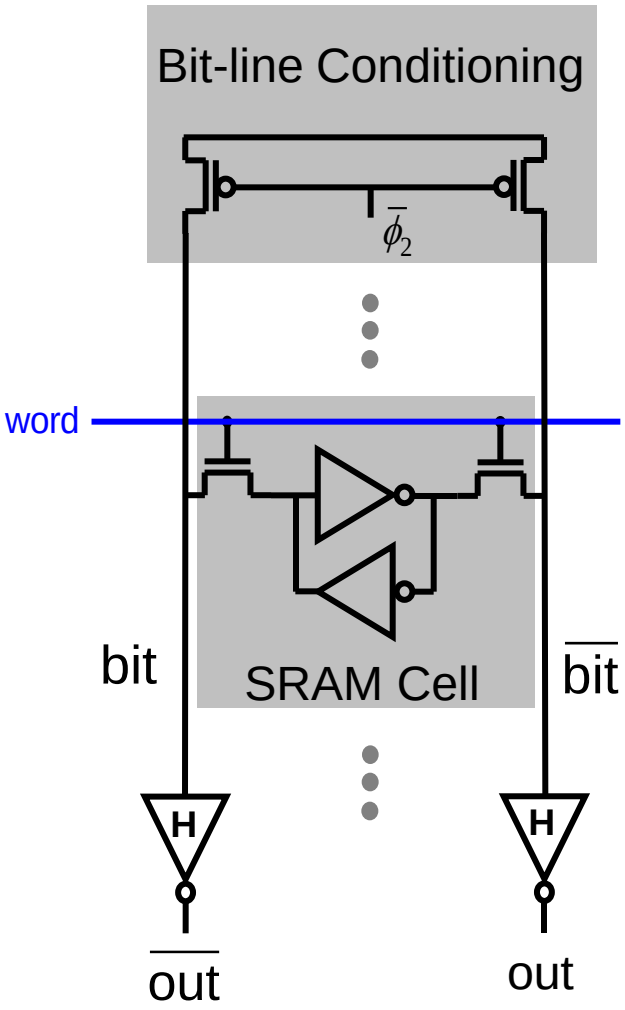
It follows from read stability that N1>>N2 hence A=1 cannot be enforced through N2. Hence A complement must be enforced through N4, implying N4>>P2. This constraint is called ***writability***.

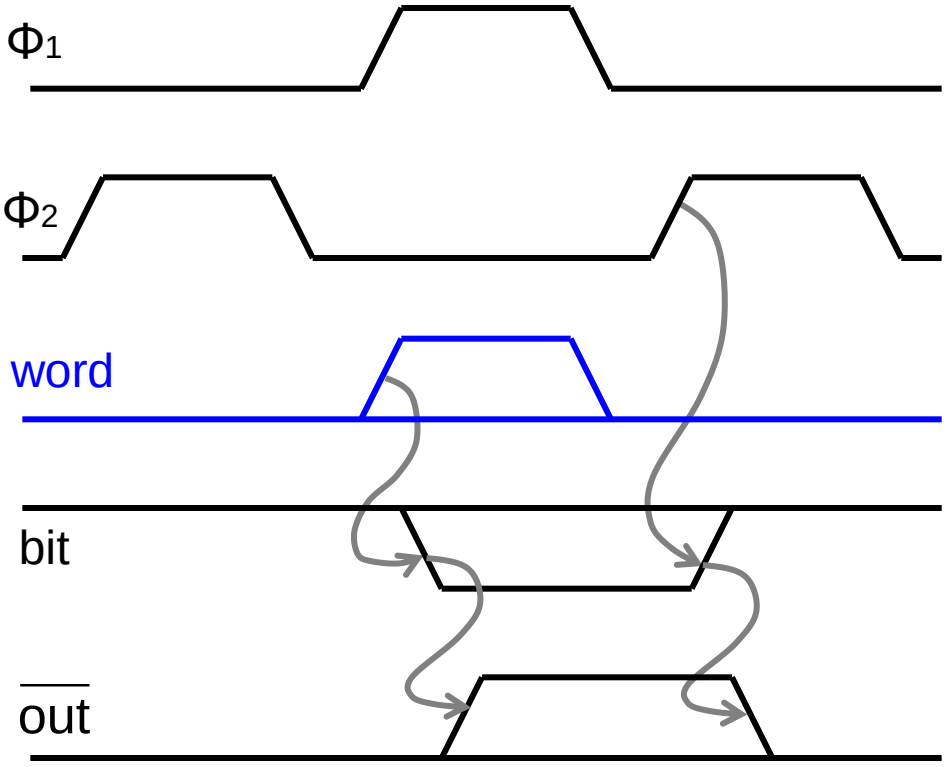




Writability

SRAM Column Read Operation

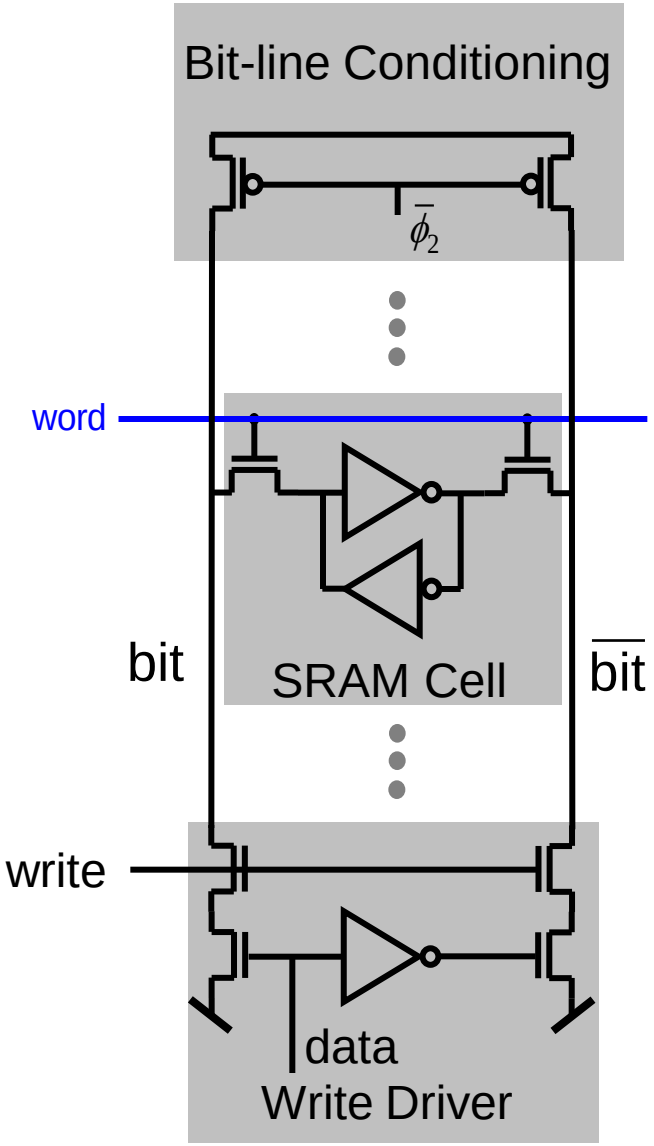




Bit-lines are precharged high

For delay reduction outputs can be sensed by high-skew inverters (low noise margin).

SRAM Column Write Operation



Bit-lines (and complements) are precharged high. At write one is pulled down.

