III. Lecture Notes

LECT 1, 12 SEP

6.978 INTRODUCTION TO VLSI SYSTEMS

Prof. Lynn Conway Off.: 36-595 Ph.: 253-3079

Handouts Today:

(i) Course abstract, goals, outline (preliminary), references
(ii) First half of Homework #1.

Cover Today:

Welcome. New Course. Pleased to have such an outstanding group of students.

> Administrative Details

Registration, a bit about homework and projects, exams, grading. The book we'll use. Materials you should bring to class

> Course Overview:

An overview of integrated system architecture and design, and the topics to be covered in this course.

> Within the Overview: an introduction to MOS technology and design in that technology. It is this portion of todays lecture on which the Homework handed out will be based.

Administrative Details:

> **Registration:** I see some here (--- if so). Please get background questionnaire ---.

Only those whom I've interviewed and approved may register for the course for credit.

With the exception of Faculty members, only those registered for credit may attend. No auditing.

> **Textbook:** A text will be distributed to those taking the course for credit. It is self contained and should be sufficient --- unless you need to strengthen your background in some area or wish to explore the research literature. In either case use the references suggested in the text.

This is a limited printing of a text to be published by next summer. Copies are in very scarce supply. They are not replaceable. Don't lose your copy. The price is feedback - I'd appreciate attention to detail and notification of errors, comments, etc.

> **OFFICE HOURS:** TUES/THURS 3-5.
> WED 10-4 (STARTING NEXT WEEK)

> **BRING TO CLASS:**

(i) COLORED PENCILS, ERASER, MAYBE COLORED FELT TIP PENS.
COLORS: BLACK, RED, GREEN, BLUE, YELLOW.

(ii) GRIDDED SCRATCH PAPER: RECOMMEND WHITE QUAD. PADS 1/4" SQ

YOU'LL NEED THESE FOR NOTES ON SOME MATL & OCCAS. SPOT QUIZZES

> **ASSIGNED READINGS:** Periodically I'll indicate sections in the text you should read & study. Those readings will often augment the lectures in major ways - are a must. I encourage you to read ahead, skim, think about any sections that interest you

HOMEWORK:

> IMPORTANT PART OF COURSE

> SHOULDN'T GENERALLY TAKE TOO LONG. IF IT DOES, YOU'VE PROBABLY OVERLOOKED SOMETHING BASIC.

> SOME PROBLEMS WILL REQUIRE DESIGN & INVENTION. IN SOME CASES ONLY A FEW STUDENTS WILL FIND A SOLUTION TO THESE. DON'T BE DISCOURAGED --- DO WHAT YOU CAN.

> HOMEWORK WILL BE GRADED AND MUST BE HANDED-IN ON TIME. BETTER TO HAND IN PARTIALLY COMPLETED HW THAN TO WAIT.

> WILL BE ASSIGNED WEEKLY. SOMETIMES HANDED OUT IN PARTS (TUE --- THUR). ALWAYS DUE FOLLOWING TUESDAY. IF YOU HAVE A PROBLEM WITH AN ASSIGNMENT, HAND IN ON TIME, THEN SEE ME DURING OFFICE HOURS THAT WEEK TO DISCUSS IT.

PROJECTS: I believe strongly in learning by doing, so:

> IMPORTANT PART OF COURSE. WILL COUNT HEAVILY IN GRADE.

> WILL BE DONE IN STAGES, WITH VARIOUS THINGS TURNED IN AS WE PROCEED, TO MAKE SURE YOU'LL BE ABLE TO FINISH ON TIME.

> INTENT: NOT TOO AMBITIOUS. COMPLETION OF A SIMPLE CORRECT DESIGN MUCH BETTER THAN AN AMBITIOUS PROJECT THAT IS INCOMPLETE OR FULL OF ERRORS.

GRADES: During course a numerical score will be accumulated: The maximum score will be (and breakdown is):

SPOT QUIZES	+ HW (10 x 10)	+ MT	+ FINAL	+ PRELIM PROJ	+ PROJ FINAL REPORT.
(50)	+ 100	+ 100	+ 100	+ 100	+ 100 = 500(+)

I'll RANK STUDENTS ACC. TO THIS SCORE. EST. GRADE BOUNDARIES BY MY JUDGEMENT, AND MOVE A FEW STUDENTS UP/DOWN ACCORDING TO MY JUDGEMENT OF OTHER NON-SCORE FACTORS ---

> QUESTIONS?

Overview:

> This course is about the arch & design of VLSI systems

Integ. System: Informally, system implemented as an integrated whole onto a monolithic material.

Show OM slide Show OM chip

Architecture & Design vs Implementation: Printing Analogy.

> In Particular, A & Des. of large digital systems such as digital computers, spec. pur. syst. for signal processing, arrays of simple processors for performing matrix computations of image processing functions, etc.

> VLSI: why "Very---"? Improvements in Implementation technologies have resulted in Suff. circuit density so that ~ tens of thousands of transistors can be fabricated on a single chip.

It is clear that at least an order of mag linear reduction can still be ~~made before~~. Thus, density increase of ~~x~~100 or more can still be and will be made.

BLACKBOARD

Individual paths ~ 4 to 6 μ wide, ~~separated by~~ 4 to 6 μ. A micron is 1/1000 mm. Chips are ~2 to 6 mm on a side. In future, lines can be reduced to less than 0.5 μ.

Those interested in techniques for high-resolution lithog might check out Prof. Hank Smith's course, 6.969 (Submicron Structures Technology)

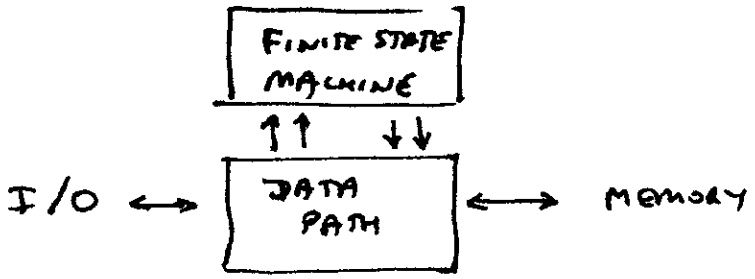
The Design of structures to be implemented on a chip is now an architects game. This trend will increase in the future.

> So, Arch & Des of (Synchr.) Dig. Systems.

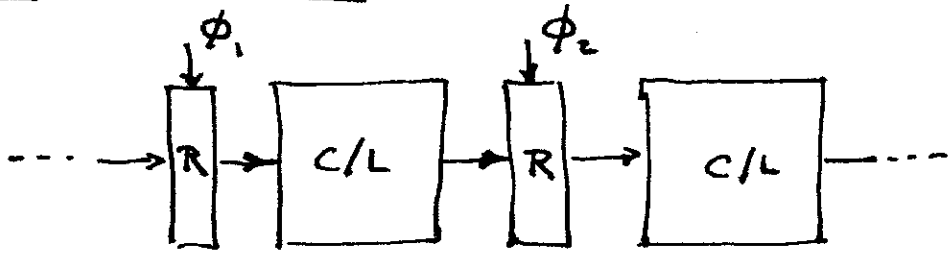
All such are composed of FINITE STATE MACHINES CONTROLLING REG-REG DATA TRANSFER PATHS

Thus we need only 2 basic types of building blocks: REGISTERS & COMBINATIONAL LOGIC (C/L)

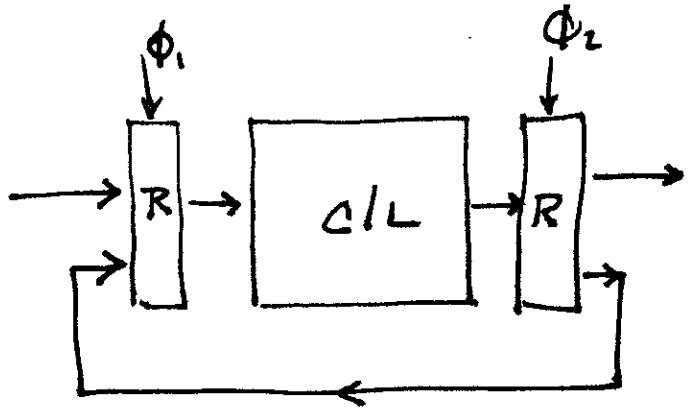
Sketch on Board } EXAMPLE: STORED PROGRAM COMPUTER:



WHERE: DATA PATH IS:



& WHERE FSM IS:



> We will use one very common integrated system technology in this course: nMOS.

The nMOS process fabricates systems which contain a particular type of transistor - the Field Effect Transistor.

Most of you will have studied junction transistors, and perhaps bipolar integrated circuit technology.

MOS-FETs are much simpler in concept, structure, and in their topological properties. You should not try to correlate their properties with junction transistor. Treat them as a new sort of entity.

> REGISTERS, COMBINATIONAL LOGIC, AND their interconnections are very easy to design & implement in nMOS.

> The challenge will be to take advantage of this, to think up architectures, to design large digital structures in a reasonably systematic way, so as to contain their complexity, to describe these structures formally, and to implement them quickly.

> The course will develop a particular design methodology for systematically designing large digital systems in nMOS.

> The methodology will apply rules & constraints to the successive mappings of a design as we proceed top-down from system to subsystem to logic to circuit to stick (topology) diagram to layout levels of design.

> I'll place great emphasis on visualizing & manipulating multiple levels of representation, rather than pushing for expertise at any one level.

NOW, LETS LOOK AT nMOS in MORE DETAIL, BEFORE CONTINUING OVERVIEW. HOMEWORK #1 BASED ON THIS

NMOS 1

> I'd like to develop the basic ideas of how systems are integrated in nMOS technology.

FLIP SCREEN

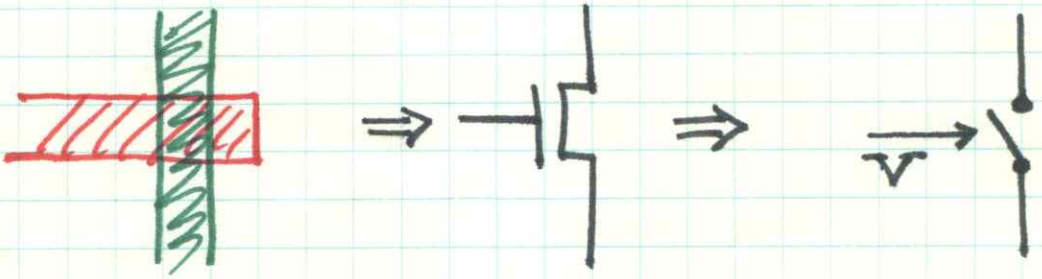
USE WHITE BOARD

VISUALIZE a chip as being a sort of 3-level printed circuit board. 3-levels of conducting material are sandwiched between insulating material. SHOW VG Sequence. Names

By photolithography we can pattern the conducting material & make contact cuts thru the insulating material to form "WIRES" on the various levels. SHOW V6 Sequence

Color codes / Level Names

> Wires on levels may cross with no functional effect, except, where RED crosses Green a transistor is created, which has the properties of a simple switch:



layout circuit symbol "function"

C/L

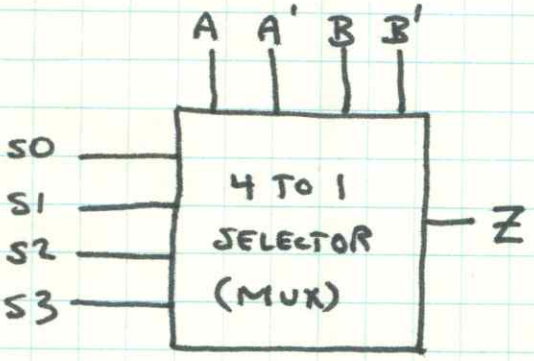
> WHAT COULD WE DO WITH THESE? Lets look at some simple examples of digital functions we can very easily implement using simple groups of switches, wired together in particular ways.

S/L

> WILL USE A COLOR-CODED "STICK DIAGRAM" TO DESCRIBE THESE STRUCTURES.

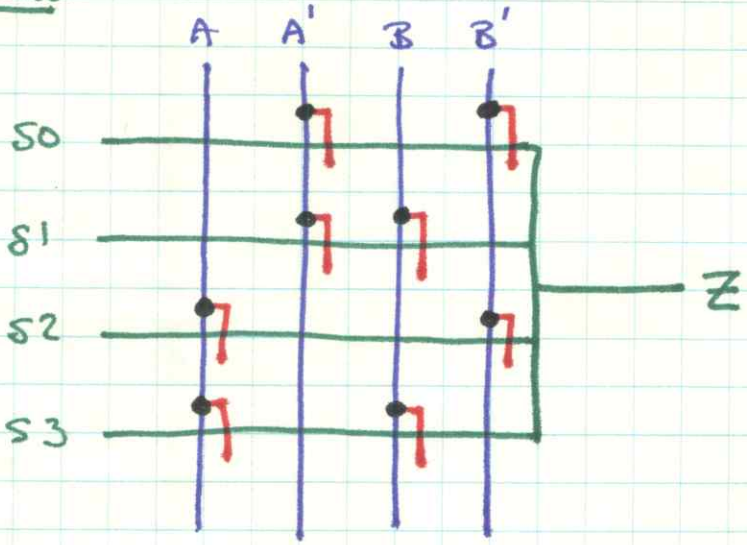
EX nMOS C/L FUNCTIONS: TOPOLOGY

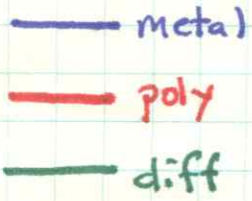
EXAMPLE: CONST. STICK DIAG OF MOS INTEGRATED STRUCTURE TO IMPLEMENT A 4 TO 1 SELECTOR:



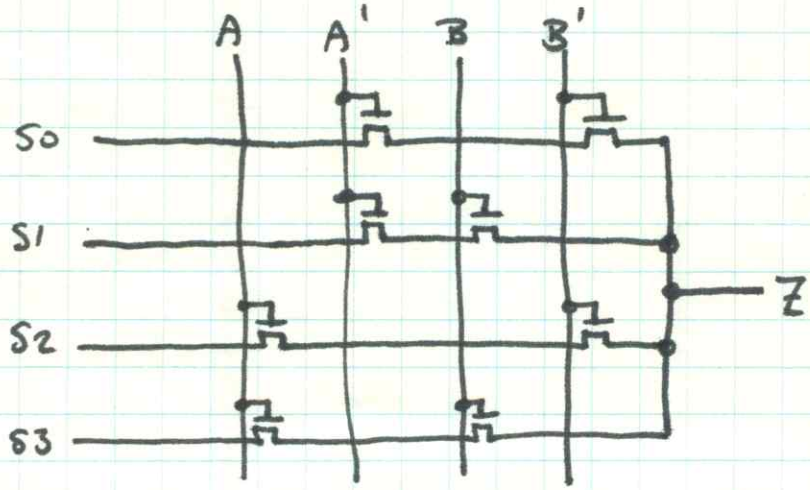
A	B	Z
0	0	S0
0	1	S1
1	0	S2
1	1	S3

SOLUTION:





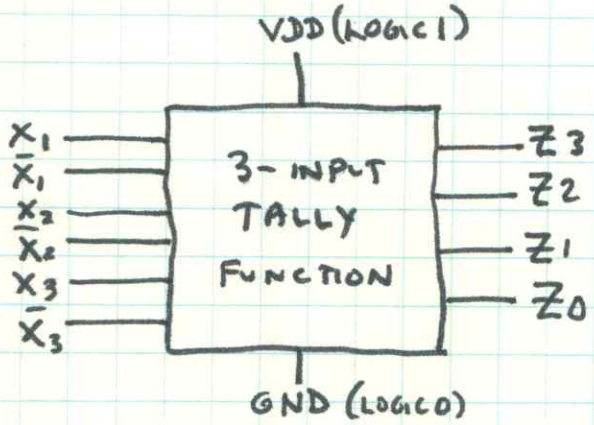
CORRES. TRANSISTOR DIAGRAM: CIRCUIT



> Interestingly, we will often find it easier & more optimal to design in this way (reminiscent of early relay switching logic) rather than use the formal methods of design synthesis using logic gates given by switching fns

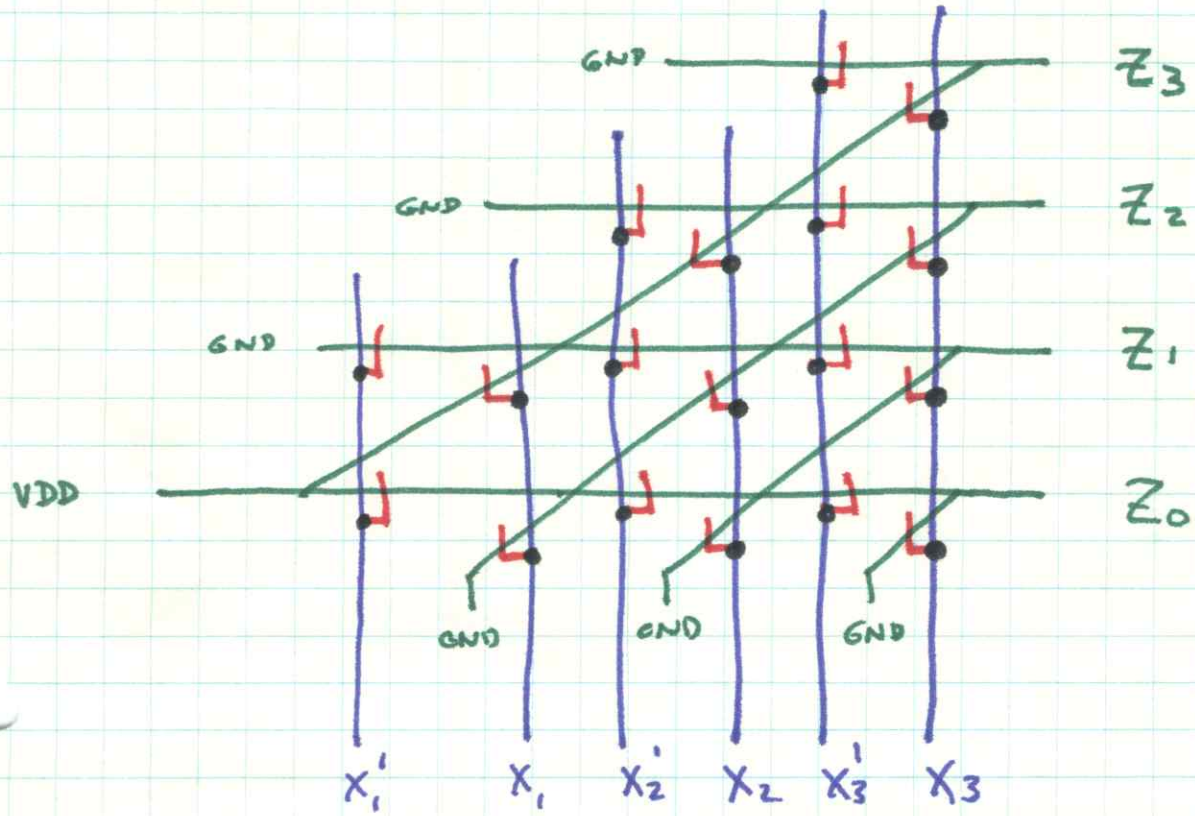
EXAMPLE: CONST. STICK DIAGRAM OF MOS INTEGRATED STRUCTURE TO IMPLEMENT A TALLY FUNCTION OF 3 INPUTS:

N INPUT VARIABLES, N+1 OUTPUTS.
IF M INPUT ARE EQ 1, OUTPUT # M = 1



X_1	X_2	X_3	Z_0	Z_1	Z_2	Z_3
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	0
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	0	1	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

A SOLUTION:



OVERVIEW (CONT.)

