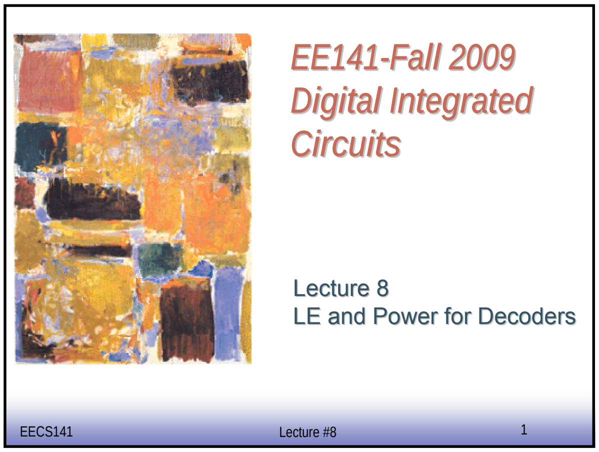
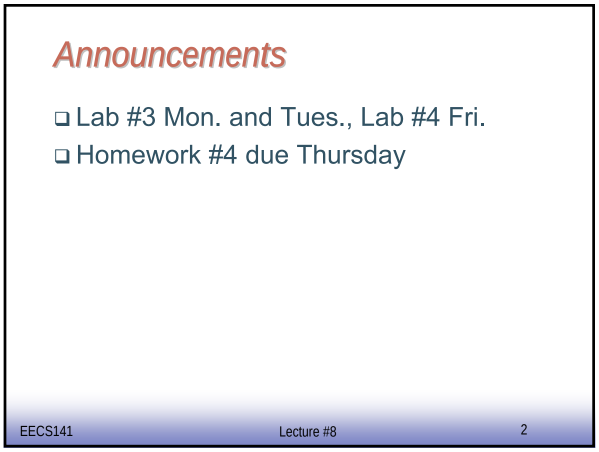


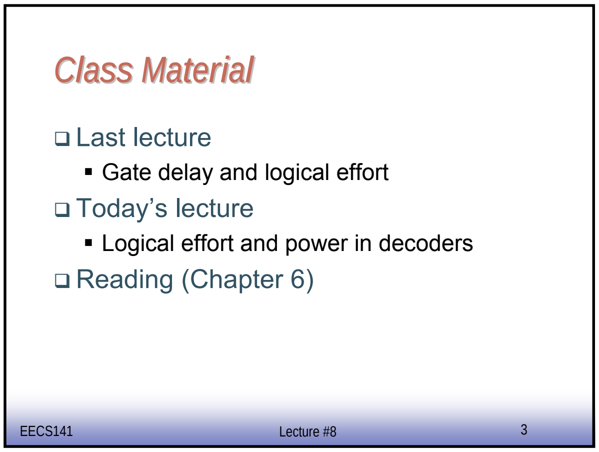
EE141-Fall 2009 Digital Integrated Circuits

Lecture 8
LE and Power for Decoders



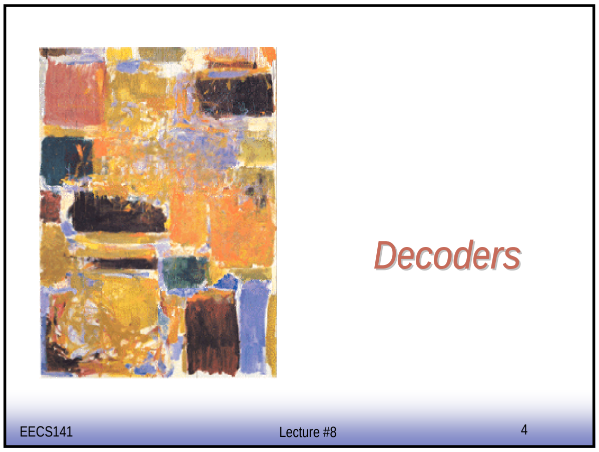
Announcements

- ❑ Lab #3 Mon. and Tues., Lab #4 Fri.
- ❑ Homework #4 due Thursday

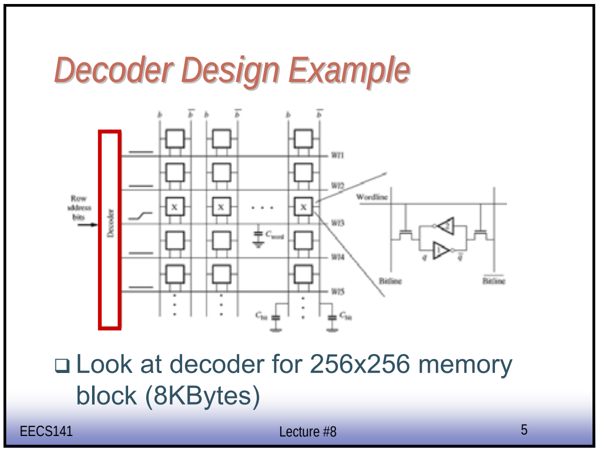


Class Material

- ❑ Last lecture
  - Gate delay and logical effort
- ❑ Today’s lecture
  - Logical effort and power in decoders
- ❑ Reading (Chapter 6)