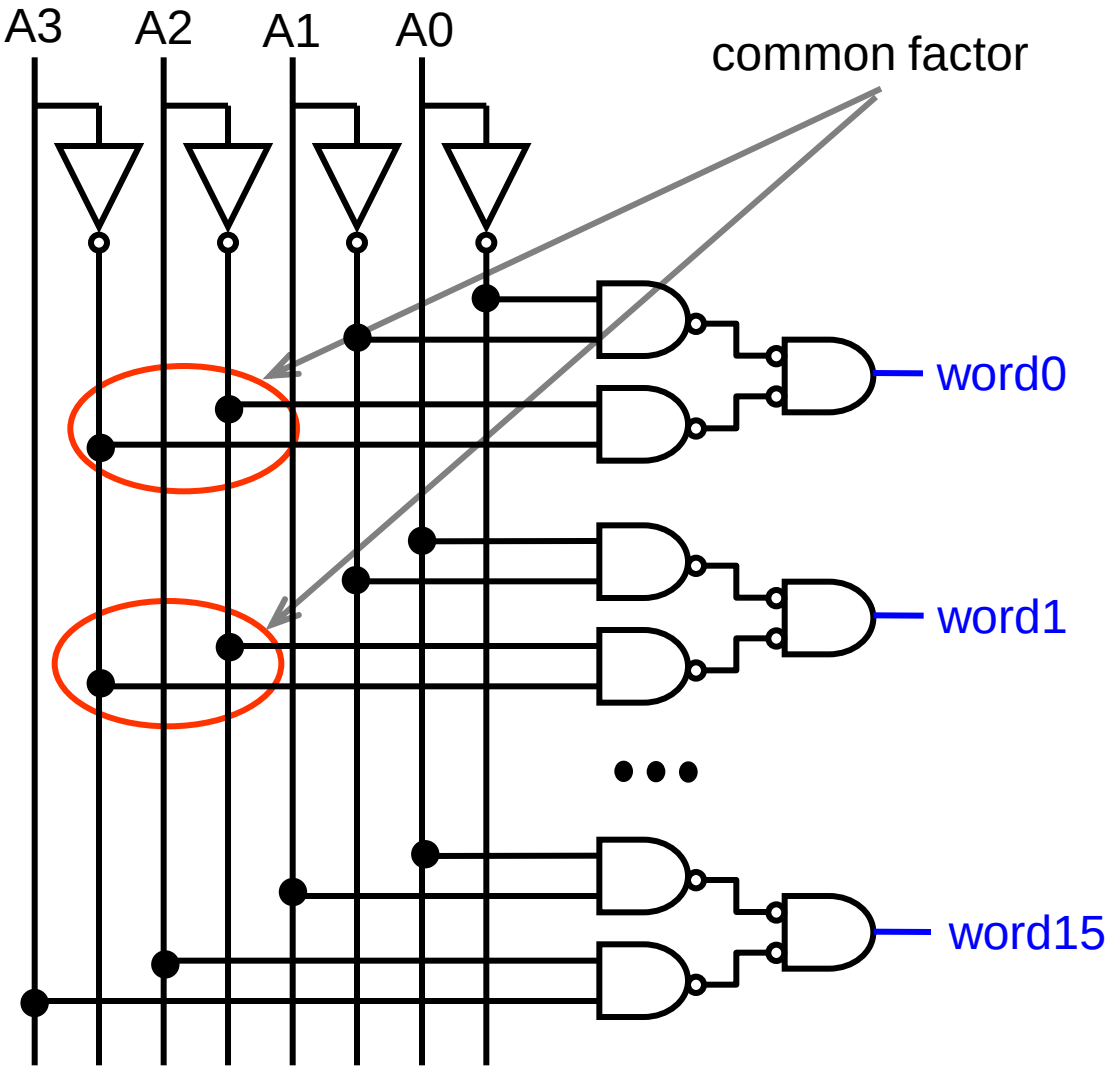
Write operation overrides one of the pMOS transistors of the loop inverters.

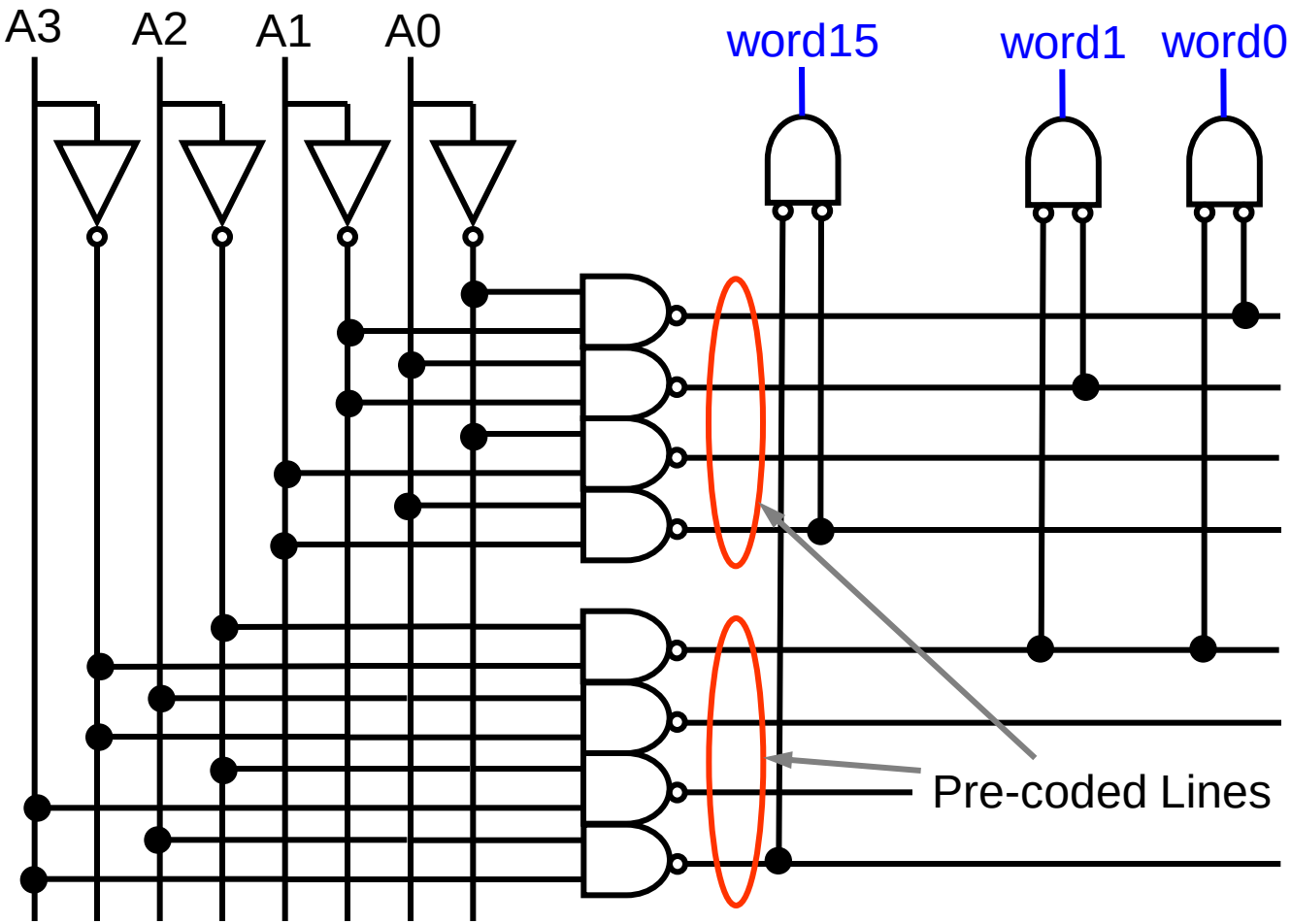
Therefore, the series resistance of transistors in write driver must be low enough to overpower the pMOS transistors.

Decoders

To decode word-lines we need AND gates of *n-k* inputs. This is a problem when fan-in of more than 4 since it slows down decoding.

It is possible to break the AND gates into few levels as shown in the 4:16 decoder.

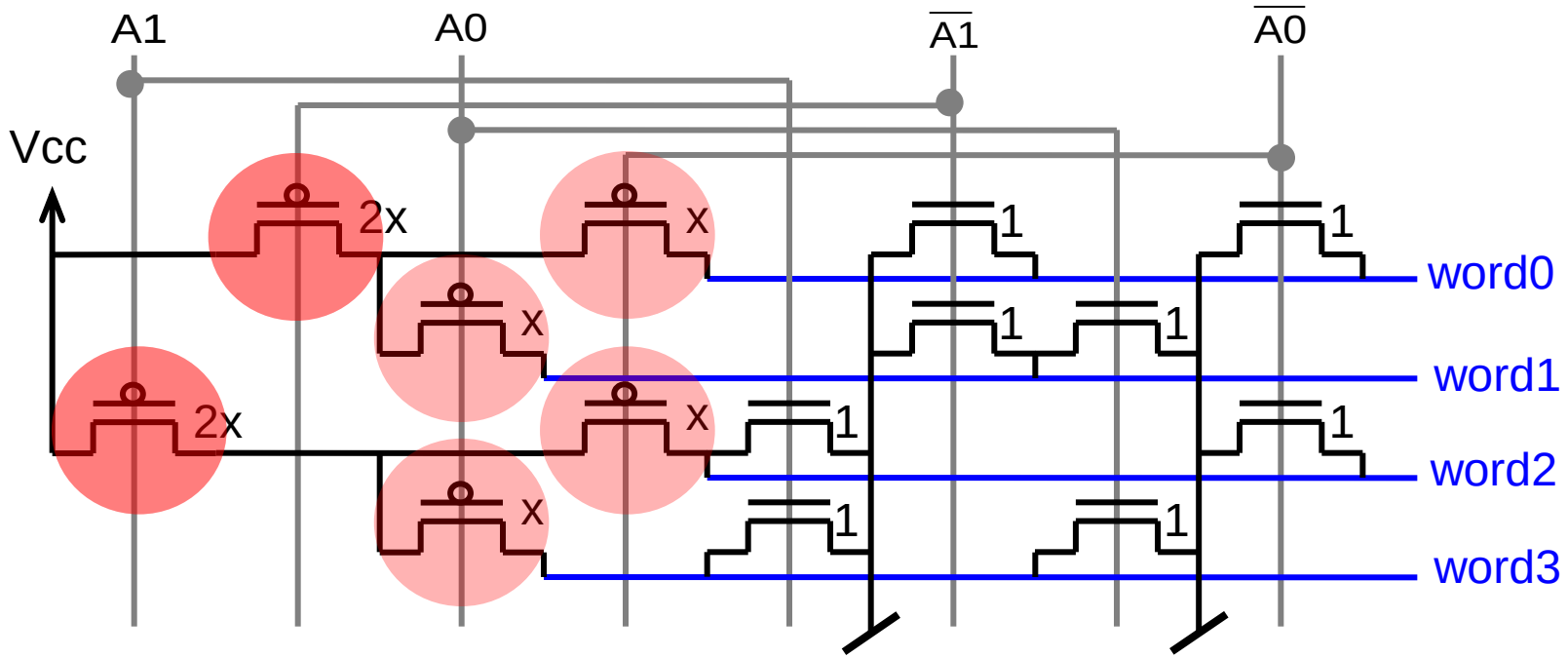




Terms repeat themselves, so pre-decoding will eliminate the redundant ones. This is called ***pre-decoding***. Less area with same drive as before.