> We'll study enough about the electrical properties of the MOS Transistor and simple circuits composed of these, so that we can:

(i) flesh out our stick diagrams to form the geometric layouts of Transistors and wires of correct size to implement working circuits.

(ii) & to be able to predict the performance of circuits in our systems: i.e. signal propogation delays and power consumption.

> We will study enough about the patterning & fabrication technologies used in industry to:

(i) develop a set of Design Rules which place additional constraints on the geometrics of our layouts, i.e. on how narrow line-widths can be [illegible] on how small the line separations can be DES. RULE SLIDE

(ii) develop a symbolic language for formally

Specifying our layout geometries, so our designs can be input to Industrial Pattern Generation machines which make the starting patterns to make masks

(iii) understand the patterning & masking & fab information needed to go along with designs so we can really get designs implemented by commercial firms.

SLIDES ON PG - MASK - CHIP

SHOW ARTIFACTS

> PROJECTS

I believe the best way to learn and master this material is by actually doing it - from top to bottom.

So, we will each do a project. Nothing real ambitious, but enough to be able to visualize every step from architecture to finished chip. More ---

> MULTI-PROJECT CHIP IDEA **SHOW SLIDE**

> FOR A SOURCE OF EXAMPLE SUBSYSTEMS & CIRCUIT LAYOUTS, FOR VISUALIZATION OF A COMPLETE SYSTEM DES. USING THE METHODOLOGY OF THE COURSE, WE'LL ANALYZE THE CALTECH "OM" COMPUTERS IN SOME DETAIL.

> NOW WE'LL BE PREPARED TO LOOK AHEAD. THE DESIGN & PROJECT EXPERIENCE WILL GIVE US THE CONTEXT TO DISCUSS THE FUTURE OF INTEGRATED SYSTEM ARCHITECTURE & TECHNOLOGY, AND TO SEE WHAT THE OPEN QUESTIONS ARE, WHERE THE RESEARCH OPPORTUNITIES ARE:

SOME TOPICS:

(i) SCALING & PHYSICAL LIMITS
(ii) DESIGN METHODOLOGY
(iii) SYSTEMS TO AID/ENHANCE DESIGN
(iv) SYSTEM TIMING & SYNCHRONIZATION
(v) ARCHITECTURE FOR CONCURRENT PROCESSING
(vi) PHYSICS OF COMPUTATIONAL SYSTEMS

HOPING THAT CARVER MEAD WILL BE ABLE TO VISIT FOR A FEW DAYS LATE IN THE COURSE & GIVE SEVERAL LECTURES ON THESE TOPICS.

> QUESTIONS?

LECTURE #2: 14 SEP

WHITEBOARD: EXAMPLES OF LECT 1, FOR THOSE WHO WISH TO TAKE NOTES WHILE BOOK IS HANDED OUT.

HAND OUT BOOK: EACH STUDENT REGISTERED FOR THE COURSE SHOULD GET ONE. YOU SHOULD BE ON THIS LIST. PLEASE SIGN BY YOUR NAME SO I'M SURE YOU GOT YOUR COPY. DON'T LOSE IT

GO OVER BOOK: SINCE LACKS INDEX, ETC., LET ME DESCRIBE HOW ORGANIZED

> ORGANIZED INTO 4 MAJOR SECTIONS. 1,2 / 3,4 / 5,6 / 7,8,9
> WE'LL COVER MOST OF CH 1-4 STUDY EX IN 5-6
> A BIT ABOUT THE BACKGROUND OF BOOK.

> A BIT ABOUT THE BACKGROUND OF THIS NEW FIELD OF ACTIVITY AS A VERY COLLABORATIVE EFFORT OF A NUMBER OF PEOPLE IN MANY COMPANIES & UNIVERSITIES. SO YOU SHOULDN'T ENVISION THIS AS JUST AN ISOLATED NEW COURSE. IT IS PART OF A YET LARGER "SYSTEM", A CONTEXT I HOPE TO TELL YOU MORE ABOUT AS WE PROCEED.

> I ALSO HOPE TO HAVE SOME OF THE OTHERS VISIT HERE. THERE MAY BE AN INFORMAL OCCASIONAL SEMINAR.

> I HOPE TO HAVE PROF. CARVER MEAD VISIT WITH US IN DECEMBER AND GIVE TWO LECTURES ON TOPICS OF CURRENT RESEARCH ACTIVITY: HIGHLY CONCURRENT SYS. & PHYS. OF COMP SYS.

HANDOUT REST OF HW#1. Clarify HW#1.

> FOR EXAMPLE: Clarify function to be implemented in Problem 2(a).*

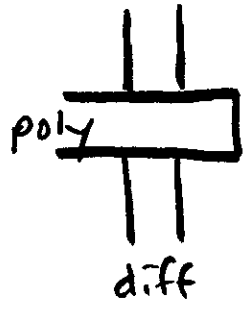
> Mention again: Some problems* will require design, even invention. In such cases only a small fraction of students may get an answer---

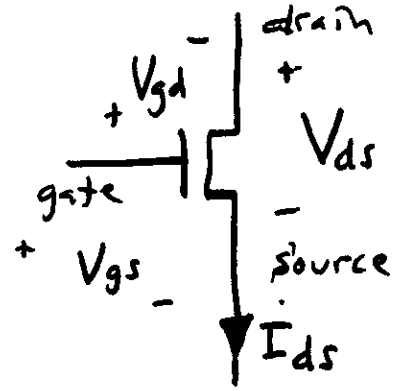
NOW, IN 50 MINUTES: INTRO TO MOS TRANSISTOR AND BASIC LOGIC CIRCUITS COMPOSED OF MOS-FETS

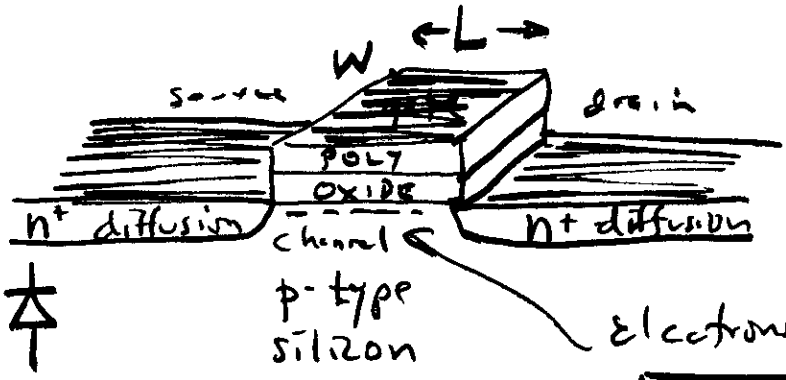
FOUR TOPICS: AND THESE ARE ALSO ASSIGNED READING IN TEXT.

> The MOS Transistor
> The Basic Inverter
> Inverter Delay
> Basic NAND/NOR Logic Gates

MOS TRANSISTOR: Put up ViewGraph

① 

② NAMES/SYMBOLS 

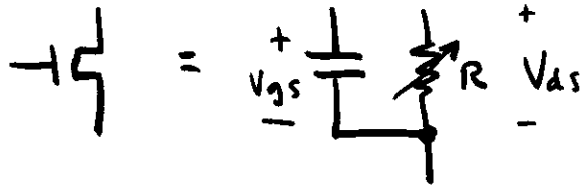
③ 

electron attracted if $V_{gs} \geq V_{th}$

Enhancement Mode if $V_{th} > 0$

- $I_{ds} = Q / \tau$
- $\tau = L / \bar{v}el = L / \mu E$
- for small V_{ds}, $E = \frac{V_{ds}}{L}$ $\therefore$ $\boxed{\tau = \frac{L^2}{\mu V_{ds}}}$
- Transit time is the fundamental time unit of our intrinsic system. All times scaled to τ of smallest FETs in system.

- Neg Q in transit $= -C_g(V_{gs}-V_{tn}) = -\frac{\varepsilon WL}{D}(V_{gs}-V_{tn})$
- $I_{ds} = -I_{sd} = \frac{\mu\varepsilon W}{LD}(V_{gs}-V_{tn})V_{ds}$
- For given $(V_{gs}-V_{tn})$, $I_{ds} \propto V_{ds} \Rightarrow$ 

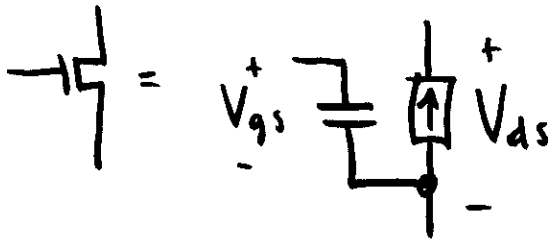
i.e. 

∴ For small V_{ds} & $V_{gs}-V_{tn}>0$ } Resistive Region

$$V_{ds}/I_{ds} = \text{"}R\text{"} = L^2/\mu C_g(V_{gs}-V_{tn})$$

- As V_{ds} increased so $V_{gd} < V_{tn}$, i.e. $V_{ds} \geq V_{gs}-V_{tn}$

Transistor enters saturation region. Further increases in V_{ds} neither increase current significantly nor decrease T

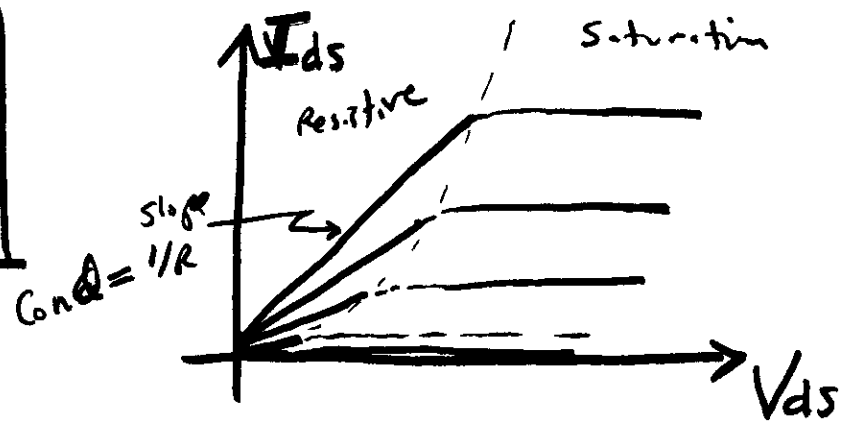


$$I_{ds} \cong \frac{\mu\varepsilon W}{2LD}(V_{gs}-V_{tn})^2$$

} In Saturation, FET behaves like current source controlled by $V_{gs}-V_{tn}$

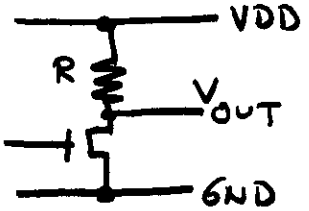
∴ Transit time is $T = L^2/\mu(V_{gs}-V_{tn})$

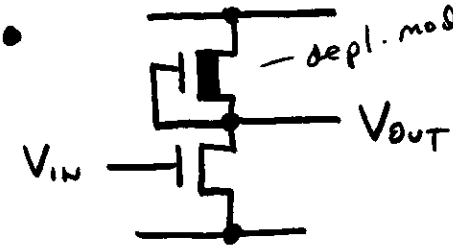
Draw & Discuss Characteristics Summarizing These Regions



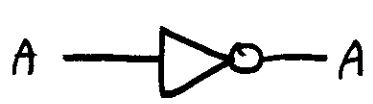
THE BASIC INVERTER: PUT UP SLIDE 2

- FCN: OUTPUT TO BE COMPLEMENT OF INPUT
- WHEN DESC LOG FCN: LOGIC-1 ≡ VOLTAGES ≥ Defined Logic Threshold
  LOGIC-0 ≡ " " < " " "
- If could make resistors, could build inv.:
  explain function
  But, R's especially of high value, long wire.



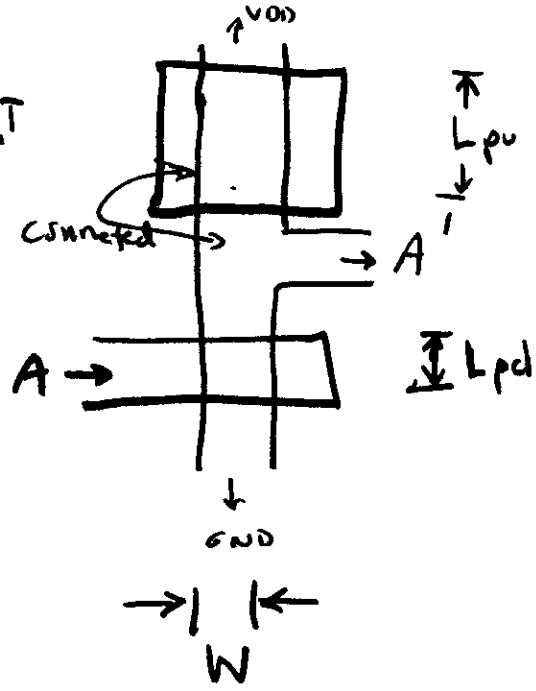
- so, We use a depletion mode MOSFET in place of a resistor
- The depletion mode MOSFET has a threshold voltage Vdep that is less than zero. During Fab ---
- 

depl. mode ⟷ with gate tied to source, $V_{gs}=0$, ∴ ON

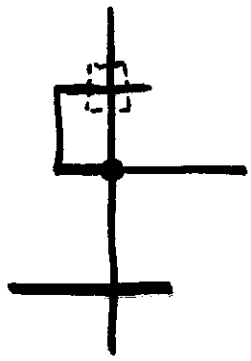


A	A'
0	1
1	0

- LAYOUT



- STICK



- For Reasonable ratios of $\frac{L_{pu}/W_{pu}}{L_{pd}/W_{pd}}$,

INPUT VOLTAGES ≥ A DEFINED LOGIC THRESHOLD VOLTAGE V_{INV} WILL PRODUCE OUTPUT VOLTAGES BELOW THAT LOGIC THRESHOLD VOLTAGE & VICE-VERSA.

- Figs 3a, 3b show characteristics of a typ pair of MOS Transistors used to impl. inverters

The relative locations of saturation differ due to the differences in threshold voltages.

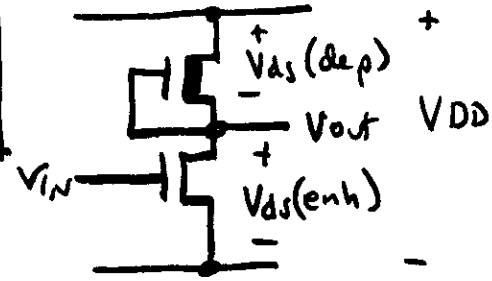
CALCULATE INV. TRANSF. CHAR: i.e. V_{out} vs V_{in}

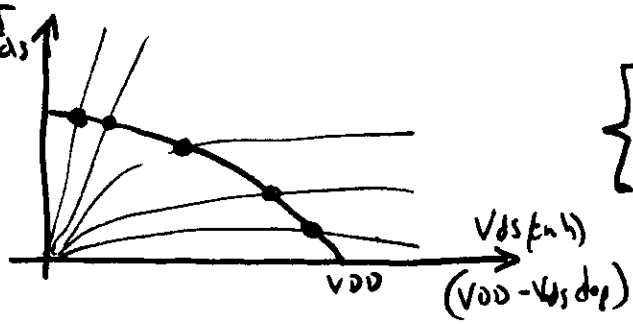
- Rather than back away at solving equations - lets use a graphical construct to determine the transfer characteristics of an inverter composed of 2 such transistors. We usually don't have good analytical expressions for characteristics anyways - just measured curves.