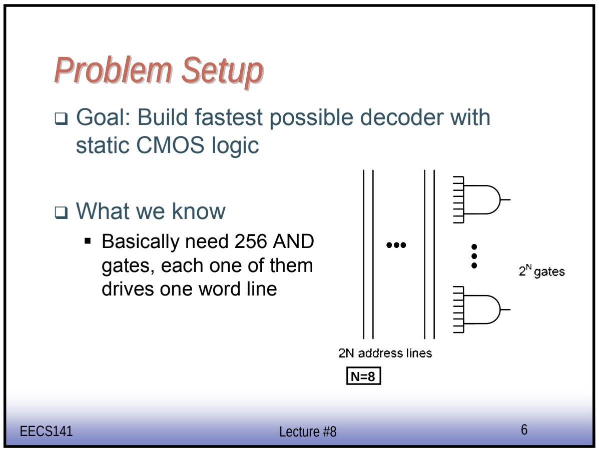


Decoders



Decoder Design Example

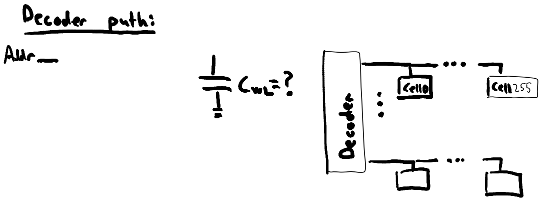
- ❑ Look at decoder for 256x256 memory block (8KBytes)



Problem Setup

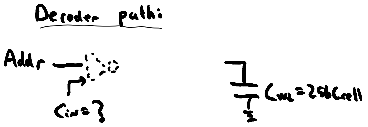
- ❑ Goal: Build fastest possible decoder with static CMOS logic
- ❑ What we know
  - Basically need 256 AND gates, each one of them drives one word line

Problem Setup (1)



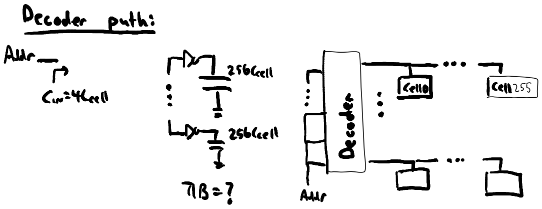
- Each word line has 256 cells connected to it
- C_{WL} = 256*C_{cell} + C_{wire}
  - Ignore wire for now (include it later in the class)

Problem Setup (2)



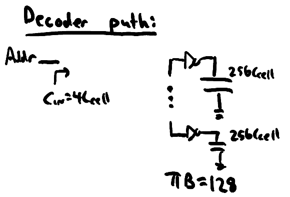
- Assume that decoder input capacitance is $C_{address}$=4*C_{cell}

Problem Setup (3)



- Each address drives 2^8/2 AND gates
  - A0 drives ½ of the gates, A0_b the other ½ of the gates

Problem Setup (4)



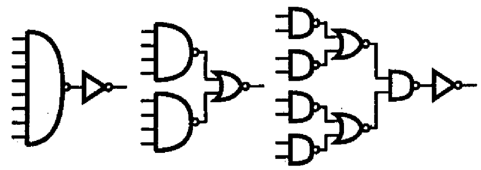
- Total fanout on each address wire is:

$$F = \Pi B \frac{C_{load}}{C_{in}} = 128 \frac{\left(256C_{cell}\right)}{4C_{cell}} = 2^7 \frac{\left(2^8 C_{cell}\right)}{2^2 C_{cell}} = 2^{13}$$

Decoder Fan-Out

- F of 2^{13} means that we will want to use more than $\log_4(2^{13})$ = 6.5 stages to implement the AND8
- Need many stages anyways
  - So what is the best way to implement the AND gate?
  - Will see next that it's the one with the most stages and least complicated gates