Lyon-Schediwy Fast Decoders

In a NOR implementation output is pulled up via serial pMOS devices, which slows transition, so pMOS needs sizing, but this consumes lot of area. Since only single word is pulled up at a time, pMOS can be shared between words in a binary tree fashion and sized to yield same current as in pull down.



Sum-addressed Decoders

Sometimes an address of memory is calculated as BASE+OFFSET (e.g., in a cache), which requires an addition before decoding.

Addition can be time consuming if Ripple Carry Adder (RCA) is used, and even Carry Look Ahead (CLA) my be too slow.

It is possible to use a $K = A + B$ comparator without carry propagation or look-ahead calculation.

Sum-addressed Decoders

If we know *A* and *B*, we can deduce what must be the carry in of every bit if it would happen that $K = A + B$.

But then we can also deduce what should be the carry out.

It follows that if every bit pair agrees on the carry out of the previous with the carry in of the next, then $K=A+B$ is true indeed.

We can therefore use a comparator to every word-line (k), where equality will hold only for one word.

We can derive the equations of the carries from the required and generated carries below.

A_i	B_i	K_i	C_{in_i} (required)	C_{out_i} (generated)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Theorem: If for every $1 \le i \le n$ $C_{\text{in\_}(i+1)} = C_{\text{out\_}i}$,

then $A + B = K$.

Proof : It follows from the truth table that:

(1) $C_{\text{in\_}i} = A_i \oplus B_i \oplus K_i$ and

(2) $C_{\text{out\_}i} = \left(A_i \oplus B_i\right)\bar{K}_i + A_i B_i$.

We'll show that for every $1 \le i \le n$,

$z_i \triangleq \left(C_{\text{in\_}i} == C_{\text{out\_}(i-1)}\right)$ implies $e_i \triangleq \left[\left(A+B\right)_i == K_i\right]$,

which will prove the theorem.

$$z_i \triangleq \left(C_{\text{in}\_i} == C_{\text{out}\_(i-1)} \right) \text{ implies}$$

$$(3)\ z_i = 1 \Leftrightarrow \overline{C_{\text{in}\_i} \oplus C_{\text{out}\_(i-1)}} = 1.$$

$$e_i \triangleq \left[\left(A + B \right)_i == K_i \right] \text{ implies}$$

$$(4)\ e_i = 1 \Leftrightarrow \overline{\left(A_i \oplus B_i \oplus C_{\text{in}\_i} \right) \oplus K_i} = 1.$$

Assume that $z_i = 1.$

Substitution of (1) and (2) in (3) yields

$$(5)\ \overline{\left[A_i \oplus B_i \oplus K_i \right] \oplus \left[\left(A_{i-1} \oplus B_{i-1} \right) \bar{K}_{i-1} + A_{i-1} B_{i-1} \right]} = 1.$$

By induction the theorem holds for $i-1,$ hence

(6) $K_{i-1} = \left(A+B\right)_{i-1},$

which is $K_{i-1} = A_{i-1} \oplus B_{i-1} \oplus C_{\text{in\_}(i-1)}.$

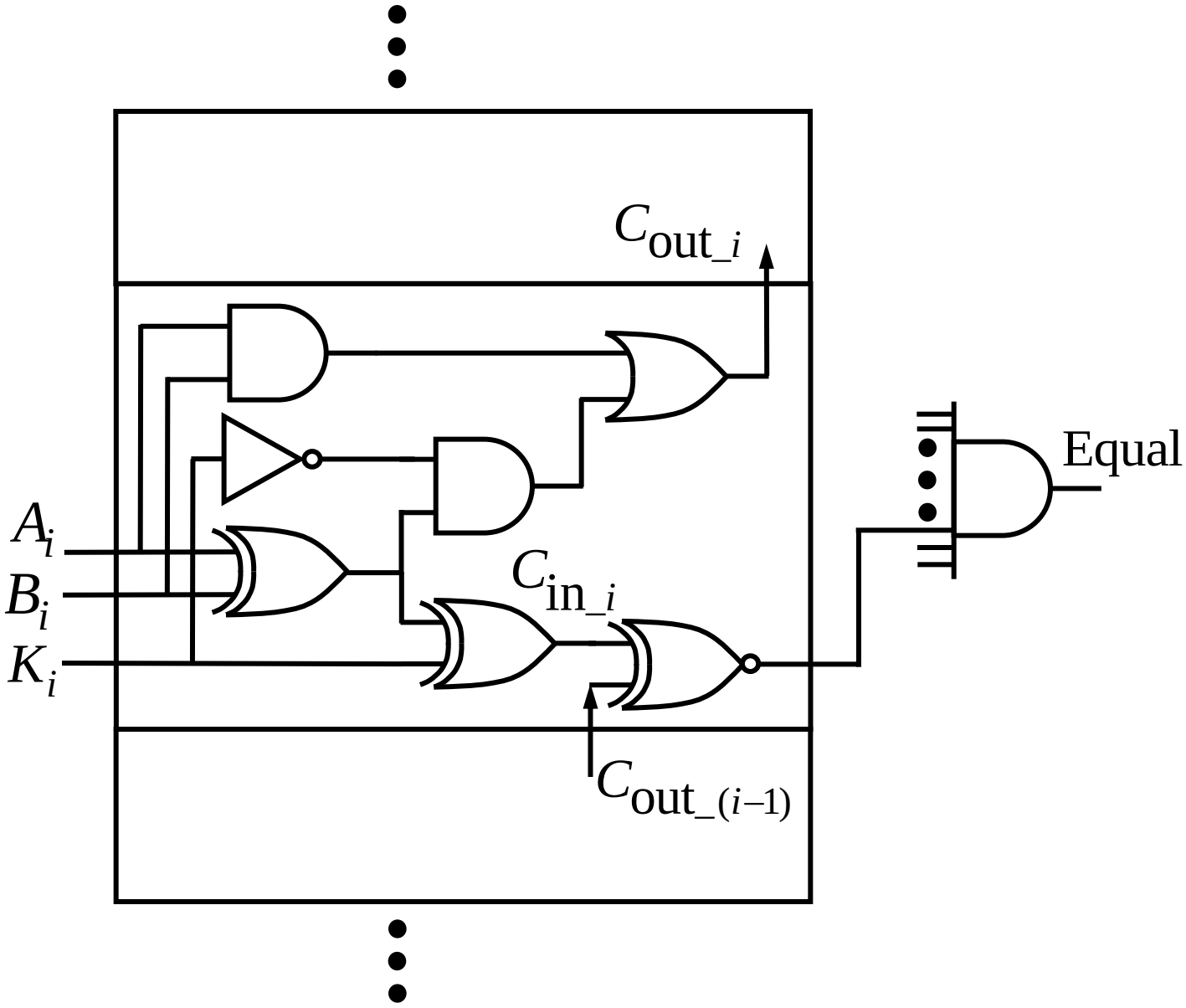
Substitution of (6) in the second brakets of (5) and

further manuipulations turns the braket into

(7) $\left[\left(A_{i-1} \oplus B_{i-1}\right)\bar{K}_{i-1} + A_{i-1}B_{i-1}\right] = C_{\text{out\_}(i-1)} = C_{\text{in\_}i}.$

which then turns (5) into

$\overline{\left[A_i \oplus B_i \oplus K_i\right] \oplus C_{\text{in\_}i}} = 1,$ implying $e_i = 1.$ ●

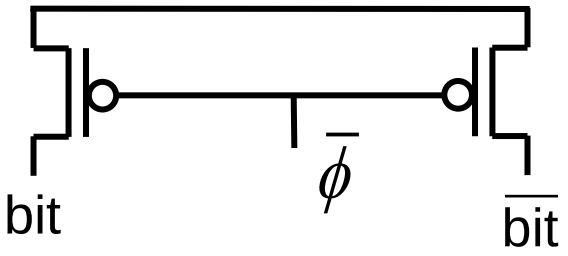

C_{out_i}
A_i
B_i
K_i
C_{in_i}
Equal
$C_{out\_(i-1)}$

Below is a comparison of sum-addressed decoder with ordinary decoder combined with a ripple carry adder (RCA) and carry look ahead adder (CLA). A significant delay and area improvement is achieved.