SLIDE

- $V_{ds}(enh) = VDD - V_{ds}(dep)$

Also: $V_{ds}(enh) = V_{out}$



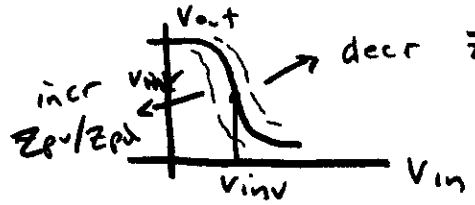
- 

{ only one curve for Depl Mode FET is relevant: That for $V_{gs} = 0$

Superimpose
$I_{ds}(enh)$ vs $V_{ds}(enh)$ (VS)
$I_{ds}(dep)$ vs $(VDD - V_{ds\,dep})$

Since currents equal, intersection of curves yields

$V_{ds}(enh) = V_{out}$ vs $V_{gs}(enh) = V_{in}$

- 

|slope| = Gain

INVERTER LOGIC THRESHOLD VOLTAGE

Logic threshold $\neq V_{tn}$ of the enh. mode FET

V_{INV} is that V_{IN} which yields an equal V_{OUT}

ITS VALUE DEPENDS ON RATIO OF

Z_{pu} to Z_{pd} i.e. $\frac{L_{pu}/W_{pu}}{L_{pd}/W_{pd}}$

(Read in Book) we find $V_{INV} \cong V_{TN} - \frac{V_{dep}}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$

- To maximize $V_{gs}-V_{th}$ and increase pulldown's current driving capability, V_{TN} should be as low as possible. But if too low, inverter outputs won't be driveable below V_{TN}.

Typically $V_{th} \cong 0.2 VDD$

- To maximize pullups current driving capability, might set V_{dep} for negative. However, for given V_{inv} & V_{th}, decreasing V_{dep} requires increasing L_{pu}/W_{pu} → typically requiring more area.

Typically $V_{dep} \cong -0.8 VDD$

- In general, desirable to have equal margin around the inverter threshold i.e. that

$V_{INV} = VDD/2$

- .2 & -.8 choices → $V_{INV} \sim \frac{VDD}{\sqrt{\frac{L_{pu}/W_{pu}}{L_{pd}/W_{pd}}}}$
- ([illegible]) This leads to a pullup/pulldown ratio of

$$\frac{Z_{pu}}{Z_{pd}} = 4:1$$

INVERTER DELAY

Look at the delay thru a sequence of inverters: this is the simplest case for estimating delays.

Define $k = Z_{pu}/Z_{pd}$: Use alt.  symbol.

See figure 4a **SLIDE**

- AT T=0, STEP VDD ONTO INVERTER 1, & LOOK AT WHAT HAPPENS.
- WITHIN ~ τ, pulldown of 1st remove $Q \approx V_{DD} C_g$ from gate of second inverter.
- Pullup of second must supply this charge to C_g of third, but it can supply only about 1/k'th the current of pulldown of 1st.
- So speak of inverter pair delay (one ~~low~~ going / high going transition)

$$\text{inverter pair delay} \cong (1+k)\tau$$

- If one inv drive more than one succeeding (identical) inverter, for example f of them,

  then delay of both up and down transitions is simply increased by a factor f_g for a fanout of f

- These simplified notions of delay are based on a "switchy" model where individual stages spend only a small fraction of their time in the mid-range voltage values near V_{inv}

NAND & NOR LOGIC GATES

- These are constructed as simple expansions of the Basic Inverter Circuit.
- Their behavior, logic threshold voltages, transistor geometry ratios, time delays also direct extensions of the analysis of the inverter **SLIDE**
- Discuss figures 6a thru 6c
- Note that the logic threshold of the NAND is given by

$$V_{thNAND} \sim \frac{VDD}{\sqrt{\frac{L_{pu}/W_{pu}}{n\, L_{pd}/W_{pd}}}}$$

so Z_{pu} must be bigger (L_{pu} longer) than in inverter

- Also, $T_{NAND} \sim n T_{INV}$, and both up/down delays longer.
- So, while NAND is easy to "stick" into circuits, it has very poor area & delay characteristics. Be careful in its use in "real" designs.
- **STICK DIAGRAM NAND & NOR GATES**
- **QUESTIONS?**

LECTURE #3: 19 SEPT:

- Pass Out HW #2(a)
- Hand In HW #1
- How Many Think They found a reasonable solution to Prob. 2(a)?
- Are you interested in seeing a solution? Stick Diag. Dicks Soln. [Put on White Board ?]

Where we are:

Put on Board Before Class

This Week We Move Up a Level: • Discuss Inverter Delays, & A Clocking scheme. Then:

- We'll learn how to make Registers.
- We'll study an example subsystem: A Stack
- We'll learn how to impl. irreg C/L in a regular way using PLA's
- We'll learn how to implement Finite State Machines using registers & PLA's.

Next Week We Move down a Level:

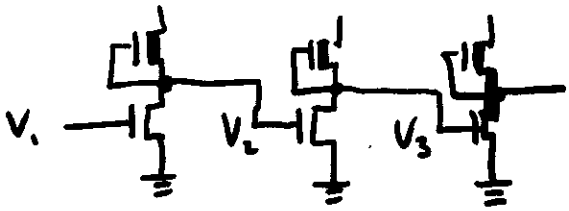
- Study the Silicon-gate nMOS process.
- Based on that, we'll develop a set of Design Rules which constrain how close we place wires, how narrow they may be, etc. (Geometrical Constraints)
- These Design Rules + rules based on the electrical properties of FET's and wires (such as the 4:1 rule) will determine how we may layout our stick diagrams.
- We'll Look at some example layouts

INVERTER DELAY:

[We'll come back to the topic of delay a number of times, treating it in more detail each time]

- Resistive Region: $\tau = L^2/\mu V_{ds}$
- Saturation Region: $\tau = L^2/\mu(V_{gs}-V_{th})$ { larger V_{ds} doesn't reduce τ

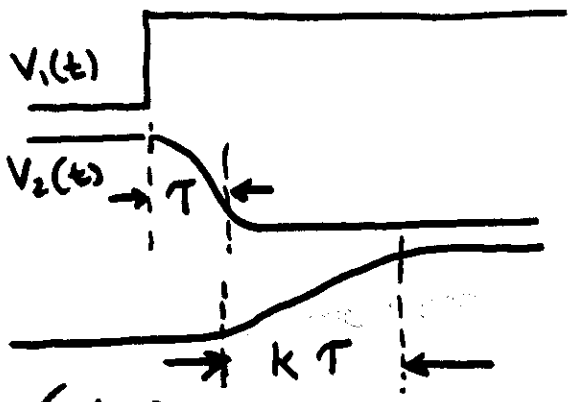
- Examine case where an inverter drives succeeding similar inverters:



Suppose: $V_1 = 0$. Then at $t = 0$,
Drive $V_1 \rightarrow VDD$.

What happens?

Graph of Effect:



- It takes ~ τ (mean) to remove the "positive charge" from the C_g of the second stage - thru the pulldown of the identical first stage.
- When second stage turns off, 3rd stage C_g charges up (thru pullup of 2nd) to VDD.

(τ of Pullup ~ 4 x τ of Pulldown)

- But pullup has less current capacity than pulldown, so this charging up takes longer, by ~ ratio $k = \tau_{pu}/\tau_{pd}$.
- So normally speak of inverter pair delay $= (1+k)\tau$
- If Fanout f (i.e. 1st feeds f next stages in //) or if next C_g is larger by factor f, then multiply switching delay time by factor f.

Some Approx Values in 1978

- Small FET's have gates $6 \times 6\ \mu m$
- Resistances

  Metal $\sim 0.1\ \Omega/\square$, Poly $\sim 15\text{-}100\ \Omega/\square$, Diff $\sim 10\ \Omega/\square$

  Transistors: $10^4\ \Omega/\square$ Note: Res FET's >> Res wires
- Capacitances: (to substrate)

  Metal $0.3 \times 10^{-4}\ pf/\mu m^2$, Poly $\sim 0.4 \times 10^{-4}$, Diff $\sim 0.8 \times 10^{-4}$

  Transistor Gates: $\sim 4 \times 10^{-4}\ pf/\mu m^2$

  Note: C_g only X10 that of wires. But wires typically X10 area of gate they feed. So typically must multiply X2 the gate capacitance to estimate delays. (Call this parasitic capacitance)
- Calculate τ Two ways: (Ballpark, to get order of magnitude)

> $\tau \approx L^2 / \mu \left(\frac{VDD}{2}\right)$ Now good, $\mu = 800\ cm^2/volt\text{-}sec$

$\tau \approx (6 \times 10^{-4})^2 / 800\left(\frac{5}{2}\right) \cong .18 \times 10^{-9} = 0.18\ ns.$

So actual inverter τ (incl. x2 for parasitics) = **0.36 ns**

> $\tau \approx R_{FET} C_g \times 2_{PARASITICS}$

$\tau \approx 10^4 \times 4 \times 10^{-4} \times 10^{-12} \times 2 \times 36_{(nm^2)}$

$\tau \approx 288 \times 10^{-12} \cong$ **0.29 ns**

- Above are actually what we would measure for τ in Ring oscillators for the best current 6 μm processes. Over many processes, typically $0.3 < \tau < 1.0$ ns

NOTATION: You've read about Notation. In partic, MIXED NOTATION

While not formalized, yet will be very useful. Will become clear by example. Useful to

> Reduce clutter in diagrams. Parts of less detailed interest can be left in higher level form.
> Diagram designs when only some of the details have been derived and/or bound.

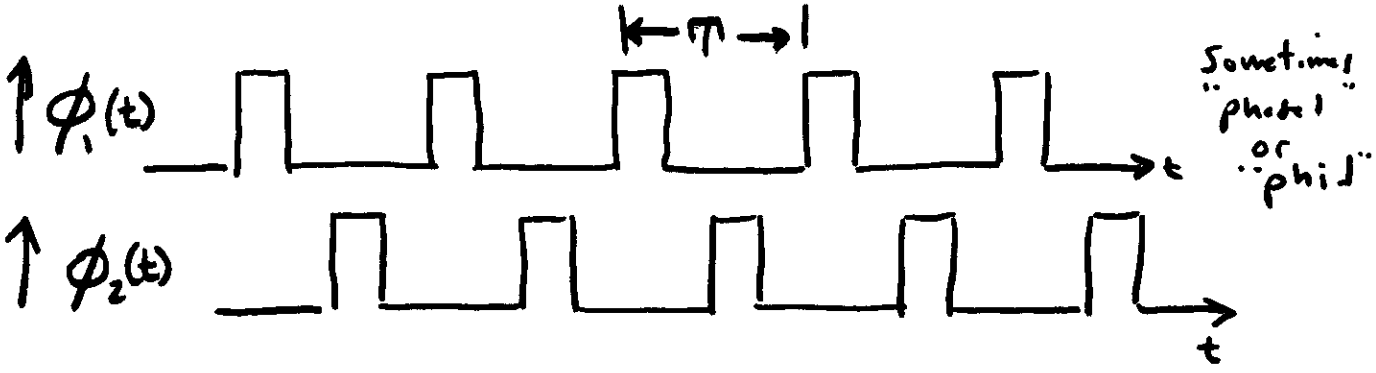
>>> SOMEDAY I easier to visualize FSM with a particular type of mixed notation

TWO PHASE CLOCKS: (START OVER ON BB)

We will use one particular clocking scheme to determine times when we'll allow data to enter and update the contents of registers in our systems:

We call it: Two-Phase, Non-Overlapping clocks

Let's PLOT THE CLOCK SIGNALS AS FCN(t):



> The signals switch between ~0 volts and ~VDD.
> Both have the same period T.
> The high times of both are shorter than their low times
> They are never both high at the same time, i.e. they never overlap.

[What the lock masters must never-never do > open]

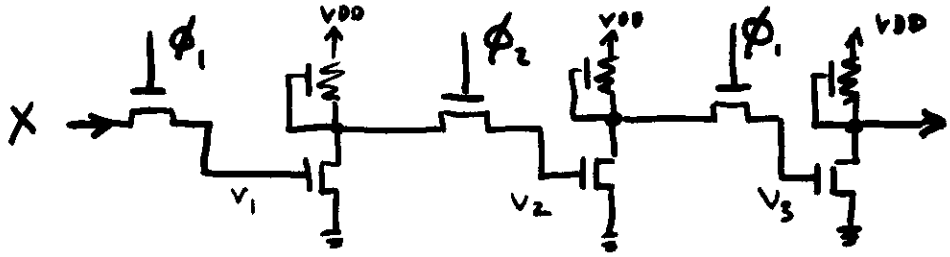
- TIMING/SYNCHRONIZATION are in the general case subtle, complex. We'll come back to this later. For the sort of systems... synch-dig. the 2-φ is perfectly O.K.

THE SHIFT REGISTER:

- Perhaps the most basic structure for moving a sequence of data bits. It is the basic structure from which we will derive our notion of "Registers" and R-R Transfer

- Draw circuit diagram:

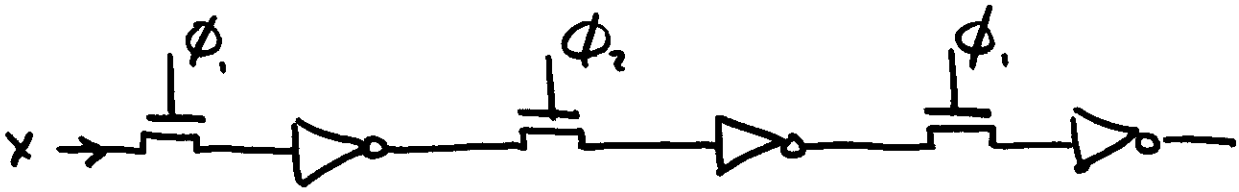
maybe graph V_1, V_2, V_3 vs ϕ_1 ϕ_2



- Describe movement of data during ϕ_1 followed by ϕ_2

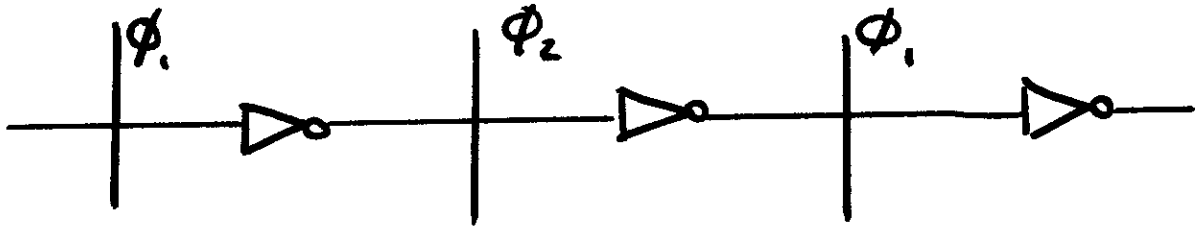
(mention term "pass transistor" or "transmission gate".)
as these transistor "switches" not part of pullup/pulldown static logic. i.e. They lead to capacitive loads only, NOT VDD or GND

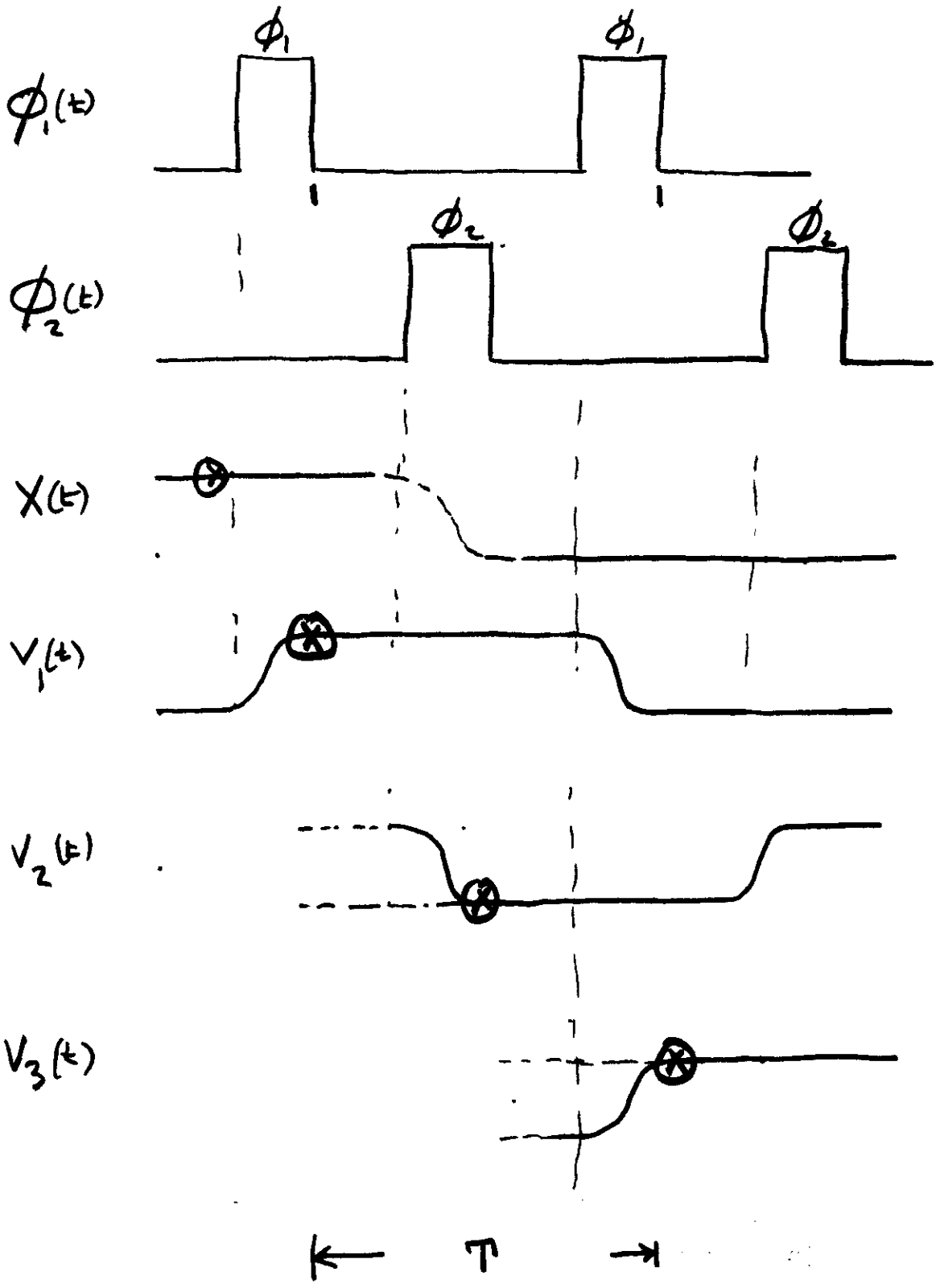
- Now show alternative diagram: mixed notation:



- Describe: especially: must envision the input of the inverter as leading to C_g, the gate of a FET.