8-Input AND

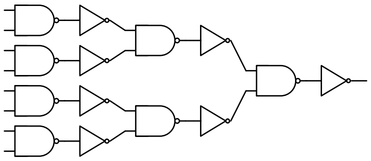


LE=10/3 1
ΠLE = 10/3
P = 8 + 1

LE=2 5/3
ΠLE = 10/3
P = 4 + 2

LE=4/3 5/3 4/3 1
ΠLE = 80/27
P = 2 + 2 + 2 + 1

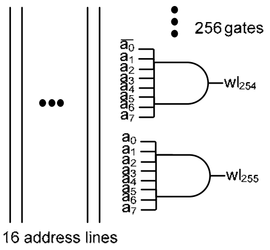
8-Input AND



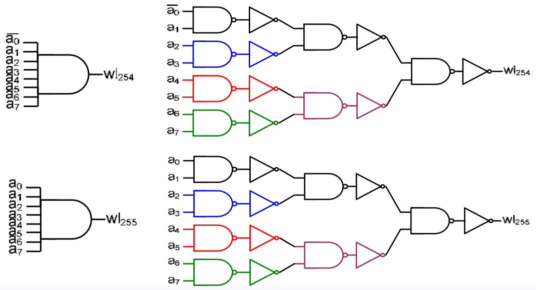
- ❑ Using 2-input NAND gates
  - 8-input gate takes 6 stages
- ❑ Total LE is $(4/3)^3 \approx 2.4$
- ❑ So PE is $2.4*2^{13}$ – optimal N of ~7.1

Decoder So Far

- ❑ 256 8-input AND gates
  - Each built out of tree of NAND gates and inverters
- ❑ Issue:
  - Every address line has to drive 128 gates (and wire) right away
  - Can't build gates small enough - Forces us to add buffers just to drive address inputs



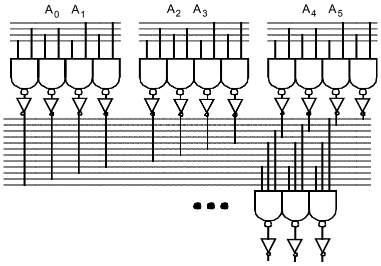
Look Inside Each AND8 Gate



Predecoders

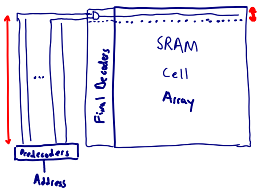
- ❑ Use a single gate for each of the shared terms
  - E.g., from A_0, $\overline{A_0}$, A_1, and $\overline{A_1}$, generate four signals: A_0A_1, $\overline{A_0}A_1$, $A_0\overline{A_1}$, $\overline{A_0}\overline{A_1}$
- ❑ In other words, we are decoding smaller groups of address bits first
  - And using the "predecoded" outputs to do the rest of the decoding

Predecoder and Decoder



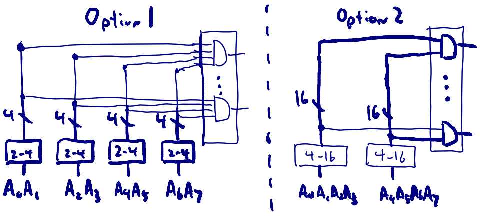
Predecoder/Decoder Layout

- Predecoder outputs run along height of the memory array.
- Decoder must match height of SRAM cell



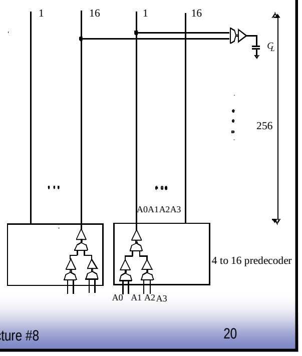
Predecode Options

- ❑ Two options for predecoding:



Predecode Options (2)

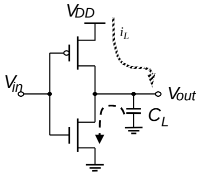
- ❑ Larger predecode usually better:
- ❑ More stages before the long wires
  - Decreases their effect on the circuit
- ❑ Fewer long wires switch
  - Lower power
- ❑ Easier to fit 2-input gate into cell pitch



What We Now Know

- ❑ Given decoder structure, input capacitance, final load
  - Can size the entire chain using LE for minimum delay
- ❑ Is this the “best” we can do in terms of power too?
  - Not necessarily – probably want to reduce sizes
    - – (especially on final decoder inputs)
  - Is there anything else we can do to improve energy even further?

Power



$E_{0\rightarrow 1} = C_L V_{DD}^2$

- ❑ If we lower the supply voltage, energy from switching capacitors drops quadratically!
- ❑ But, how does this impact delay?
  - Need to look more closely at transistor behavior…

Next Lecture

- ❑ MOS transistor model