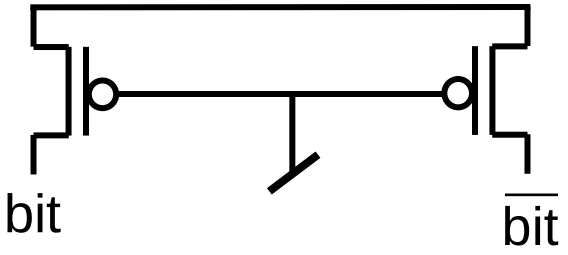
n	RCA		CLA		FAC	
	Delay	Area	Delay	Area	Delay	Area
8	26.95	0.28	23.35	0.45	13.53	0.30
16	45.75	0.58	29.22	1.16	14.56	0.64
32	85.47	1.53	37.51	3.59	15.63	1.53
64	164.17	3.79	44.80	8.71	17.80	3.55
128	320.30	10.80	53.54	24.22	18.93	9.80

Bit-Line Conditioning Circuits

Used to precharge bits high before R/W operation. Most simple is the following circuit.

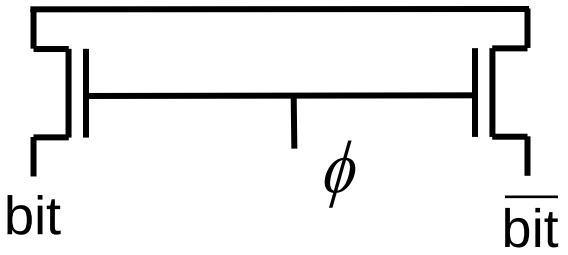


If a clock is not available it is possible to use a weak pMOS device connected as a pseudo-nMOS SRAM.



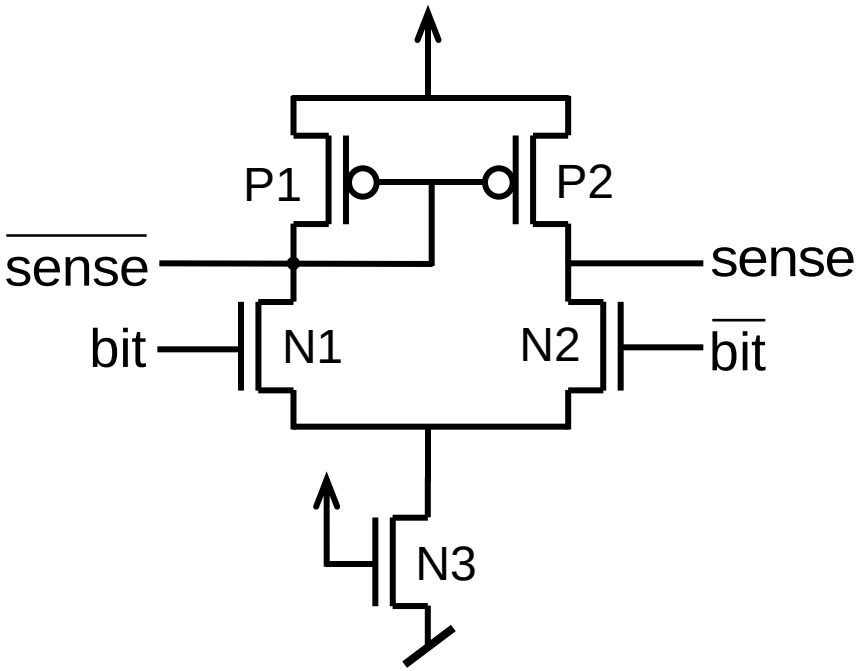
Precharge can be done with nMOS, a case where precharge voltage is V_{dd}-V_t.

It results faster R/W since swing is smaller, but noise margin is worsen.

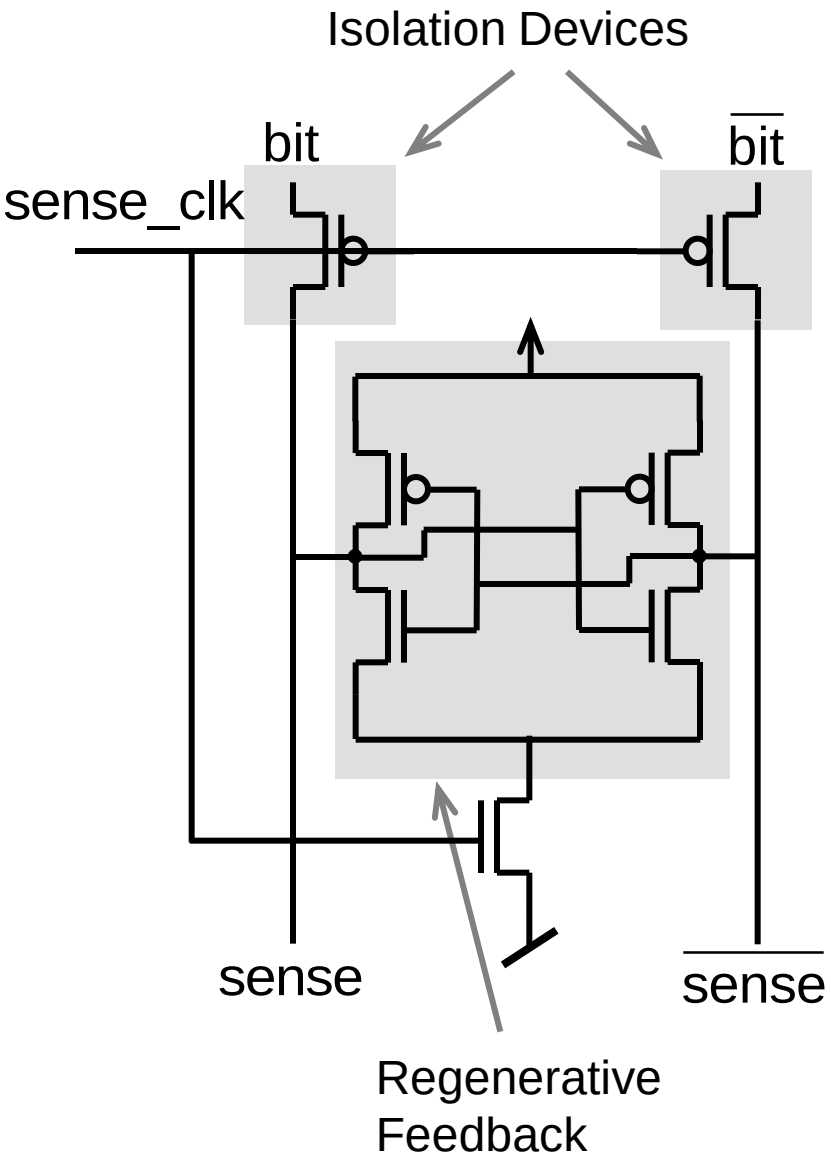


Sense Amplifiers

Each column contains write driver and read sensing circuit. A high skew read inverter has been shown. Sense amplifier provides faster sensing by responding to a smaller voltage swing.



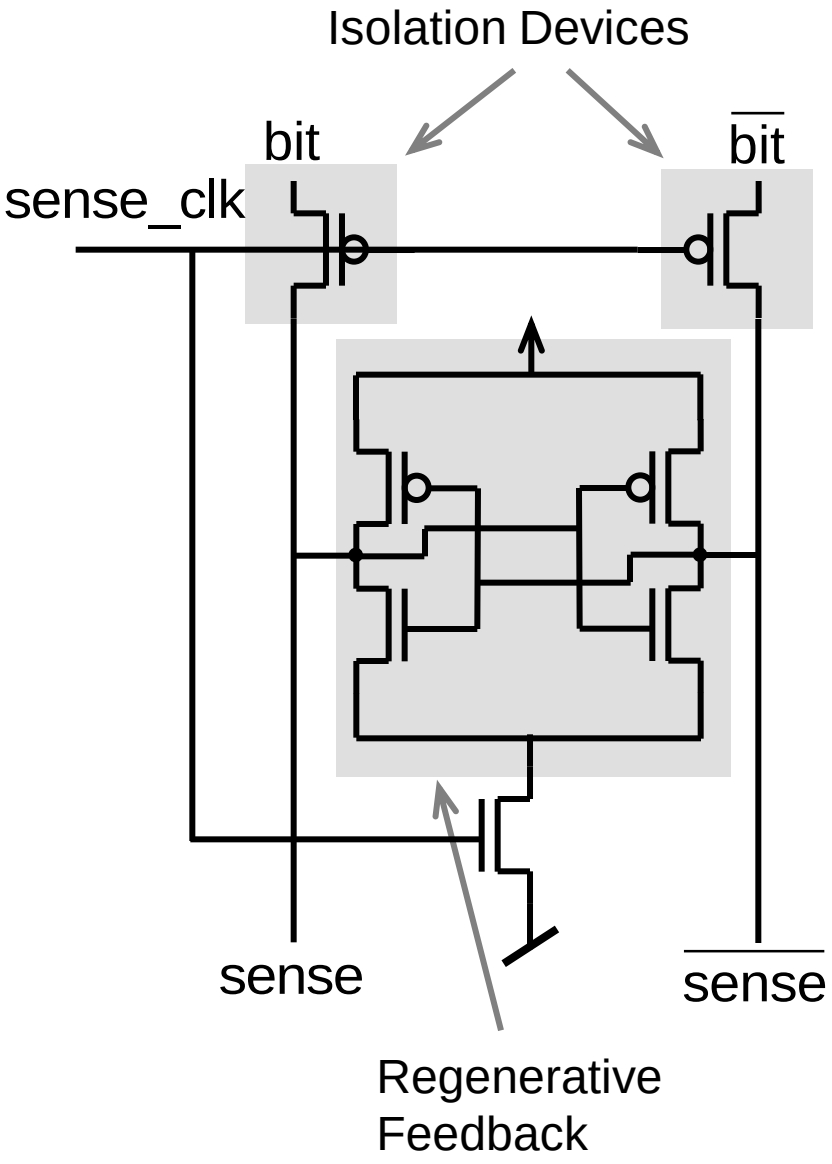
This is a differential analog pair. N3 is a current source where current flows either in left or right branches. Circuit doesn't need a clock but it consumes significant amount of DC power.