- Now start to show part of stick diagram:



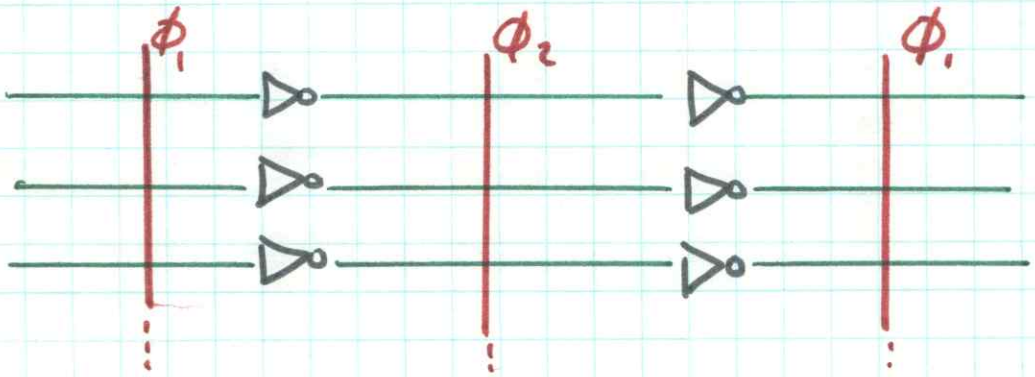


$V_1 \rightarrow V_3$ in $T = "\phi_1" + "\phi_2"$

Continue to Build on Shift Register Idea:

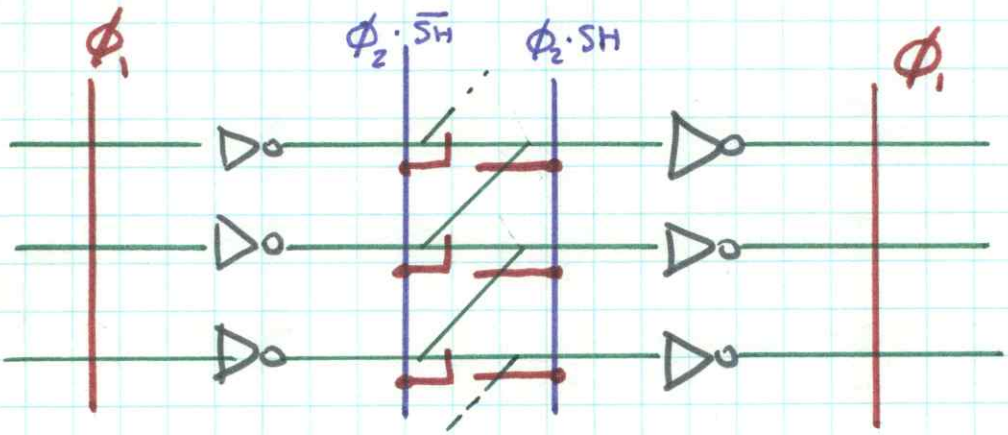
How could we move a sequence of words from register stage to register stage - rather than just a sequence of bits?

- By stacking together several shift registers in parallel, as follows:



- But this is just moving data around. How do we control the data movement: Ah: by putting some switching or C/L function in between register stages:

An example: Shift up / straight thru register stage:



(FLIP OVER TO VG SCREEN)

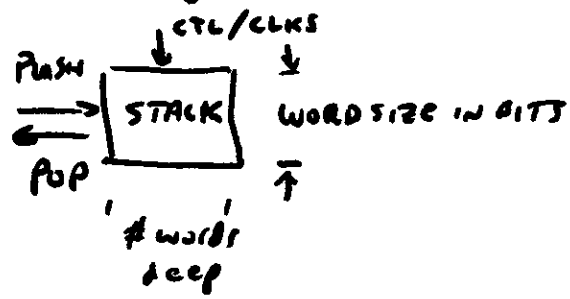
- DIFF LVLS OF ABSTR. SIMULTANEOUSLY PRESENT. HOW TO VISUALIZE FCN OF SUCCESSIVE INV LOGIC STAGES SEP. BY PASS TRANSISTORS?

  SHOW & DISCUSS FIG 6 SLIDE

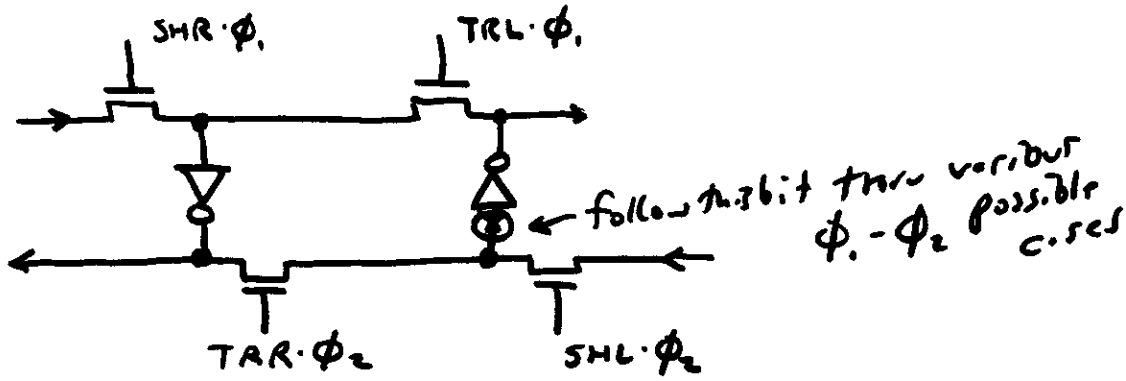
- HOW TO IMPL SIMPLE REGISTERS:

  SHOW & DISCUSS FIG 7-9 SLIDE

- DESIGN OF A STACK SUBSYSTEM (INTRO: TO BE CONT. NXTIME)
- TALK THRU BASIC IDEA:



- First Conceive of a cell design for one bit of one track:



- SHOW SLIDE AND DISCUSS LAYOUT (OR SKETCH)

- SHOW CONTROLLER CHIP SLIDE.

  (WELL SEE HOW TO COMPLETE THE OVERALL SUBSYSTEM DESIGN NEXT TIME, INCL. GEN. THE CTL SIGNALS)

Today: STACK CONTROL, The PLA, Finite State Machines
Next week: The silicon gate nMOS process, Layout design rules, examples of layout from stick diagrams

Lecture #4: Thurs 21 Sept:

- Before we begin today's mtl: talk about References, Projects, Seminars.

  > References: Bring books to class & talk over NEXT TIME

  > Projects: Look at schedule. Talk over.

  Mention looking for 2 people to work hourly - to keep the lab open ~ mid afternoons into evening.

  Mention software packages that must be written to fully support effort: SCAN & PARSE a very limited subset of CIF2.0 (as defined in text), instantiate design file by symbol calls referring to symbol defs, & then plot the resulting boxes on HP plotters. However, it will be several weeks before we absolutely need this software.

  > I'll be handing out another book in ~ 2 weeks which will contain more info, examples of cells & corresponding code, etc.

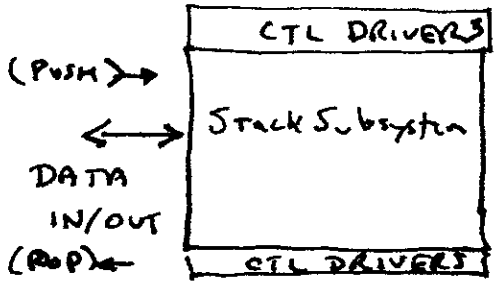
  > SEMINAR SERIES: In addition to Carver Mead lecturing, there are a number of individuals currently actively in research in the area of integrated systems. Some of these people will become very well known as time passes.

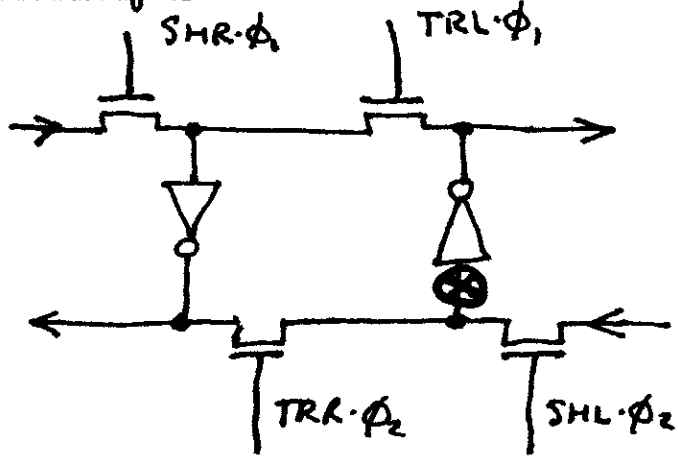
  IF have time

  Mention: Carlo Sequin, Bob Sproull, Chuck Seitz, Doug Fairbairn, Dick Lyon, Wayne Wilner Rich Davies, H.T. Kung, Bob Hon, etc.

- Get a feeling for interest in this series
- Think about times: will hand out questionnaire next week

Continue with Stack Example:





- Note data moves HOR, control lines run vertically.

- Walk thru again: If on ϕ_1, & on ϕ_2, then Fcn:

TRL	,	TRR	→ NOP
SHR	,	TRR	→ PUSH
TRL	,	SHL	→ POP

- **SLIDE: TIMING DIAGRAM:**

> ϕ_1, ϕ_2 always running.

> if NOP most of time, then see that TRL, TRR, occur most of time.

> We need only <u>one</u> control signal (call it "OP") to cause (if <u>on</u> at ϕ_2) the TRR to be followed by $\overline{\text{TRL}}$ and <u>SHR, TRR</u> [PUSH]

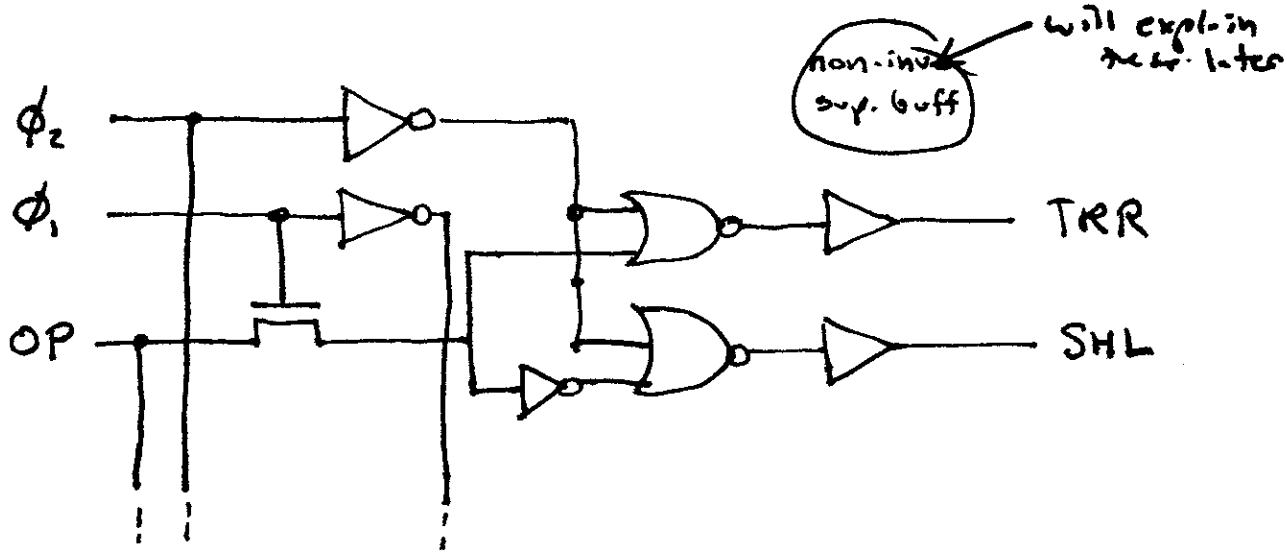
> <u>OR</u> to cause (if <u>on</u> at ϕ_1) the TRL to be followed by $\overline{\text{TRR}}$ and <u>SHL</u> [POP]

Again note: Timing diagrams, even of just sequences of 1's and 0's can be difficult to interpret.

How do we generate TRR, SHL, TRL, SHR ?

From OP, ϕ_1, ϕ_2 to input to drivers which operate the control lines running across the stack:

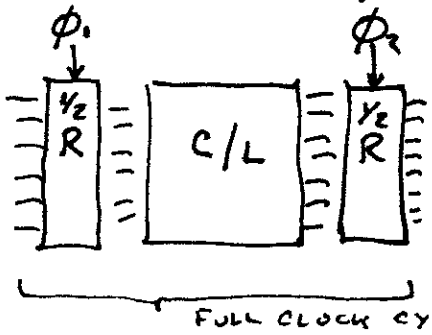
Here's a possible design:



- When ϕ_1 is High, if OP high then drive SHL.
  if OP low then drive TRR.
- When ϕ_2 is high, both TRR & SHL go low, and stay low during the $\phi_2 - \phi_1$ off time.
- note: be careful in these sorts of designs: When ϕ_1 goes off, whichever NOR gate output is high stays high till ϕ_2 comes ON!
- TRL & SHR designed similarly. See Fig 10c. USE FOR ANALYSIS OF OM. BUT, I think it leads to complications
- IMPORTANT POINT: There is a full period between an OP and its next occurrence. Can use to set up counter op with some one line. This can overlap ϕ_1, ϕ_2 ops. Trick used to reduce # micro-code lines.

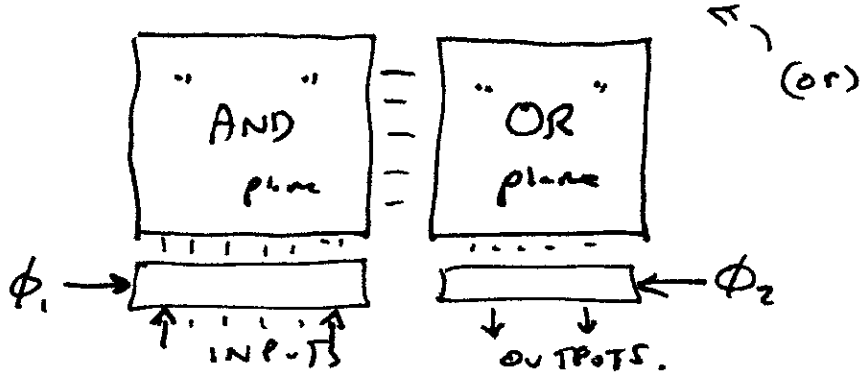
Now, Can make REGs, some forms of C/L. If could only implement arbitrarily irregular C/L in some regular way, we'd have all we need:

The PLA: What we want is C/L to place between Register stages:



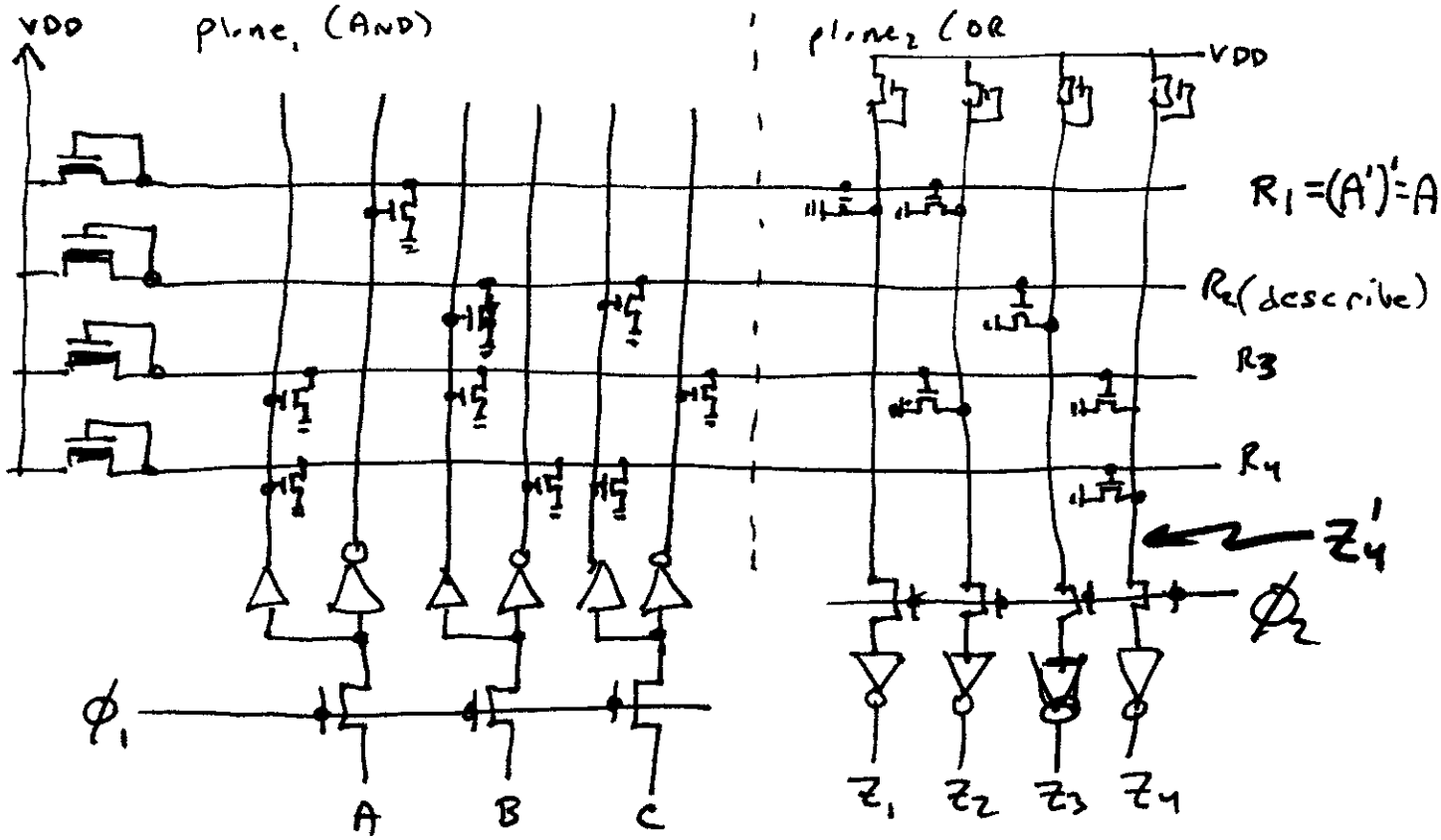
COULD USE A MEMORY, BUT THIS WOULD REQ ALL 2^n poss comb. of inputs × # bits in output. Often wasteful. (mention PROMS)

We impl. the PLA in the following overall subsystem structure:



An Example Circuit: TO ILLUSTRATE FCN OF THE AND & OR Planes. THEY ARE REALLY NOR planes, but as you know NOR-NOR logic (if inputs are avail in TRUE & comp form) can generate all C/L functions of the inputs, just as AND-OR logic can.