To speed up response bit-lines are disconnected at sensing to avoid their high capacitive load.

The regenerative feedback loop is now isolated.

When sense clock is high the values stored in bit-lines are regenerated, while the lines are disconnected, speeding up response.



Sense amplifiers are susceptible to differential noise on bit-lines since they respond to small voltage differences.

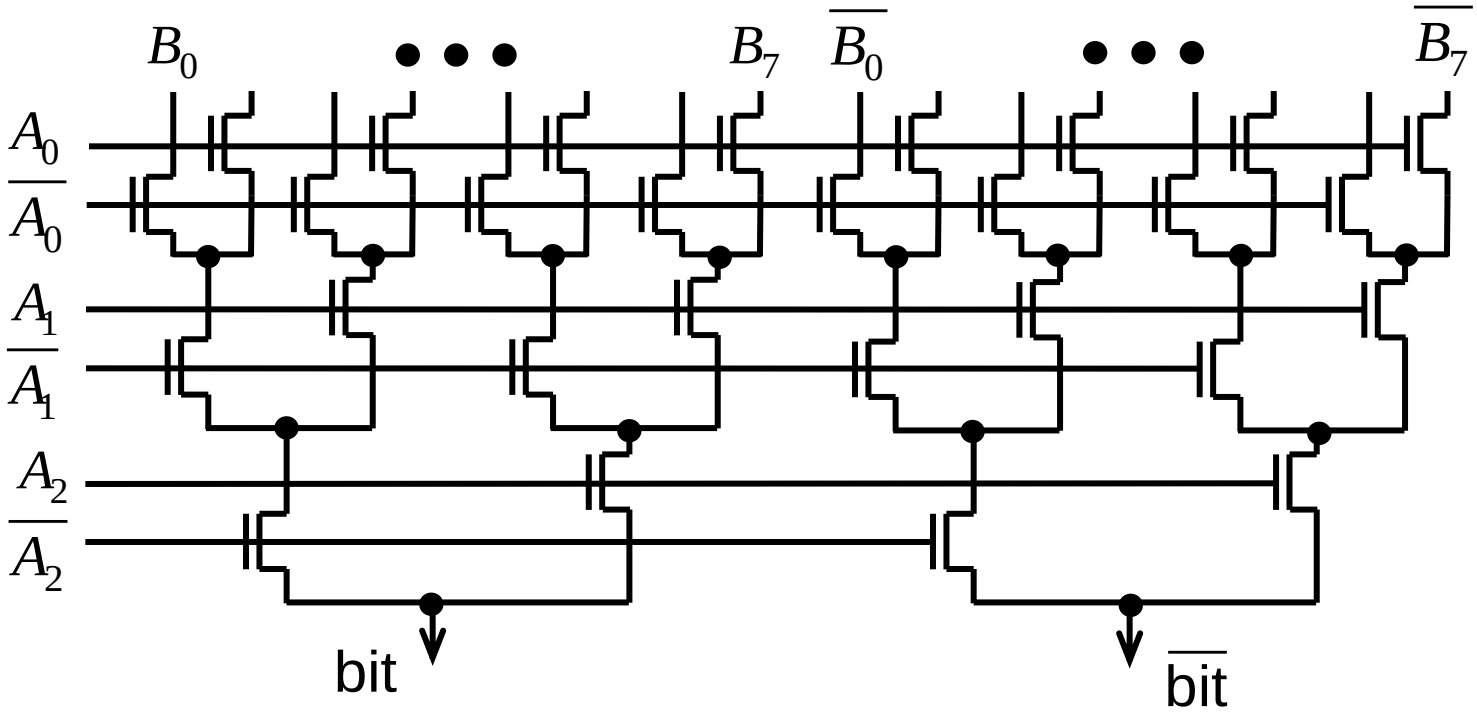
Column Multiplexers

The SRAM is physically organized by 2^{n-k} rows and 2^{m+k} columns.

Each row has 2^m groups of 2^k bits.

Therefore, a 2^k:1 column multiplexers are required to extract the appropriate 2^m bits from the 2^{m+k} ones.

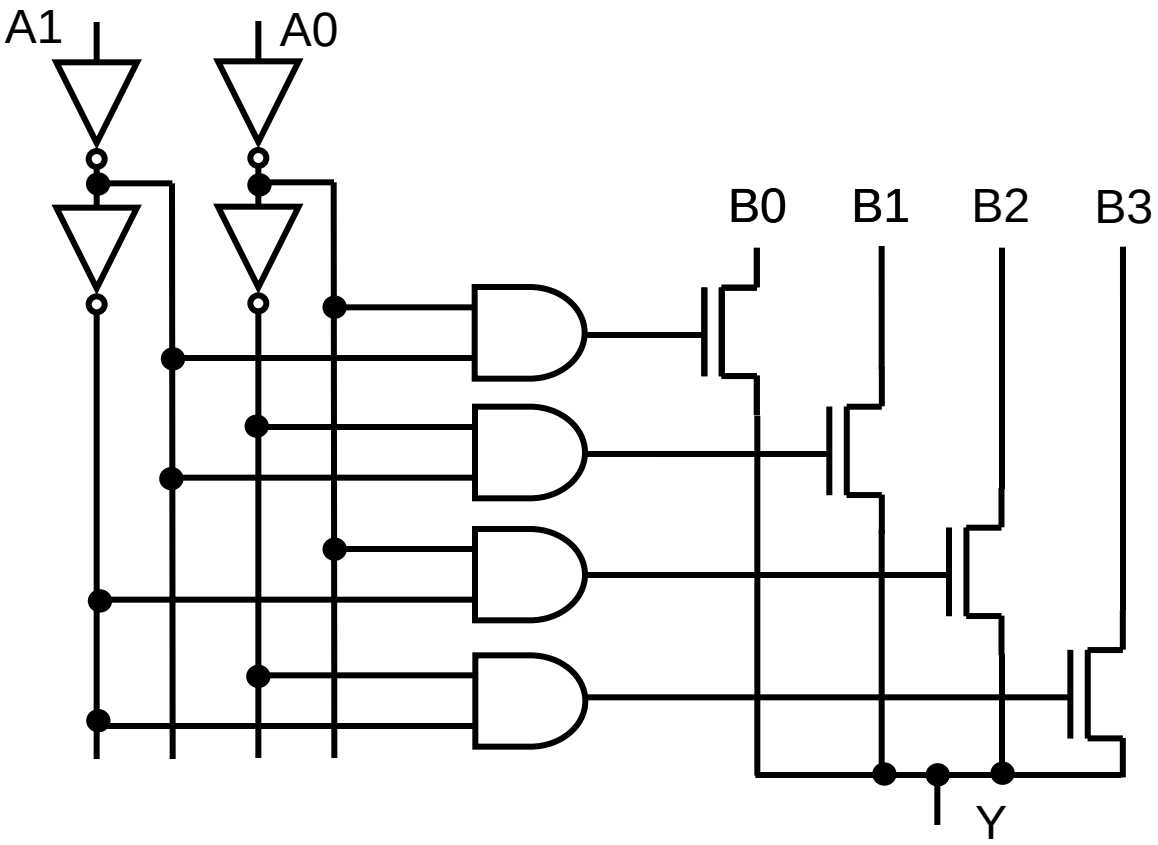
Tree Decoder Column Multiplexer



To sense Amps and Write Circuits

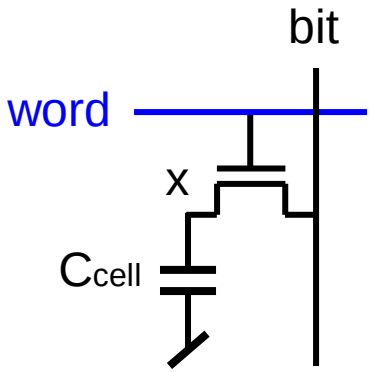
The problem of this MUX is the delay occurring by the series of pass transistors.

It is possible to implement the multiplexer such that data is passed trough a single transistor, while column decoding takes place concurrently with row decoding, thus not affecting delay.



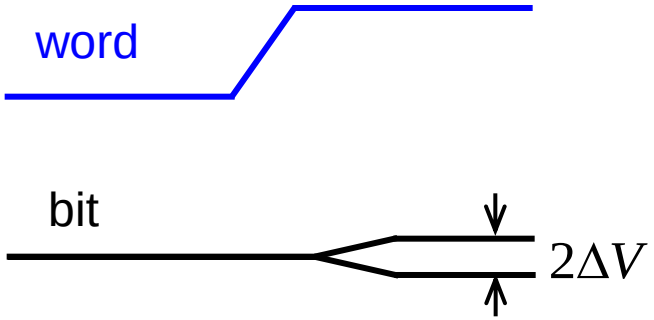
DRAM – Dynamic RAM

- Store their charge on a capacitor rather than in a feedback loop
- Basic cell is substantially smaller than SRAM.
- To avoid charge leakage it must be periodically read and refresh
- It is built in a special process technology optimized for density
- Offers order of magnitude higher density than SRAM but has much higher latency than SRAM

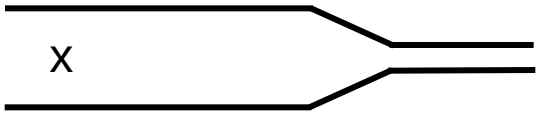


A 1-transistor (1T) DRAM cell consists of a transistor and a capacitor.

Cell is accessed by asserting the word-line to connect the capacitor to the bit-line.



On a read the bit-line is first precharged to $V_{DD}/2$. When the word-line rises, the capacitor shares its charge with the bit-line, causing a voltage change of ΔV that can be sensed.



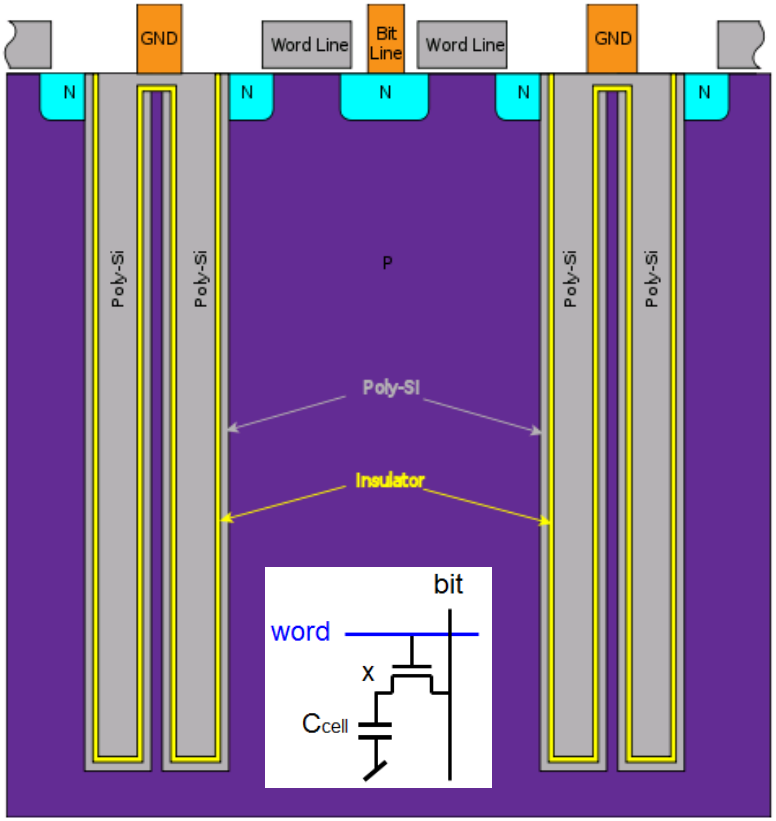
The read disturbs the cell contents at x, so the cell must be re-written after each read.

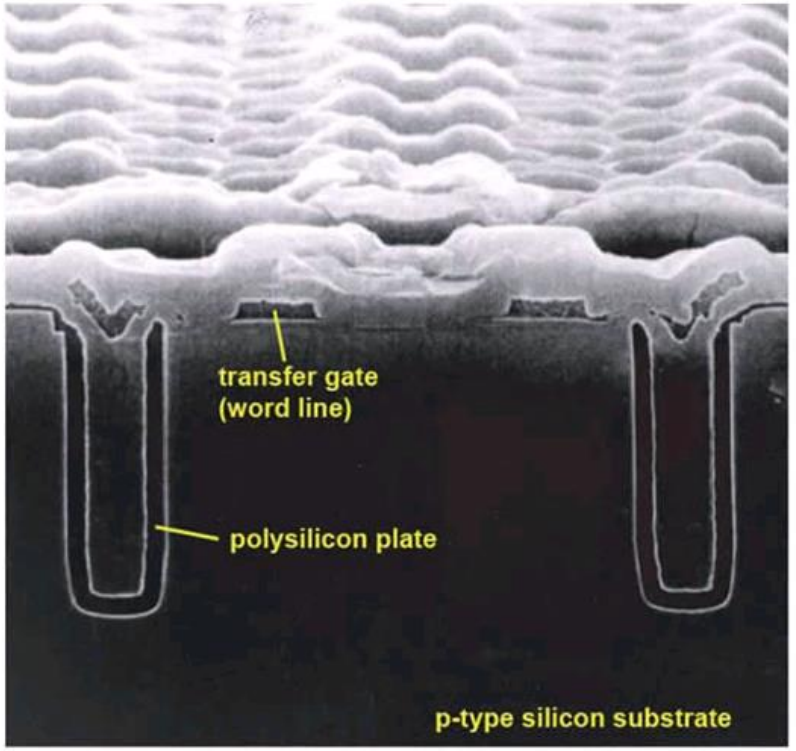
On a write the voltage of the bit-line is forced onto the capacitor

DRAM Cell

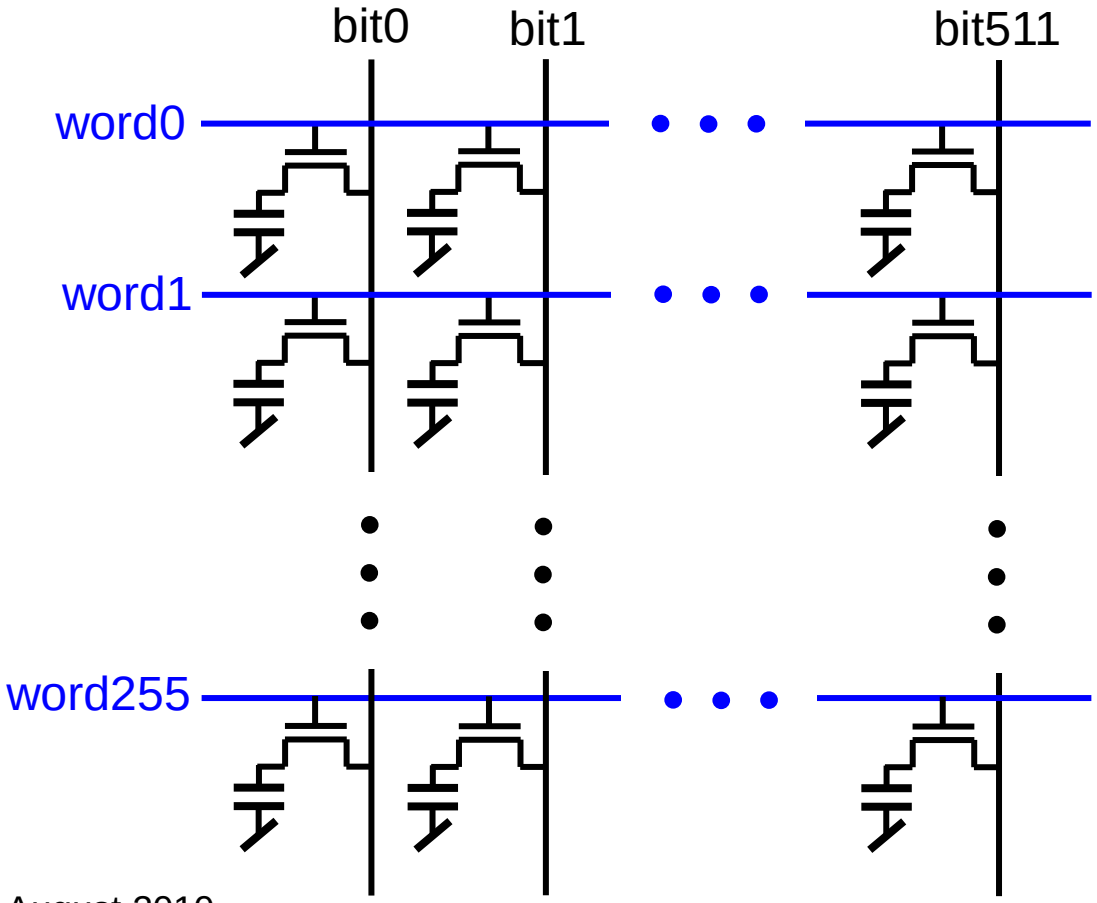
$$\Delta V = \frac{V_{DD}}{2} \frac{C_{cell}}{C_{cell} + C_{bit}}$$

C_{cell} must be small to obtain high density, but big enough to obtain voltage swing at read.