EXAMPLE: Note: AM showing the ϕ_1, ϕ_2 registers to indicate how it is <u>imbedded in system</u>. Also to anticipate the Finite State Machine.



<u>AND Plane</u>: If line across plane is high, & FET present, it pulls output low
<u>Notice how the plane, forms the "AND" of the inputs</u>

i.e. (ROW,)

$R_1 = (A')' = A$

$R_2 = (B + C)' = B'C'$

$R_3 = (A + B + C)' = A'B'C$

$R_4 = (A + B' + C)' = A'BC'$

<u>OR Plane</u>: If <u>row is high</u> it <u>pulls vert in. low</u>, <u>so output high</u> <u>(if Transistor)</u>

Notice how the OR plane + Reg row forms the "OR" of input rows

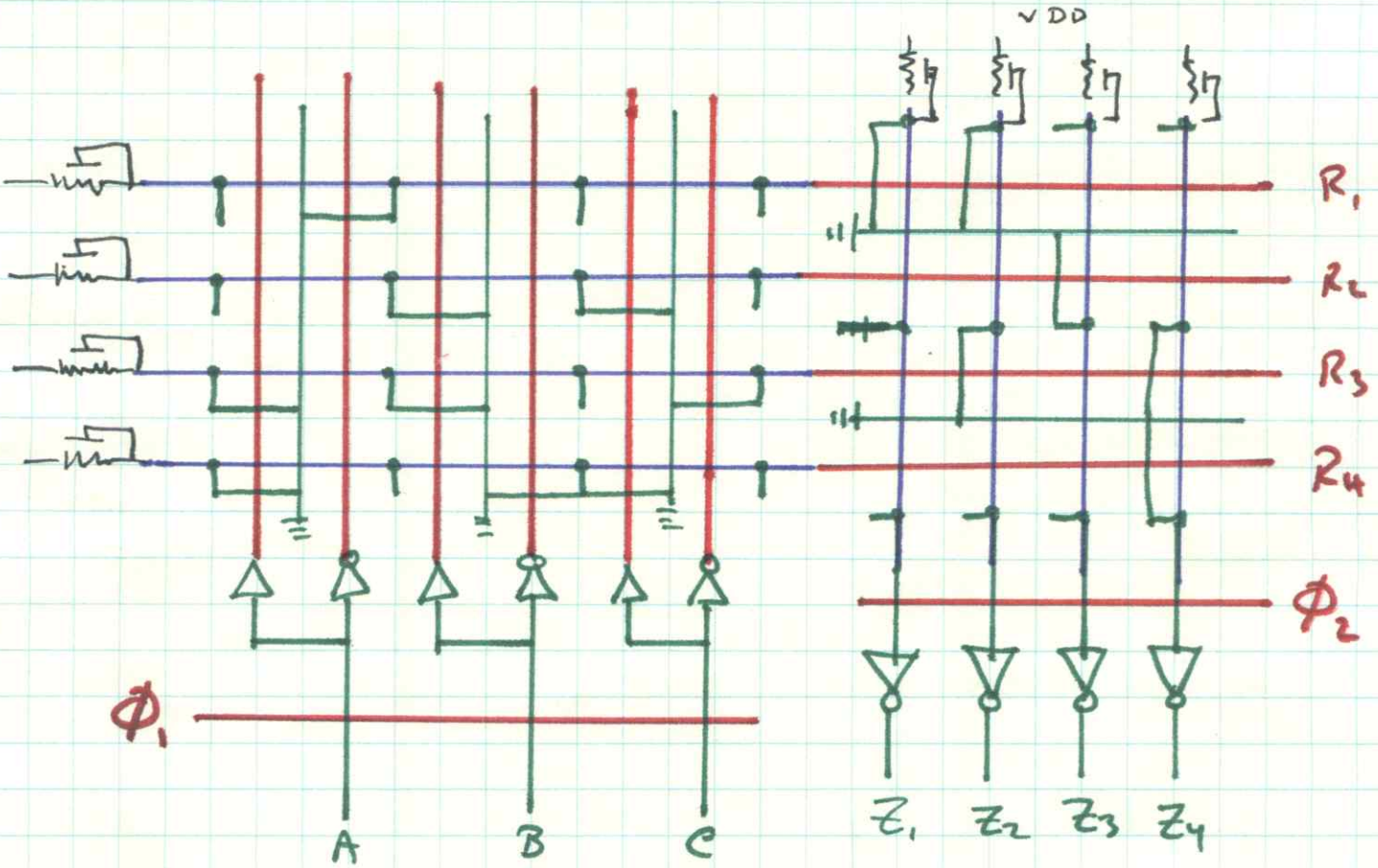
<u>EX:</u> $Z_4' = NOR(R_3, R_4) = (A'B'C + A'BC')'$

Thus: $Z_4 = A'B'C + A'BC'$

[FLIP OVER SHEET/USE WHITEBOARD]

STICK DIAGRAM THE PLA EXAMPLE

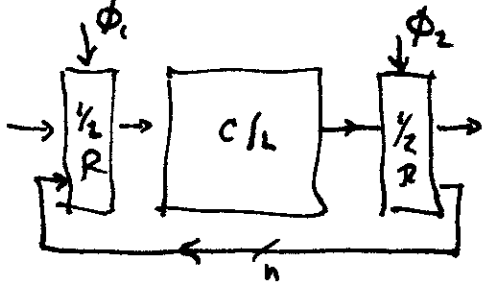
- Run control lines in POLY
- Pullup to output lines in Metal
- Run Ground paths in Diff between alternate poly lines
- Both planes the same, just tilt over an AND plane to get an OR plane.
- Put in input regs/drivers. Pullups.
- Program with transistors at appropriate places.



- The overall size of the PLA is a function of:
  - > # INPUTS, # PRODUCT TERMS, # OUTPUTS,
  - > and the length unit to which we scale our design rules on wire widths/separations (λ)
- Since use NOR form, delays are not too bad.

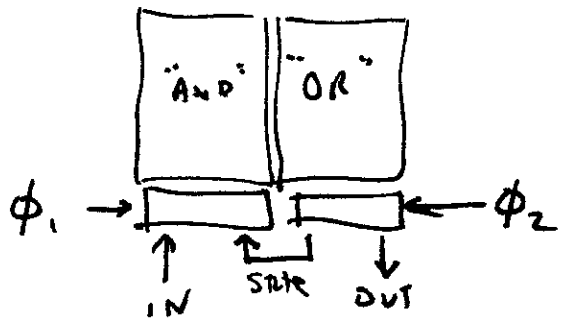
Finite State Machines

- In many cases in the processing of data, it is necessary to know the outcome of the current proc. step, before proceding with the next.
- The results of the present may be used as inputs to the next. They may determine which of several possible next steps we select to do.
- The following configuration can be used to implement a processing step having this regmt:



Some of the outputs are fed back around to the input register.

- This implements a Finite State Machine. The machine has a finite number of states as encoded by feedback paths (# states = 2^n). Now the output and next state are C/L functions not only of the input, but also of the present state.
- We'll usually use the form: PLA finite State Machine



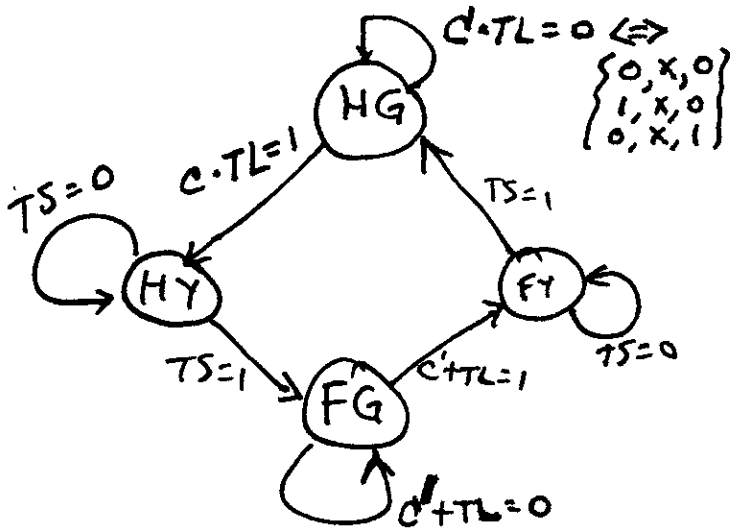
Show part #

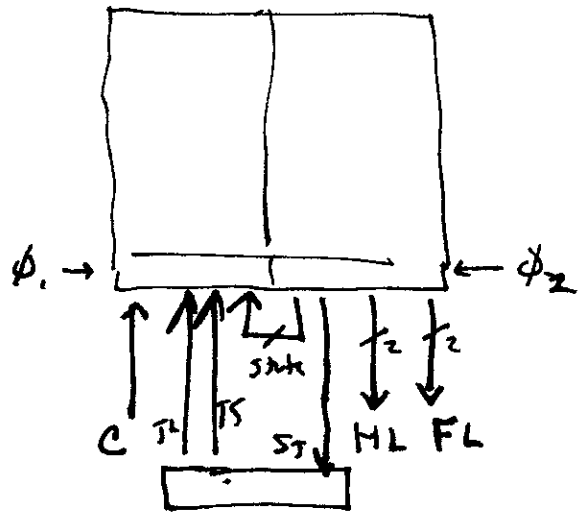
Design a Traffic light controller:

Lets go thru a complete example: **SLIDE**.

- Busy highway intersected by a seldom used farmroad.
- Detectors installed which cause a signal C to go high when cars are at either point (C).
- If no cars are at position C, we wish to control the traffic light so that it remains green.
- If cars are present, want highway lights to cycle thru caution to red, and then farmroad light to green.
- Farmroad light to remain green only while detectors signal cars present, but never longer than some timeout. Farmroad light then cycles thru caution to red, & HW light returns to green.
- Highway light is seen not interruptable again for some fraction of a minute.
- We usually begin by sketching out a **state diagram** consisting of circles and arrows, circles indicating states, arrows indicating poss. transitions, what input causes them, and what output results:

In this case: C, TS, TL





- Describe The Symbolic Transition Table:

An alternative form is shown in (Fig 15d).
This tabular form begins a procedure for mapping the function of the state diagram into a PLA Finite State Machine.

The idea is: In Pres. State IF input is (next state will be) and output will be

(In Pres. State IF input is) — Symbolic Input in ϕ_1

(next state will be and output will be) — Symbolic State Output in ϕ_2

- Describe the encoded transition Table:

We now simply assign binary codes to the states, inputs, outputs to form an encoded transition table. (Fig 15e)

- Now, an algorithm described in Text Ch3 p 24-25 indicates how to construct the stick diagram of a PLA FSM having correct # inputs, state lines, outputs, rows & how to Program / Place the transistors as a function of the encoded transition table entries

You should study this example an convince yourself that the stick diag. impl. the tabulated function.

Do Before Traffic Light Controller

- Mention Problem 8: Draw state diagram, and walk thru possible transitions, clarifying notation.
- Problem 8 is just to implement in a PLA, FSM for already defined problem. We'll do more designs in future assignments where you'll have to create the starting state diagram, given a written description of the problem.

- SIZE: Note: We'll find that the Traffic light controller FSM even in '78, occupies only ~ $1/125^{th}$ of a chip. It can run at a clock rate $\sim 10^7$ times as fast as the real-time prob. requires.

  By late 80's, it might occupy ~ $1/25000^{th}$ of a chip, and run 10^4 times faster.

- Discuss Two Motivational Subjects:

  use of arrays of FSMs, FSM vs "microprocessors"

  > H.T. Kung's array processor algorithms.

  > Image processing right in a display.

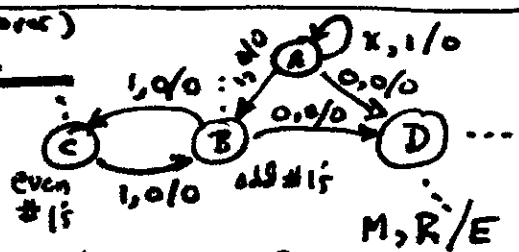
TODAY: • The Si-Gate nMOS Process
• Layout Design Rules

THURS: • Examples of Layouts, From St.2k diagrams.

LECT: SEP 26

LECTURE #5 (TUES)

Admin • Handout Homework #3 ←
• Handout Questionnaire
• Collect Homework #2



• Announce: New Room, St. next time: 38-400

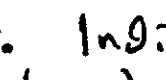
• Explain Questionnaire: Lab sched / poss. Seminars.

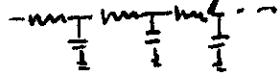
Discuss HW #1:

- Most did very well. If have questions, see me.
- Show best solution to #2(a) - J. D. Brocks, several others had similar solutions. **(ON BOARD)** Interesting thing: Area goes linearly with # inputs. Solution I'd shown before, and those that most got, went as N^2.

Anticip. next week's lectures

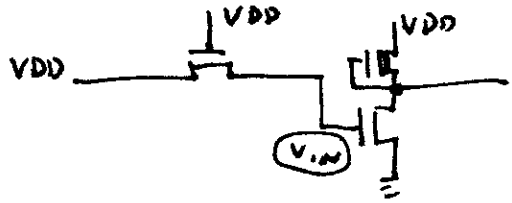
- Show 2 versions of selector. Indicate whats to come in delays in pass transistors i.e. $\propto n^2$. So, not too many of these: They simplify/compact layout, but add delay.





- So, there are real limits to how far we can carry the next "arrays of pass transistors" idea. We'll get more into this next week.

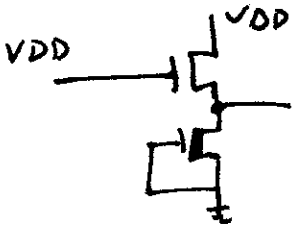
- Note: Already with that if couple inv. logic with pass transistors, then must use 8:1 pullup/pulldown ratio. Why?



We'll calculate this more next week

VDD is highest voltage. If VDD on gate of pass-FET, then V_{IN} can at most be $VDD - V_{tn}$. So, for Vout to go as low with $V_{IN} = VDD - V_{tn}$, as would normal inverter with $V_{IN} = VDD$, must have higher ratio

- bring this up now because: Some people used depl mode p-pulldowns & enh mode pullups. This was o.k. given info you had. But be careful:



Suppose want a non-inverting level restoring stage: If input = VDD, output at most = VDD - Vtn.

If you keep going, each stage would have lower output. As we'll see, such a stage would be very slow also.

- All this points out we must look a little closer at our basic circuits, their delays, etc. (Next week) Before we go into stick diagramming.

3(c): More switches. Some thought it very interesting, some went on to find PLA optimization

Note: There are no best solutions. It all depends on context, and what constraints are imposed by next higher level.

The Silicon-Gate Process: • Overview. Then Intro to

• Patterning: Now we're going to look closer at our PCB technology. At the steps in the process of building up N layers. From this we will develop the ideas on which the layout design rules are based. We'll somewhat abbreviate/simplify the description today — but still cover the essentials. We'll talk a bit more later in the course about the details of the present day process we'll actually use.

- Overview: (Slide from Ch. 4) Talk Thru Slide

Archit. - Des - Sticks - Layout - Des layout then MAKE THE MASTER PATTERNS FOR EACH LAYER (MASKMAKING) then transfer these into each layer at appropriate processing (wafer Fab)

Hold up Artifacts: MASK. WAFER.

- We'll come back to how to describe layouts, and make masks later. We need to develop the geometrical rules for generating layouts before we can have any to describe.

  We deduce these rules from an understanding of the process of patterning (via masks) during processing

- PATTERNING: There will be 5, or 6, (or sometimes more steps) involving transferring a mask pattern into a layer during processing.

  Rather than repeat the details of this each time, Let's go thru it once, and then go on to describe the particular sequence of layers patterned without repeating all the details of the patterning itself.

- So How is a mask pattern transferred into a layer. Probably the classical case is the patterning of SiO_2. This is done several times during the nMOS process.

Talk Thru 2 SLIDES. Maybe use Board.

> Oxide grown (expose bare wafer to O_2 in furnace) on silicon wafer.
> coated with a photo resist material: a particular organic compound.
> 2 kinds of resist: Positive (UV light [ionizing radiation] breaks it down.
Negative (") hardens it.
> Let's use positive example here: Place mask at or near surface, expose
> Wherever radiation passes thru openings in mask, it enters the resist, SiO_2, Si. No effect on SiO_2, Si. But, breaks down resist.
> Develop it in organic solvent which rapidly dissolves broken down resist (but only very slowly. Fat cell attacks other).
> Now we use a selective etchant in this case HF, which dissolves SiO_2 but not resist and not Si.
> Use stronger organic solvent to remove resist.
> Have Transferred the Opaque Mask pattern into SiO_2 (Positive)

(MORE CONCERNED NOW WITH CHARACTERISTICS OF STRUCTURE CREATED RATHER THAN ALL DETAILED STEPS)

- Now: Let use a slide sequence to get a 3-D view of the nMOS process sequence, looking at just the vicinity of one enhancement mode MOS-FET: **GO THRU SEQUENCE.**

> [Note use of Negative resist in the first case shown, i.e. Compl. of Opaque pattern is transferred into the resist.]

> [Note missing ion mask. We'll come back to that later when we look at sequence covering a more complex structure which contain Depl. Mode pullups]

> Emphasize that this is going on every where across wafer at same time. The process steps are Pattern Independent. Its like Developing Film.

- Now: Lets take a look at a single slide which contains a great deal of info: Shows the entire sequence and details of profile of a more complex structure: an Inverter, being built up as successive process steps occur. **TALK THRU SLIDE.**

Be sure to mention:

> thin oxide regrowth before poly is put down (Fig 11)
> note how poly blocks diffusion (Fig 12)
> Contact cuts only go just so deep.
> Metal over cut to poly next to diff: Buttin Contact (more later)

> Mention Sixth mask: Overglass & then cut to Pads

- Mention that most books/articles concerning the process show these profiles. We usually don't need to see these. Ph: and note are not many books that show the other view yet.

LAYOUT DESIGN RULES

Now we've gotten enough of an idea about the process to develop a set of rules which describe permissible layout geometries.

- What are major problems in the process? FOR GIVEN PROCESS:

>> There is some standard deviation in line-widths

$$\sigma = \sqrt{(X - \bar{X})^2}$$

$$\sigma = \sqrt{\frac{\Sigma (x_i - \bar{x})^2}{n}}$$

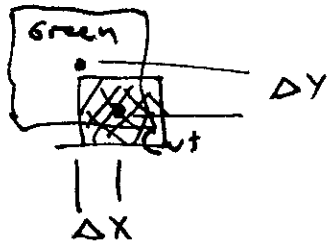
> If make them too small, they'll sometimes disappear!

> If put them too close together, they'll sometimes touch!

>> There is some standard deviation in interlayer registration

(We haven't said how these are re-registered yet. We will later in course. But you can see the problem

For example: a contact:



- These deviations are typically of the same order, i.e. not economically useful to have one << the other. They are being pushed down together.

- Normally, there is known for any given FAB line, a minimum line width / line separation. This is an <u>empirical value</u>: if you try to make lines smaller/closer, you'll get into trouble.

- We define the <u>half-width</u> of the minimum lines as the <u>length unit λ</u>. for the particular process

$$\lambda = \frac{\text{minimum width/line}}{2}$$

- <u>Think of λ</u> as some moderate # multiples of the standard deviations σ_w and σ_t.

- <u>Think of λ</u> as the "resolution of the process"

[We will develop the set of design rules in dimensionless form: i.e. as a set of ratios of permissible geometries to the length unit λ]

- These rules will have some reasonable longevity. They've been "designed" with future scaling effects in mind. At any time on any particular Fab line, some of the rules individually may be weakened leading to denser layouts.

- <u>However</u>: We prefer for prototyping, teaching, communicating ideas, to have a standard simple set of rules which will last a while. Designs done with these rules can simply be scaled down to be run on future fab lines.

- <u>For a product</u>, in highly competitive market, you will likely do another optimization to the detailed rules of the manufacturing Fab line.

The Rules:

SHOW SLIDE: RULES,

- As said, take as given some min. line half-width.

> So Fig 15 min diff line = 2λ
> Fig 23 " poly line = 2λ

> Min Sep in some: min poly sep (Fig 18) = 2λ

> But We usually use min diff sep (Fig 16) = 3λ
since if high voltage of one, if too close, their depletion regions may overlap and current flow between them.

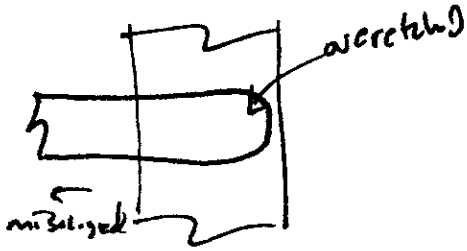
Note: Some processes use a special step to eliminate this problem. — — —

> Now: What about two level separations:

Fig 19: Must keep at least λ bet. poly & diff (uncovered) or will get an unwanted large capacitance when red overlays green

> Fig 20: When Form a transistor (poly over diff)

Must overlap by a least 2λ. Reason, if ever there isn't an overlap, will get a short circuit



An EXAMPLE OF SPECIAL RULE: [If poly width > nλ, then overlap could be ≥ 1.5λ]

Continue with Slide 1

- Fig 21 shows examples: overlap Poly 2λ, poly-diff sep 1λ, diff must be 2λ wide (tricky one)
- To Form Depl. mode FETs: Yellow region must extend 1.5λ beyond gate. & must be 1.5λ from any enhancement mode gates. Fig 22
- Contacts: See Figures 23, 24, 28.
  - > Cuts must be min. linewidth size: min = 2λ × 2λ sqr
  - > Must be surrounded by at least one λ on all sides, to insure contact, and no contact with unwanted layer
  - > Cuts to diffusion must be 2λ from nearest gate to positively insure no shortout of FET
  - > usually use many small cuts to contact large area of diff, which decreases contact resistance.

Fig 27/28

- METAL: Due to Steep Slopes, rough terrain, Metal should be 3λ wd 3λ sep
- BURING CONTACT: If want (except contacts o.h. 1λ) to contact Poly to Green, use Buring contact (describe)

if time ↓

- Buried Contact: There is another way. We sometimes use it, but don't recommend it in general. With another mask step, can cut the thin oxide in selected regions prior to putting down poly, and then special sequence (ask Rb later) can make direct poly-diff contact. Rules are very foundry dependent.
  Advantage: much smaller contact area. Can run poly over metal over top. Makes some layouts much more compact

---

- In principle can contact to POLY over Gate: But, don't do this to minimum size gate especially if have 2λ on all sides. Draw sketch

6.978. Lecture #6. Thurs, Sep 28.

- Collect Questionnaire (avail. to those who didn't get it)
- Handout HW#3 part 2 (also part 1 " " " " ")
- A few HW #1 left, handout.

- Today: (MORE RULES) Examples of layout from stick diagrams. Included will be a few more rules, and some tricks.

> Review Last Time: (Show & Discuss 3 SLIDES:)

The Si-Gate nMOS process, & the design rules.

A Little History

> Before Proceeding, reflect on diff. bet. Si-Gate nMOS & the earlier LSI process: metal-gate PMOS.

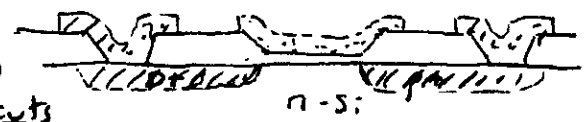
Apart from steadily increasing density as photolith & alignment & proc:

- Metal Gate has only 2 layers of interconnect Metal & Diff. While Si-Gate has 3 layers Metal, Poly, & Diff.

(Actually on ~2 1/2 since always have T where red > green)

- Metal Gate: T formed where thick oxide patterned between a metal over diff crossing? actually over a patterned gap between two diffused regions:

> Diff First placed, > Then thick oxide, (use oxide cuts to control location)



> Then cut to channel, thin oxide, cuts Then metal. [Gate not self-aligned]

- Mobility of Holes is only ≈ 1/2 to 1/3 that of electrons.

- pMOS was a much easier process to get going, for many reasons.

- So layouts in pMOS tended to be more of a hack, due to interconnection difficulties, while as we've seen, (LCD to Poly cell approach) Layouts in nMOS really have interesting topol. properties.

- Metal gate p-MOS suffered from alignment dependent large first order variations in gate-source, gate-drain capacitances due to large, variable overlap.

- Combine with lack of any consistent design methodology, you can see why LSI at first appeared uninteresting to system architects – the attitude was that such circuit and layout (Rubik) should be left to how hires & geeks. ~~However, the game has suddenly changed~~. Getting things to work was a hassle, and often required simulations of details of the circuitry (i.e., consider timing problems in unsystematically designed logic when there are huge variable capacitances hanging on everything).

- However, the game has suddenly changed. Of course most people will unnecessarily carry all the old traditions into the new game. That is fortunate for you, because they will underestimate what you can do, and will not be able to compete. (They'll think you're lucky if layout, when in fact you're mapping algorithms tightly directly in silicon structures.

→ The "Steel" ANALOGY: working a reactor melt to .001"

LAYOUT:

> Constraint on choice of levels:

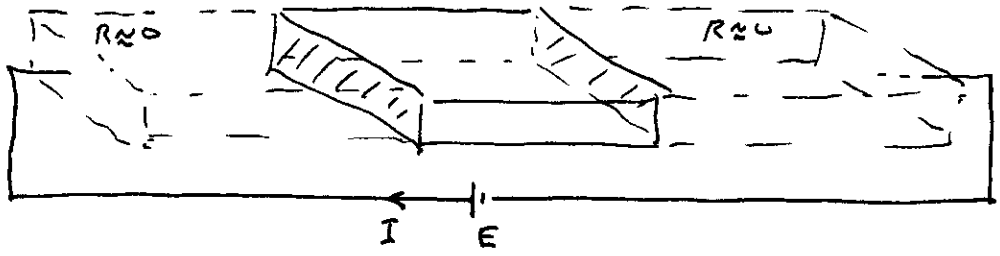
Remember we seldom have to worry about voltage dividing between wires and FETs, since wires much lower (factor of $\sim 10^3$) in resistance. Except be careful of long runs of poly that have to carry much current since poly sometimes may have $R \sim 100\,\Omega/\square$.

Poly ok for clock lines, and we'll see other long run uses later. but not O.K. for routing VDD & GND (except for short crossovers)

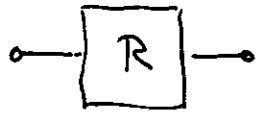
Use Metal & DIFF for routing VDD & GND.

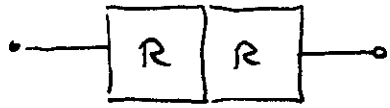
Speaking of resistances:

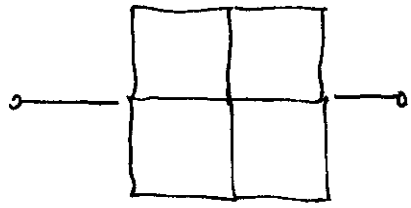
> The way I think about sheet resistivity:



Place □ between really good conductors. Measure $R = \frac{E}{I}$

So have: 

Now if did this:  $R' = 2R$

& if did this:  $R'' = R$

So, for given thickness, Resistance is same for □ of any size.

Thus we quote the "sheet resistivity in Ω/□ (ohms per square)

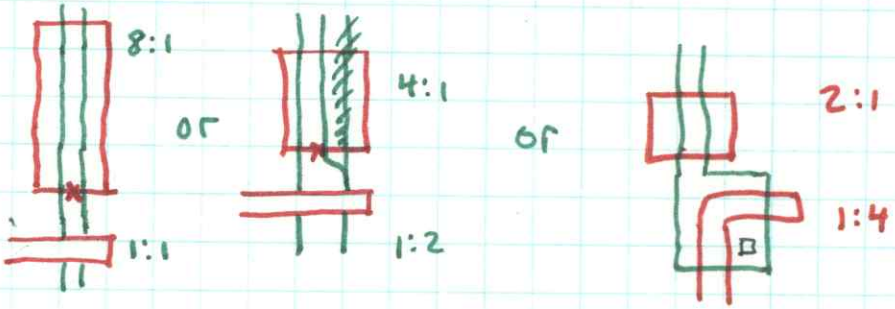
& Calculating the resistance of a layout is fairly easy: we just compute its effective L/W ratio and multiply by the sheet resistivity.

 IN DIFF, at 10Ω/□, This would have R ~ 100Ω

R_M	≅	0.1 Ω/□
R_p	~	15 - 100 Ω/□
R_d	≅	10 Ω/□
R_g	~	10^4 Ω/□

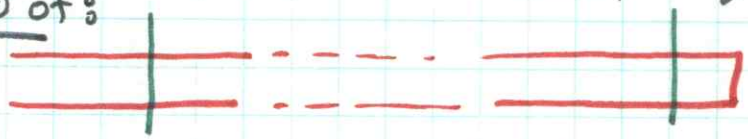
LAYOUT IDEAS: • Variety of ways to obtain any given Pullup / Pulldown Ratio:

Consider: 8:1

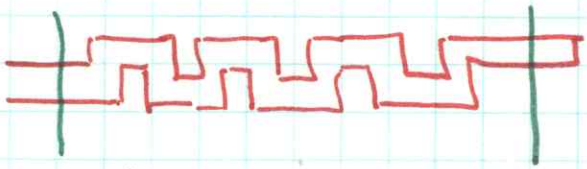


- How would you make a really big pulldown?

Instead of:

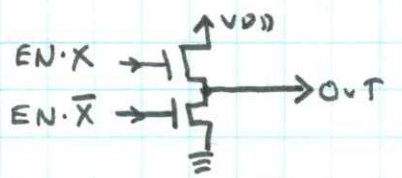


You could use a caterpillar:



(FOR GIVEN RATIO)

- CHOICE OF whether to use long Pullup / ~~narrow~~ pulldown (or) shorter pullup / wider pulldown depends on large variety of factors: (space constraints may dictate a choice)
  - > wider pulldown version consumes more power
  - > wider pulldown version drives later stage faster
  - > wider pulldown version is slower to be driven by small preceding-stage. (more next week on calculating some of this)
- EXAMPLE: IN OUTPUT DRIVER, MAY WANT REALLY BIG T's TO DRIVE OFF-CHIP CAP. LOAD: CONSIDER "TRI-STATE DRIVER":



SHOW SLIDE

(we'll use something like this only smaller in layout)

> SPEAKING OF POWER : ANOTHER LAYOUT CONSTRAINT

- We'll usually run VDD & GND to subsystems in METAL. There is a limit to how much current/unit cross section that metal can carry

- Metal migration: a phen. (not well understood) where if a current density threshold is exceeded, the metal atoms start physically moving in direction of current. If small constriction: current higher there, metal moves faster there, next thing you know it blows like a fuse.

- For Aluminum: This limit is a few times 10^5 a/cm^2 i.e. a few milliampers/μm^2

Let's use a limit in this course of 1ma/μm^2. Note the scaling we will predict shows Vertical dimension & all voltages scaling directly with the horizontal scaling. But pwr density will remain same. Thus current densities will increase.

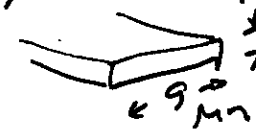
If you want your syst. design to last a while, you might use an even more conservative value, say 0.5ma/μm^2

Unless someone finds a good way to fab high aspect ratio wires 9μm 1μm

---

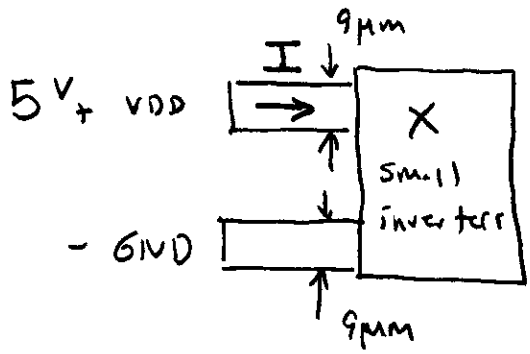
> Now what does this mean in practical terms in today's layouts

Metal Wires are 3λ wide = 9μm, and ~ 1μm thick



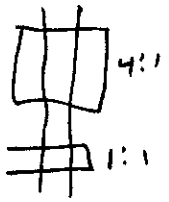
Consider the Question: How many minimum size 4:1 inverters could power lines of minimum size support?

i.e.:



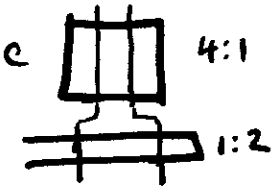
what is X when I/area = 0.5 mA ?

For 4:1 inverters: (min size) (pulldown)



ON Resistance = $(4+1)\ 10^4\ \Omega = 5\times10^4\ \Omega$

$\therefore I = \frac{5}{5\times10^4} = 10^{-4}\,a = 0.1\,ma$

- So a wire 9μ wide ~~could~~ has cross section = $9\,\mu m^2$, and can supply $9 \times 0.5\ ma = 4.5\ ma$ so **X ~ 50**
- But usually half are on, half are off so **X ~ 100**
- And if they are 8:1 rather than 4:1: **X ~ 200**
- But if make 8:1's like 

then **X ~ 100**

> WHAT THIS MEANS IS THAT MIN SIZE METAL LINES WILL SUPPLY A MODEST SIZE SUBSYSTEM, BUT NO MORE.