Like SRAMs, large DRAMs are divided into sub-arrays, whose size represents a tradeoff between area and performance. Large sub-arrays amortize sense amplifiers and decoders among more cells but are slower and have less swing due to higher capacitance of word and bit lines.

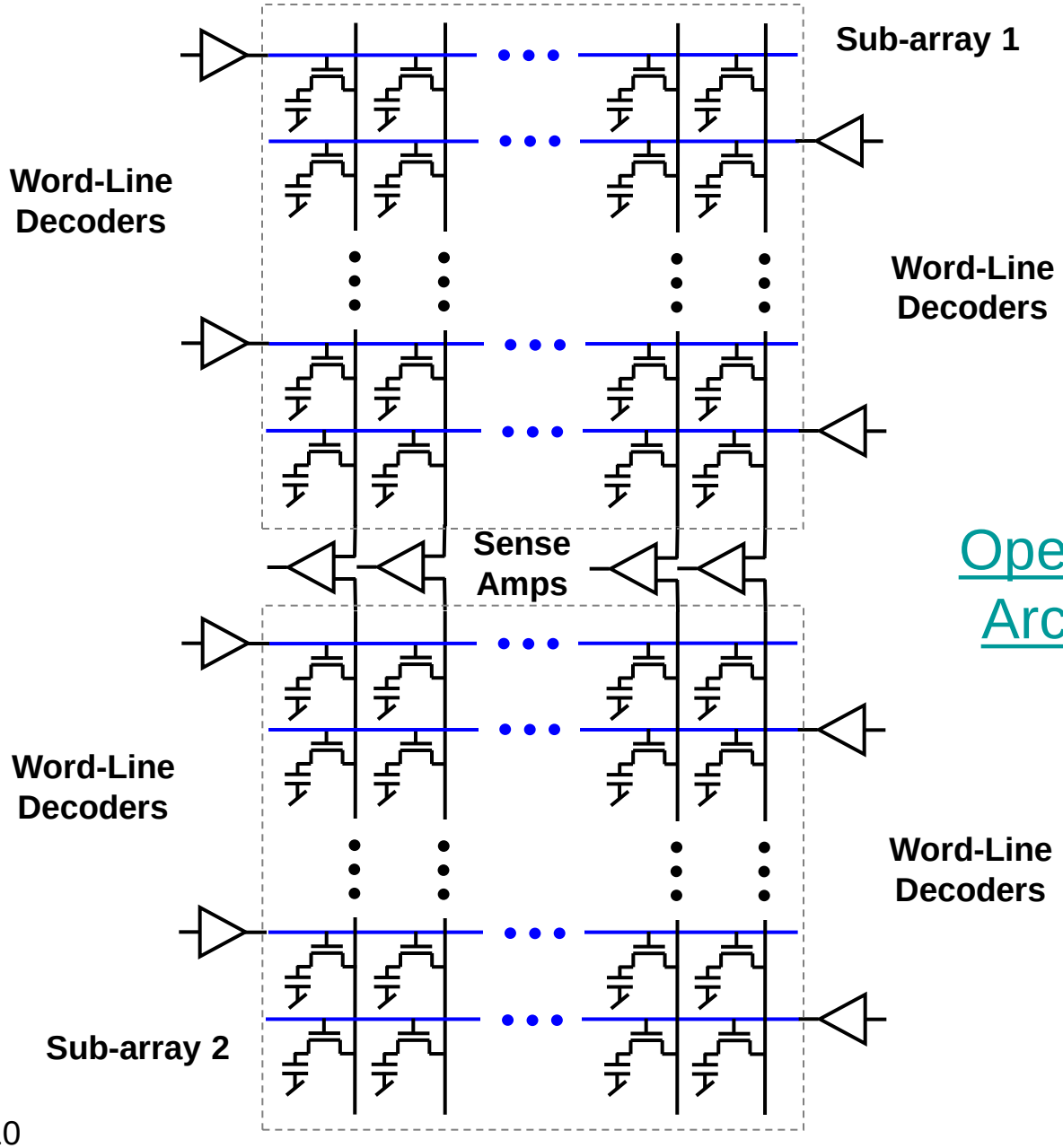


Bit-line capacitance is far larger than cell, hence voltage swing ΔV during read is very small and sense amplifier is used.

Open bit-line architecture Is useful for small DRAMs. It has dense layout but sense amps are exposed to differential noise since their inputs come from different sub-arrays, while word line is asserted in one array.

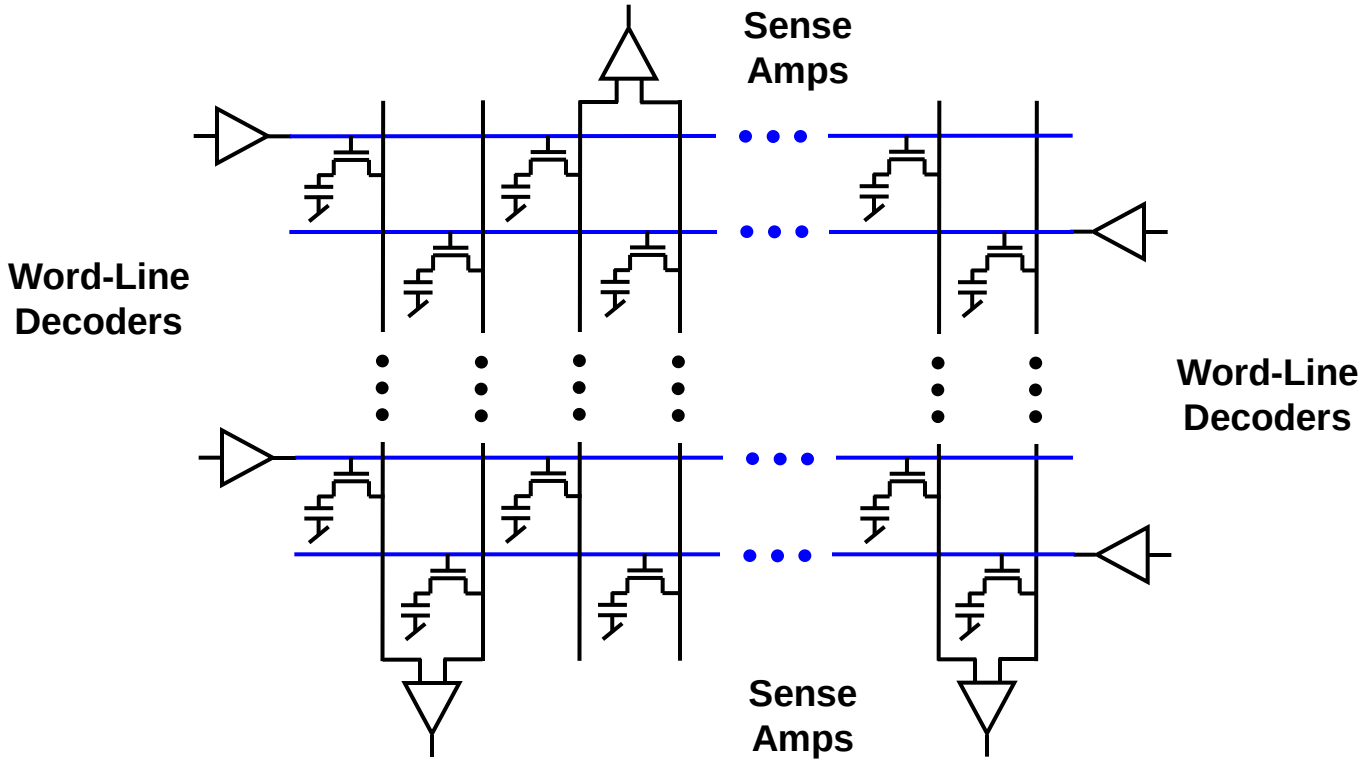
Folded bit-line architecture solves the problem of differential noise on the account of area expansion. Sense amps input are connected to adjacent bit-lines exposed to similar noise sources. When a word-line is asserted, one bit line is being read while its neighbor serves as the quiet reference.