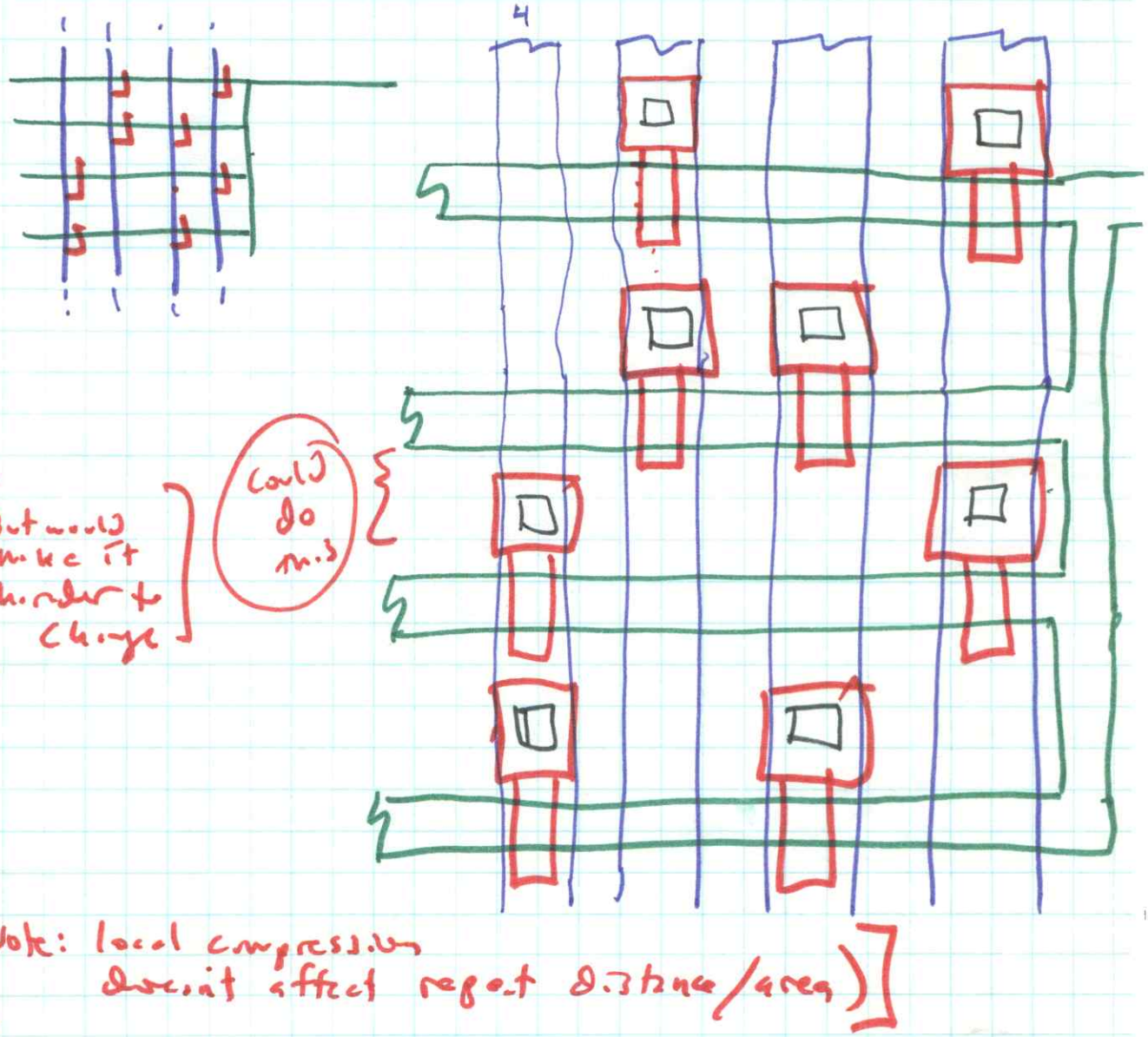
As we move up to larger subsystems, and groups of them, we at some point must start calculating the current requirements and widening the power mains appropriately:

SLIDE: FRONTISPIECE

SLIDE: CHAP 5 FIG 24

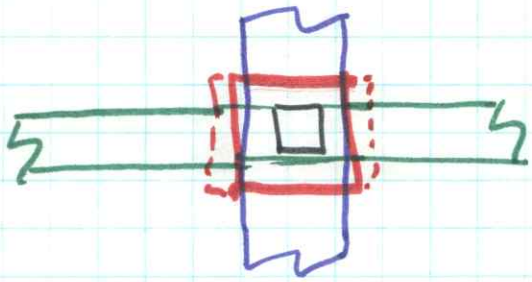
- Contribution of pass transistor to average dc power will be switching power dissipated by driving circuits at edge of arrays. (we'll go into more detail later in course)
  For now: look for pullups / shift logic
- Don't use poly or diff at these max densities. If must make crossover/crossunder use Diffusion. Widen it a bit. Keep it short. Else if >100's of diff --- it will start dropping voltage significantly.

LAYOUT EXAMPLE: on whiteboard:



[Note: local compression doesn't affect repeat distance/area]

Now note: we could do this: The rules allow:



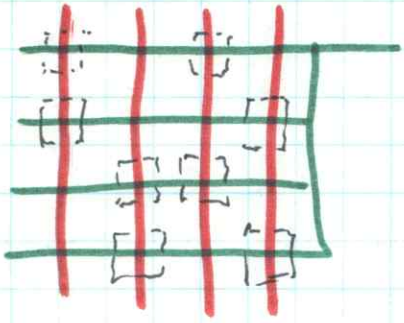
[But recommend if do this, enlarge to 1.5 to 2 lambda at edges over green]

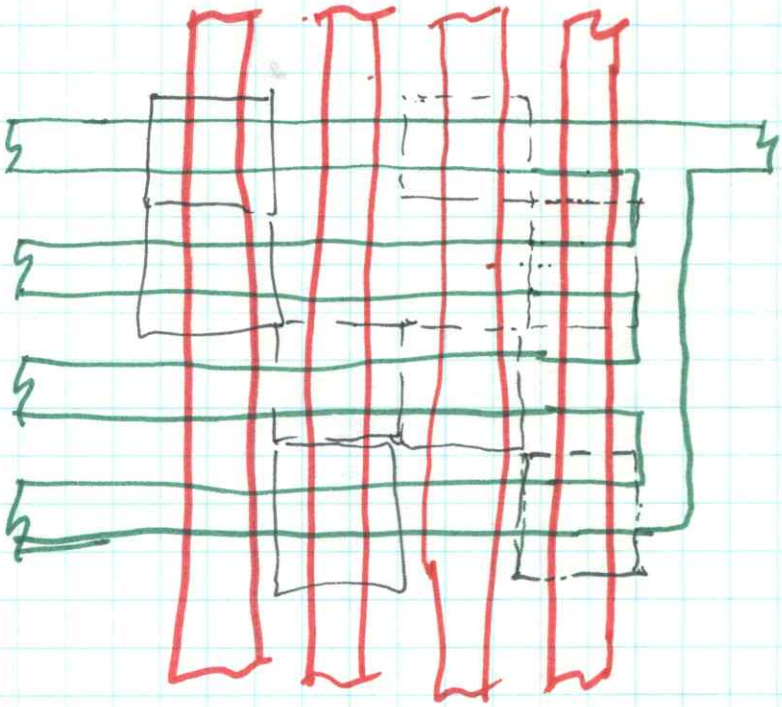
(Don't want to short out)

[Also, this may not scale well as oxide gets thinner]

- But Note: Delays get longer! No such thing as a free lunch.

LAYOUT EXAMPLE:





> LAYOUT EXAMPLE: START INTO & GO AS FAR AS CAN IN SRCELL

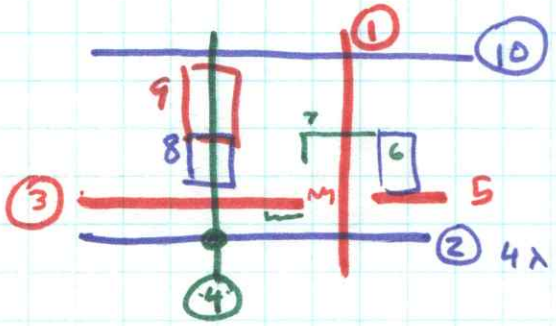
(SLIDE & WHITEBOARD)

If we're going to use a cell a lot, may want to work hard to make it small, fast, low power, etc.

But extreme compression (use of 45° lines and other ~~x~~s, AND MORE & MORE FINE STRUCTURE) CONFLICTS NOT ONLY WITH DESIGN TIME BUT ALSO LAYOUT DESCRIPTION (AMOUNT OF CODE).

[BIOLOGY ANALOGY: WANT SUBSYSTEMS THAT PERFORM FUNS USING ARRAY OF SINGLE CELL TYPE, PERHAPS SURROUNDED BY INTERFACE CELLS, ALL OF WHICH IN ADDITION TO SPEED LOW POWER ETC IS DESIRABLE IN MIN. AMOUNT OF CODE (GENES)]

WALK THRU design



Pullup/pulldown?
8:1
[what is it here: ~9:1]

A guess at min area. 3:1 pullup, 1:3 pulldown

> Now what rule makes it 21λ wide? Why but it 20λ wide?
[3λ diff-diff]

> Metal Lines could be narrower, but wouldn't make it smaller in this case. Hint at how to make it smaller
[moving cutting contact. Poss Diff]

> this version draws a bit of power. If used 16λ : 2λ pu and 2λ : 2λ pulldown, would use ~1/3 power

But would have longer pu transit time, and wouldn't drive outputs as fast, but would be less load on inputs.

> Actually might not be much bigger — might be able to angle the pullup around a bit

---

> If have time (unlikely), could start PLA cell layout.

EXPLAIN/CLARIFY HW #3 problems 10 and 11

i.e. read carefully to identify constraints given and constraints not present

[Sketch layouts as in Fig 86 chap 4
SHADE IN COLORS LIGHTLY
IF you want to apply another constraint do so but state it!
(SHOW SLIDE OF TOPOLOGY OF STACK
[SUGGEST TRACING DIRECTLY MIRRORED CELL TO CHECK CELL-CELL DESIGN RULES]

6.978 LECTURE #7. TUESDAY OCT 3

TODAY: SOME MORE CIRCUIT & DELAY CALCULATIONS WHICH AFFECT LAYOUT GEOMETRIES ··· SYST. DESIGNS

(i.e., NOW THAT YOU KNOW WHAT LAYOUT IS LIKE, YOU SEE WHY ITS A GOOD IDEA TO HAVE THINGS IN ORDER AT THE HIGHER LEVELS BEFORE COMMITING TO ALL THAT WORK !!!)

THURS: SUBSYSTEM, CIRCUIT, STICK, & LAYOUT OF SEVERAL INTERESTING SUBSYSTEMS.

(examples. on order of small to moderate first project)

[NO CLASS NEXT TUESDAY]

THURS (NEXT WEEK): { HOW ARE DESIGNS ARE IMPLEMENTED (INTRO)
HOW TO DESCRIBE LAYOUTS: USE OF SYMBOLIC LAYOUT LANGUAGE.

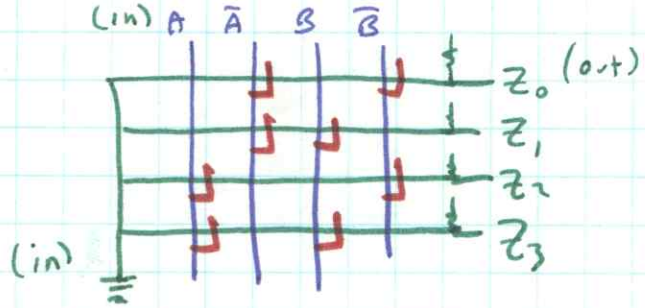
- HAND IN HW #3
- ILL HANDOUT HW #4 next time. I want to see how you did with these layout problems first --- PROBABLY WILL BE A STICK + LAYOUT OF ONE FET
- Start Thinking About projects. During next two weeks you'll finish all the basics you need to begin. Also we'll see more examples. Perhaps sketch out any ideas of interest. I urge you all to find collaborators: for at least design checking. Talk to others - it may help you get an idea - if you have more than one idea - share with others. I will hand out a questionnaire sometime soon to see how you are coming along on these preliminaries. Constraints, Alternatives, Post Mortems?

6.978. LECTURE #7 TUES OCT 3

- Discuss HW #2 before we begin lecture:

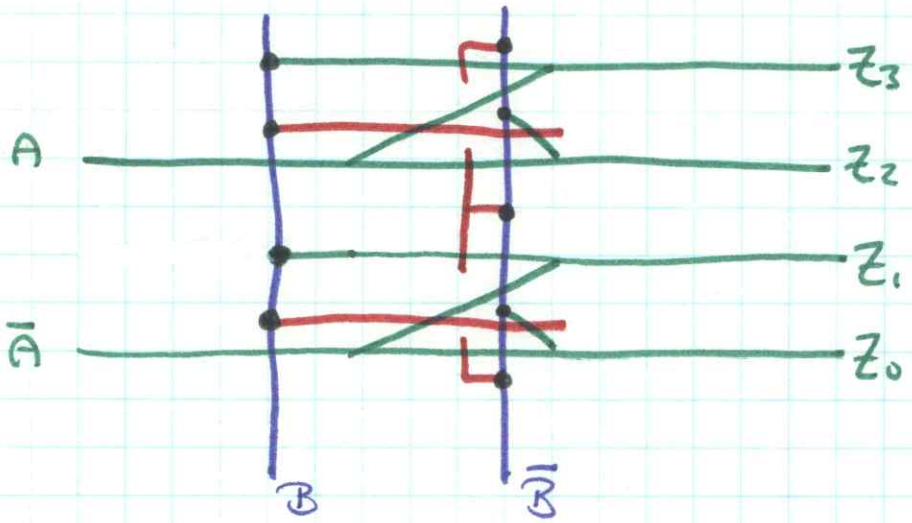
(PROB 5)

> As we'll see later, our old friend the "selector" circuit will turn up to have many uses when installed in different ways. A solution to 5(a) is to reverse the input/output, and hook up VDD & GND:



A B	z_0 z_1 z_2 z_3
0 0	0 1 1 1
0 1	1 0 1 1
1 0	1 1 0 1
1 1	1 1 1 0

> There are many other variants of this solution which are similar in structure. Ah! But look at the solution Clement Leung discovered using only a switch array with no VDD & GND:



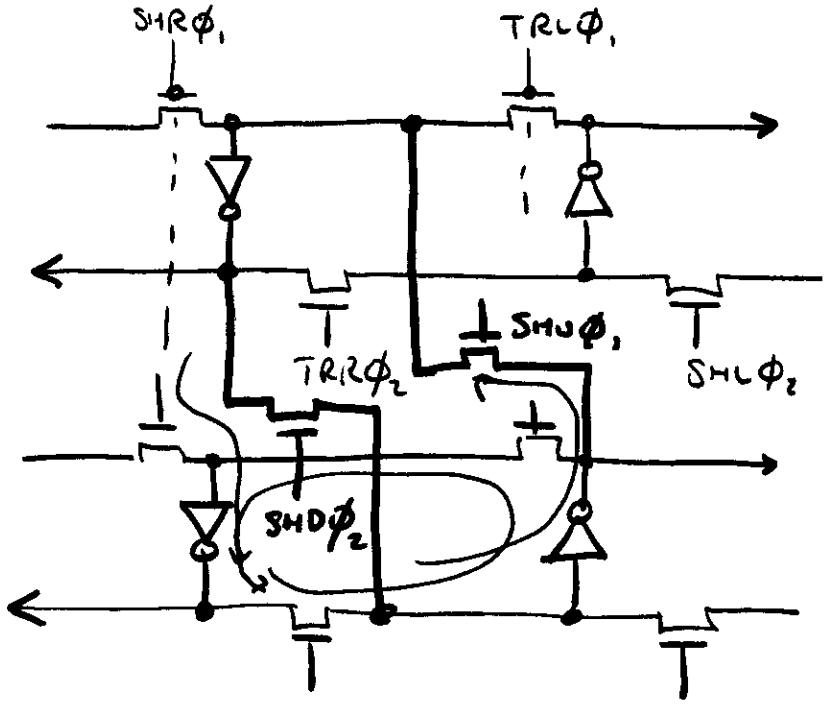
For example: If $\bar{A}=1$, then $\bar{B}=1$ passes to z_0, & $B=0$ blocks move up and more down, and $\bar{B}$ passes $B=0$ to z_1

If $\bar{A}=1$, then $B=1$ passes to z_1, $\bar{B}=0$ blocks more through, $\bar{B}=1$ passes $B=0$ to z_0

, etc.

HW #2 (cont.)

> **THE 2-D STACK:**
(Prob 6)



Some errors: running straight up/down, so signal propagates across array (data lost).

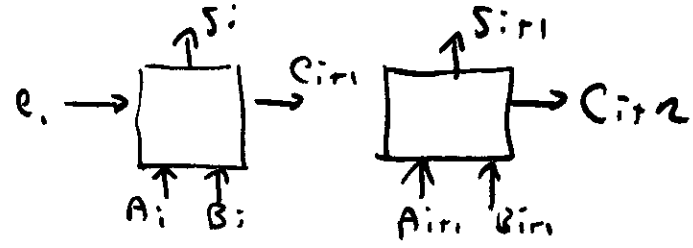
[Note that inverter output goes thru clocked line into node having no other inputs active, and outputs blocked by unclocked lines. For example, this doesn't happen →⊣]

Another error: Cycling SHU, SHD ended up shifting data right or left

> **MANY** (Stick) layouts possible. Don't know what might be best layout.

> **The Adder:** Most people got right answers for details of PLA's for problems 7 & 8.

However:



Draw Don't clock the carries

unless specify **CONTEXT** As Serial Adder

HW (cont.)

- I don't check all details of PLA code, just one or two product terms, ~ 1 output, & consistency. If you aren't sure of them, recheck them, & ask questions.
- Note importance of **context** of next level: does structure at one level not only satisfy the "rules" of that level, but fit properly at the next. Example: the "clocking of the adder."

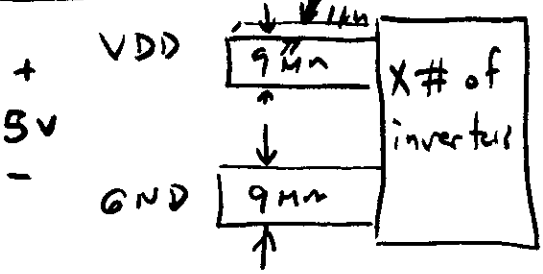
FROM LAST TIME: Rushed thru an important point. Repeat it too make sure you've noted it (Also, as you must now see - we want to get things right at the higher level before we start hacking out the layouts):

OM SLIDES

Question: How many minimum sized inverters will a minimum size VDD -- GND wire supply?