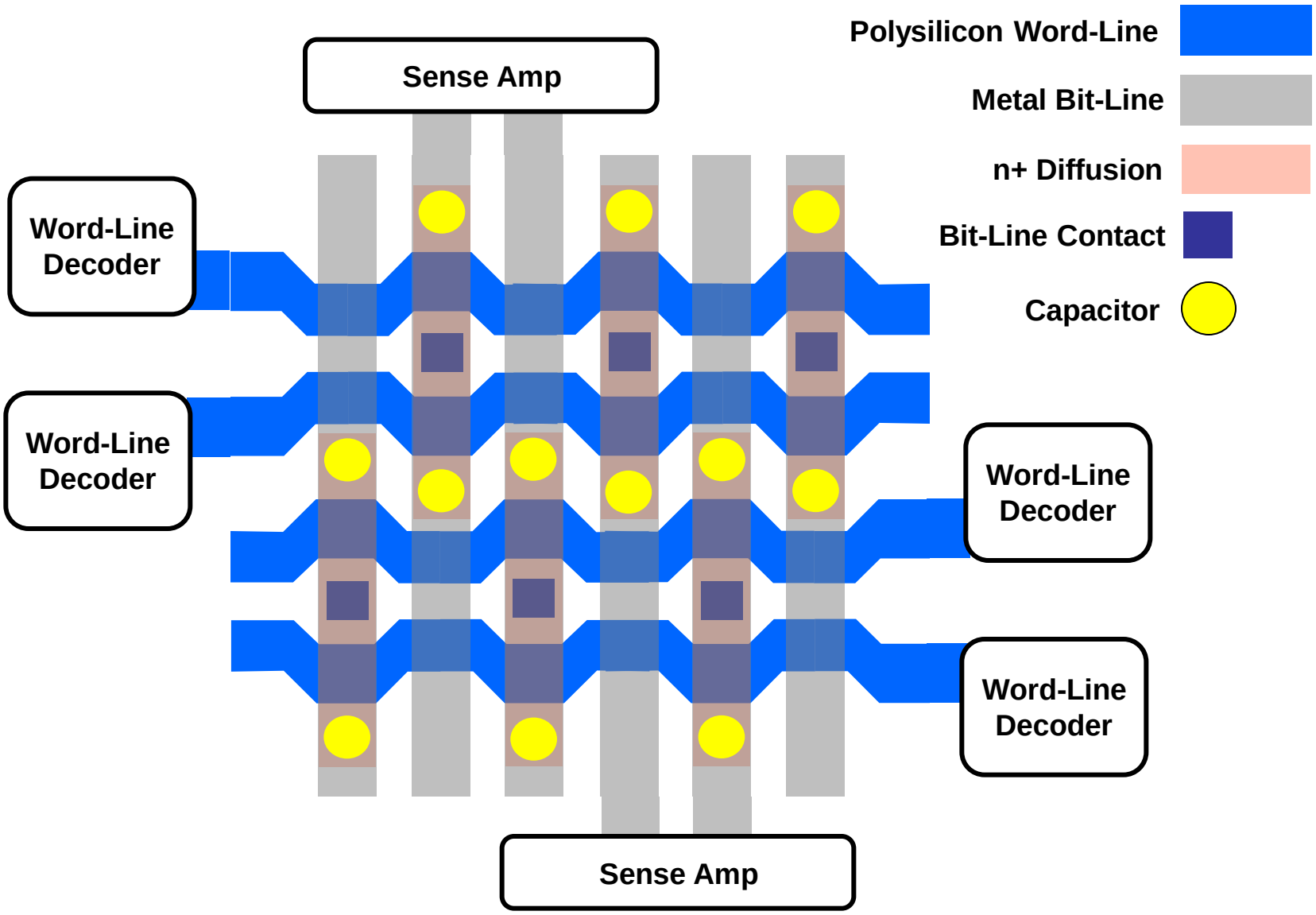
Smart layout and aggressive manufacturing design rules (e.g. 45 degrees polygons) enable effective area increase of only 33%.



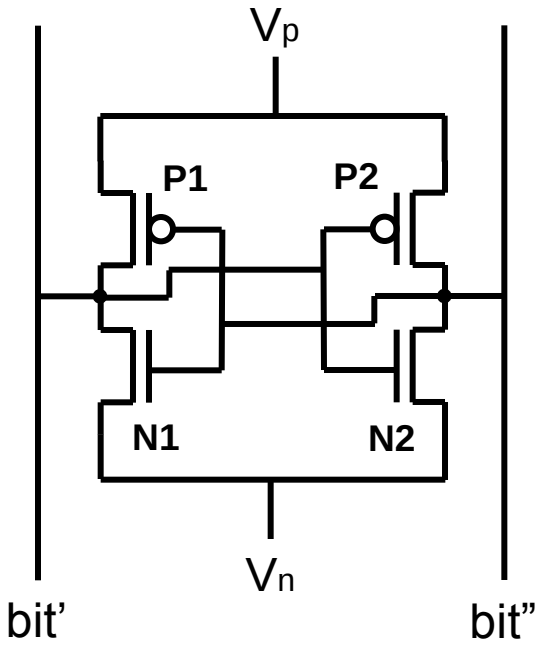
Open Bit-Line Architecture

Folded Bit-Line Architecture




Sense Amp
Word-Line Decoder
Word-Line Decoder
Word-Line Decoder
Word-Line Decoder
Sense Amp
Polysilicon Word-Line
Metal Bit-Line
n+ Diffusion
Bit-Line Contact
Capacitor

DRAM Sense Amp



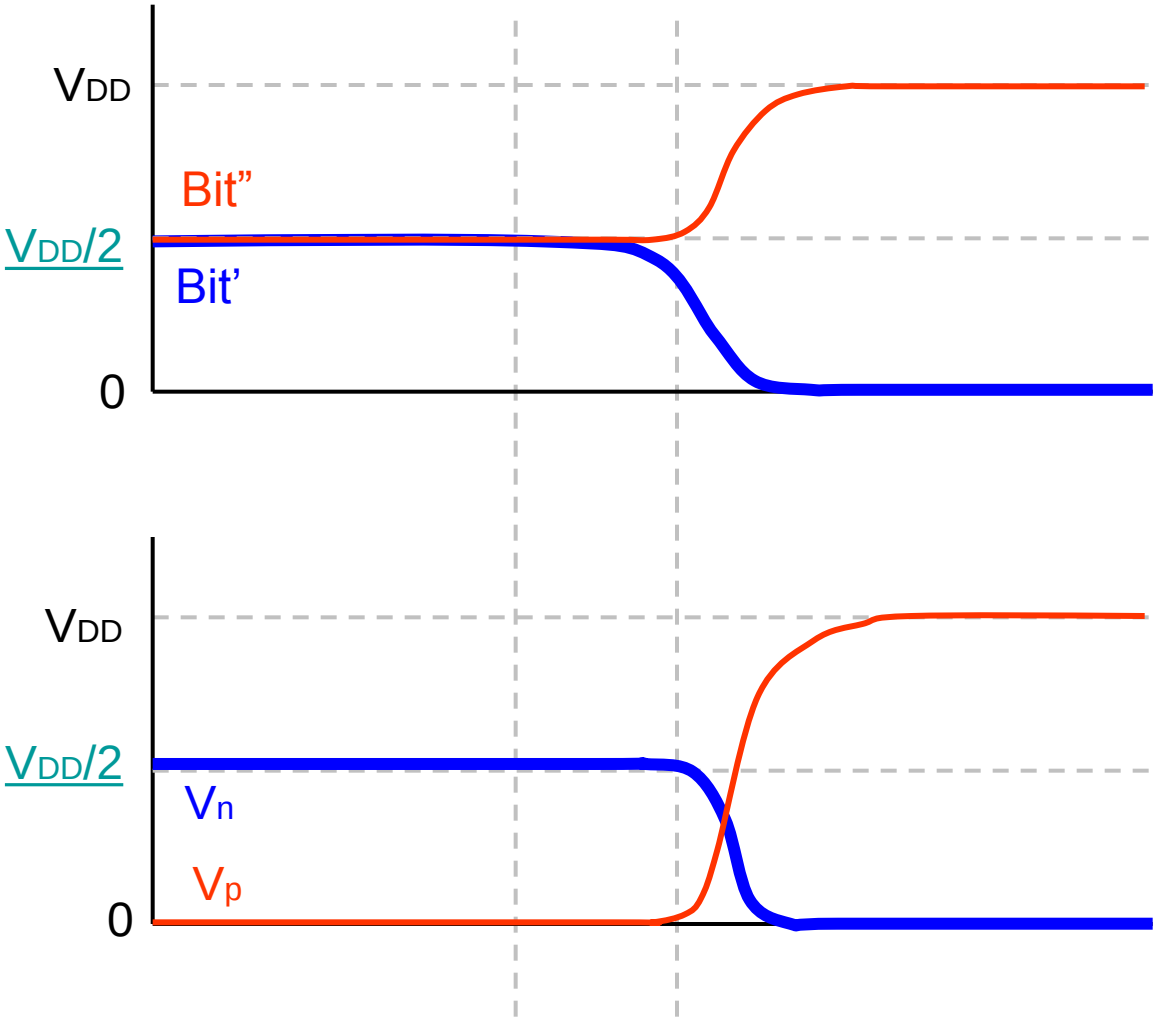
bit' and bit" are initialized to $V_{DD}/2$.

V_p=0 and V_n= $V_{DD}/2$, so all transistors are initially OFF.

During read one bit-line is changing while the other stays float in $V_{DD}/2$.

Let bit' change to 0. Once it reaches $V_{DD}/2-V_t$, N1 conducts and it follows bit'. Hence V_n is pulled down.

Meanwhile bit" is pulled up, which opens P2 and raise V_p to V_{DD}.


VDD
Bit”
VDD/2
Bit’
0
VDD
VDD/2
Vn
Vp
0