Line is $3\lambda = 9\mu m$ wide by $\sim 1\mu m$ thick:

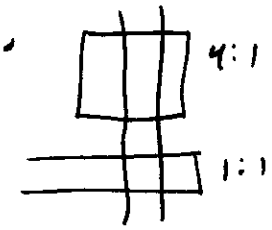
Have a bunch of inverters:



FOR 4:1 INVERTER,

$R \sim (4+1)10^4 = 5\times10^4\,\Omega$

$\therefore I = \frac{5}{5\times10^4} = 0.1\,mA$



> **LIMIT** $\tilde{=} 0.5\,ma/\mu m^2$, $\therefore$ 9μm wire carries ~ 5ma $\therefore$ **X ~ 50**

> But Half are usually off $\therefore$ X ~ 100

> And if 8:1's, then then X ~ 200

> But if mike 8:1's like 

then X ~ 100

BEGIN MAIN LECTURE:

AS WE'VE SEEN, CIRCUIT/SYSTEM CONSIDERATIONS OF POWER & DELAY AFFECT LAYOUT GEOMETRIES:

> WE ALMOST ALWAYS USE MIN L PULLDOWNS, FOR MIN T, BUT PULLUP LENGTH IS A FCN OF RATIO CONSIDERATIONS.

> WIDER PULLDOWNS (& PULLUPS) DRIVE LOADS FASTER, BUT ARE SLOWER TO BE DRIVEN.

YOU'VE SEEN HOW TEDIOUS LAYOUT IS (AND THERE IS STILL ONE MORE STEP IN INSTANTIATION: LAYOUT DESCRIPTION TO GO! ... WE GET TO THAT NXT WEEK).

> SO LET'S GO BACK AND MAKE SURE WE UNDERSTAND DELAYS IN DRIVING CAPACITIVE LOADS A BIT BETTER. ALSO, HOW TO MAKE BETTER DRIVERS. ETC

DRIVING LARGE CAPACITIVE LOADS:

SLIDE: Remember Fanout? The bigger the load, the slower it is driven.

Question: What do we do if we have a really BIG load? For example, going off-chip?

How can we drive a big C_L in minimum time, starting with signal on gate of MOSFET of C_g?

(1) Define $C_L/C_g = Y$.

Intuitively, we might think to drive a larger inverter from C_g, then a larger one, etc., until at some point we have an inverter big enough to drive C_L.

(2) Suppose we cascade inverters, each larger by a factor f.

Then each stage has a delay of fT (true to first order) we'll see why later

③ If inverter delay is τ (or prop. to τ forgetting k for the moment) then N such stages have a delay of $\approx N f \tau$. (Will see why in S.nov. Sutton)

④ But $f^N = Y = \frac{C_L}{C_g}$

⑤ If use large f, need fewer stages (smaller N) but each stage will have longer delay.

If use small f, need more stages, but each will have shorter delay.

Suppose $Y=16$, could use $f=2, N=4$ or $f=4, N=2$

What N minimizes overall delay for given Y?

$f^N = Y$; $\ln(Y) = N \ln(f)$: $\therefore N = \ln Y / \ln f$

Delay of one stage $= f\tau$

Total delay $= N f \tau = \ln(Y) \left[\frac{f}{\ln(f)}\right] \tau$

$\therefore$ Delay is proportional to $\ln Y$

Figure 5 plots $\frac{f}{\ln(f)}$ as fcn of f, normalized to its minimum value of e

The minimum total delay $= \tau$ times e times natural ln C_L/C_g

$$\text{Min Tot. Del.} \sim \tau e \ln\left[\frac{C_L}{C_g}\right]$$

when $f = e$

O.K. What does this mean? The implications are really quite important:

> Off chip drivers go fastest when you build up with a factor of e per stage.

> But, speed isn't everything. If back off to $\tilde{f} = 6$, Almost as fast, but less area.

> SHOW ON DRIVER SLIDE

[Discuss the whole issue of the additional Benefit To be derived from VLSI: Less offchip Boundaries to cross.]

[Use of inword compatible Designs; Not optimized to current chip size, to develop scalable designs for VLSI]

There will be further important uses of this simple set of ideas in the development of a Theory of the space, time, and energy costs of computation in hierarchically organized systems, in Chapter 8 pp 49-57

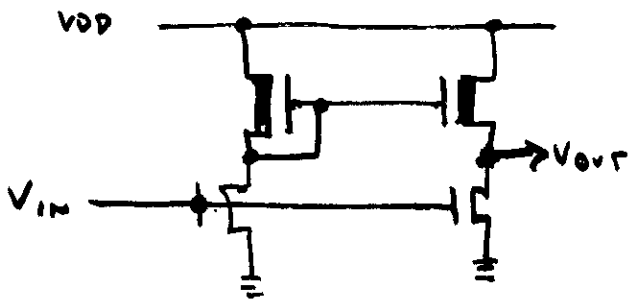
SUPER BUFFERS

- Ratio Logic as we've seen has an asymmetry:

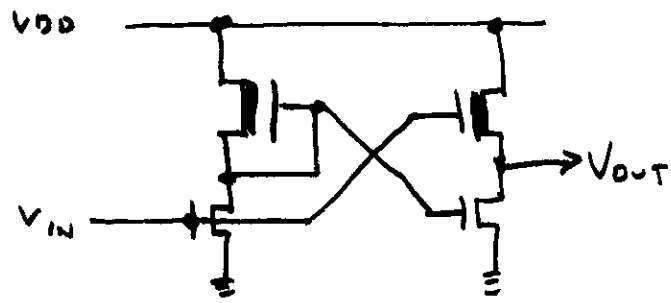
  It can discharge a C_L thru its pulldown much faster than it can charge one thru its pullup.

- There are ways of getting around this problem, esp. useful for drivers (at edges of arrays):

- Here are two circuits which are approximately symmetrical in their drive capability, even though they have 4:1 Z ratios:



INVERTING SUPER BUFFER



NON-INVERTING SUPER BUFFER

- IN EACH CASE, WHEN PULLDOWN INPUT TO 2ND STAGE IS ~VDD, THEN THE PULLUP GATE ~ 0V AS IN USUAL INVERTER. SO PULLDOWN THE SAME

- BUT, WHEN PULLDOWN INPUT TO 2ND STAGE GOES TO ZERO,

  THE PULLUP GATE GOES RAPIDLY TO VDD, SINCE IT IS ONLY LOAD ON PREVIOUS STAGE.

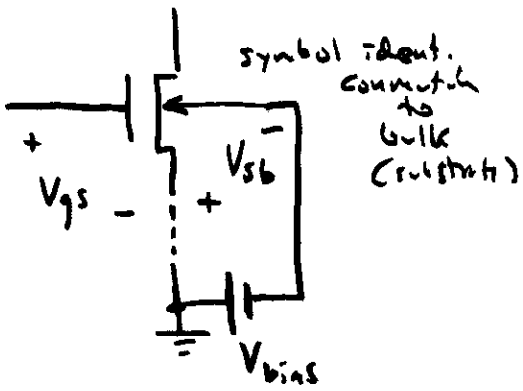
> The PULLUP is TURNED on with ~ 2x the voltage it would normally have with gate tied to source.
> Since in saturation, $I \propto V_{gs}^2$, the SUPER BUFFER PULLSUP about 4x as fast as regular inverter.
> i.e., PULLUP / PULLDOWN are ≈ symmetrical. $\frac{1}{4} = \tau$

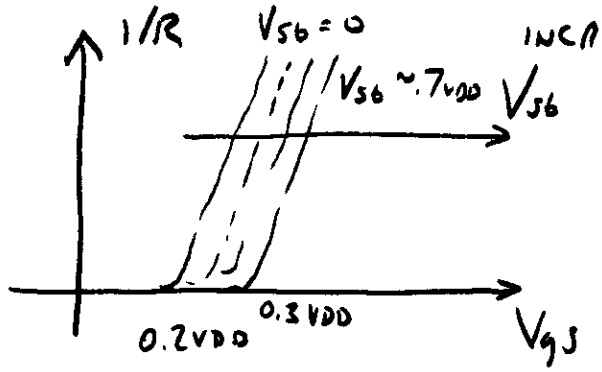
[that's why we got away with τ in the last section]

some eg. cases of using super buffer.

The Body Effect: We've considered V_{Th} to be a constant, independent of the source to substrate voltage (i.e., only a few of V_{gs}).

- It ain't quite that simple. Consider:





- So, V_{Th} gets larger as we raise the source voltage up from ground. Comment again on V_{gs}, V_{sb} ---

Why Is This Increase Important?

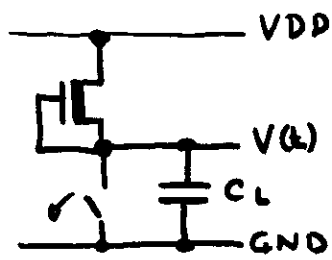
- Because it increases 2 effects we've already introduced informally:

  (1) The difference in time and final output voltage between an enh. & depl. mode FET driving a capacitive load.

  (2) The need to increase the Pullup / Pulldown Z ratio, when coupling logic stages by pass transistors.

DEPLETION MODE vs ENHANCEMENT MODE Pullups:

Depletion Mode (what we'll normally use):



- In the mid to latter stages of a rising transient, $V_{gs} \geq V_{th}$ and $V_{gd} \geq V_{th}$
- So the pullup is in its resistive region.
- The final stages of the rising transient are given simply by the exponential:

$$V(t) = VDD\left[1 - e^{-t/RC_L}\right]$$

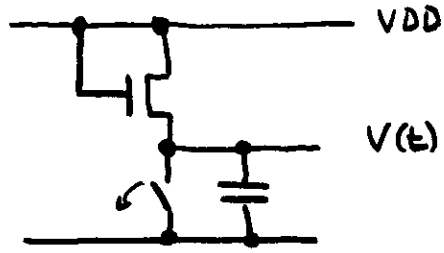
i.e., V(t) goes rapidly to VDD, with time constant RC_L.

- For inverter ratio k, pulldown tran time τ and gate C C_g, the time constant is

$$RC_L \sim k\tau C_L/C_g$$

Enhancement Mode

(used in early MOS)



- Since $V_{gd} = 0$, the pullup is in saturation whenever $V_{gs} > V_{th}$
- The problem: As the output voltage approaches $VDD - V_{th}$, the current supplied by the FET decreases rapidly
- In the Book, we derive: for large t:

$$V(t) \cong VDD - V_{th'} - \frac{C_L LD}{\mu\varepsilon Wt}$$

SHOW SLIDE AND COMMENT

[worsened by body effect]

Depl. mode crisply goes to VDD.
Enh. slowly (never) quite makes it to $VDD - V_{th'}$ (body effect)

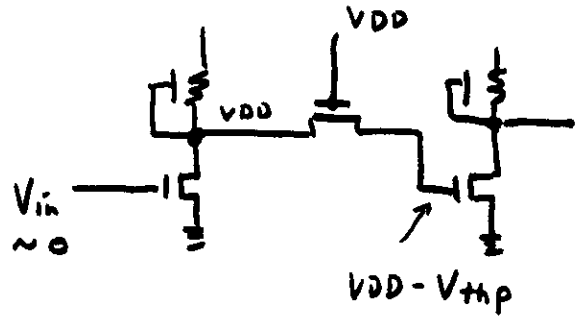
(IF TIME)

Pullup/Pulldown Ratios For Inverting Logic Coupled by Pass Transistors.

- We found earlier that $4:1 = Z_{pu}/Z_{pd} = \frac{L_{pu}/W_{pu}}{L_{pd}/W_{pd}}$

  yields equal inverter margins and also provides output sufficiently less than V_m for input = VDD.

- For stages of inverters coupled by pass transistors, such as



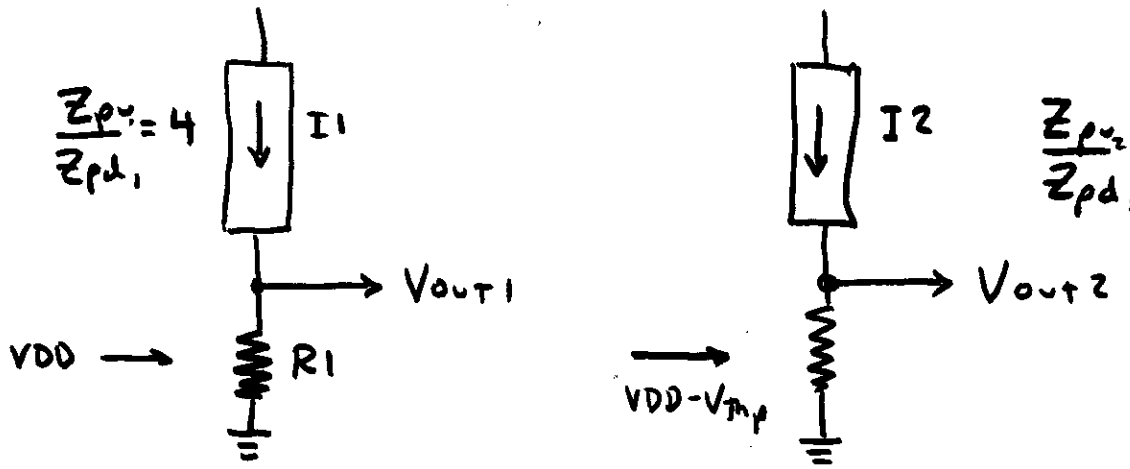
- If input to first is zero, and thus output ~VDD. If Pass T input is VDD, then at most the input to the second stage is $VDD - V_{thp}$.

- Since Pass T source is near VDD, V_{thp} is near its maximum of $\approx 0.3VDD$.

Question: What must be the Z_{pu}/Z_{pd} of second stage, if it is to have its output go as low with input = $VDD - V_{thp}$, as would a 4:1 with input VDD?

- With INPUT near VDD, the pullup is in saturation, and pulldown is in resistive region.

compare the two equivalent circuits:



- For V_{out1} to equal V_{out2}, $I_1 R_1$ must = $I_2 R_2$.
- IN SAT: $I_{ds} = \frac{\mu \varepsilon W}{2LD}(V_{gs} - V_{th})^2$ (eq 5)
- IN RES: $R = \frac{V_{ds}}{I_{ds}} = \frac{L^2}{\mu C_g (V_{gs} - V_{th})}$ (eq. 3a)

Substituting we will find:

$$\frac{Z_{pu_1}}{Z_{pd_1}}\left[VDD - V_{th}\right] = \frac{Z_{pu_2}}{Z_{pd_2}}\left[VDD - V_{th} - V_{thp}\right]$$

- Now: V_{th} of pull downs is $\approx 0.2\,VDD$, and V_{thp} of pass Ts is $\approx 0.3\,VDD$ to $0.35\,VDD$. (to be safe)

BODY EFFECT

$\therefore \; Z_{pu_2}/Z_{pd_2} \sim 2\, Z_{pu_1}/Z_{pd_1}$

$\frac{Z_{pu_2}}{Z_{pd_2}} \sim 8:1$

(WE USUALLY EITHER SPECIFY OR MEASURE ~~FOR INVERTERS NAND, ETC~~)

~~- WILL HAND OUT HOMEWORK NEXT TIME~~
~~- IF TIME, TALK~~

REMEMBER, THINK ABOUT PROJECTS A BIT.

[Think about what you'd like to do. Sketch out some ideas. Talk to others. Would you like to collaborate? On project? or just for checking? Any ideas others might use?]

6.978. LECTURE #8: OCTOBER 5

- BE SURE YOU'RE REGISTERED ... WILL SET UP ACCOUNTS BASED ON WHOSE REGISTERED (OR MAYBE) NETWORK
- TODAY: EXAMPLES:
  - LAYOUT OF THE PLA
  - DESIGN OF THE BARREL SHIFTER
  - DESIGN IDEAS FOR A SERIAL BIT-STRING COMPARATOR

- WE'RE ~1/3 THRU COURSE. SO FAR STARTUP TRANSIENT -- NEW STUFF -- HW
- NEXT 1/3 BEGIN TO LEARN BY DOING -- STUDY EXAMPLES, DO INFORMAL STUFF IN LAB -- FIND OUT WHAT DO WE NEED TO KNOW --
- FINAL 1/3 PRESSON TO FINISH A PROJECT. SPECULATE ABOUT FUTURE

- LETS USE WHAT LEARNED SO FAR, STUDY EXAMPLES:
  TO CLARIFY MATL. PERHAPS RAISE SOME QUEST.
  GIVE US INSIGHT INTO POSSIBILITIES OF DES. INTEG. STRUCTURES

- WE'VE SEEN THAT THE PLA IS AN IMPORTANT SUBSYSTEM.
  WE'LL USE IT TO BUILD C/L, AND FINITE STATE MACHINES

- ALMOST EVERY SYSTEM WE BUILD WILL HAVE SOME PLA IN IT.

- LETS REVIEW THE STRUCT/FCN OF PLA, & THEN DEVELOP A LAYOUT

- WE CAN ALL USE THIS LAYOUT IF WE WISH IN OUR DOINGS

- THIS LAYOUT ISNT "HARD" OR "TRICKY" IN TERMS OF DESIGN RULE CHECKING.

  BUT IT IS A NEAT & COMPACT LAYOUT. THINGS JUST SORT OF FALL INTO PLACE

- THIS EXCERCISE WILL ILLUSTRATE THE USE OF & IMPORTANCE OF BREAKING DOWN A LAYOUT INTO A SMALL # OF MANAGEABLE, IDENTICAL, CELLS,

  WHICH CAN BE REPEATED TO BUILD UP THE LAYOUT.

O.K. First lets review PLA design:

- Keeping in mind now that we're leading up to LAYOUT.
- i.e., how can we keep things SIMPLE & not Kludgey!

SLIDE OF PLA Circuit (TALK THRU FUNCTION)
NOR-NOR

- Recall that both Planes are similar, just tilted ---
  And that Pullups just along each edge, in similar positions.
- The interiors are all the same - except we're going to have to somehow place the transistors (program the PLA).
- Looks like the INPUT & OUTPUT parts are different, but each INPUT, each OUTPUT are just repeats.
- SO FAR, MAYBE JUST FOUR KINDS OF CELLS ???
  WE'LL SEE
- NOW, REVIEW STICK DIAGRAM:

SLIDE OF PLA STICK DIAG (TALK THRU STRUCTURE & FUNCTION A BIT)

- Now we see specific starting points for layouts of cells.
- Also, we see that maybe there are more than 4 cell types: what about the GROUND CONNECTIONS
  what about connecting the two PLANES
- i.e. The key question is how to break up the LAYOUT into simpler similar cells,

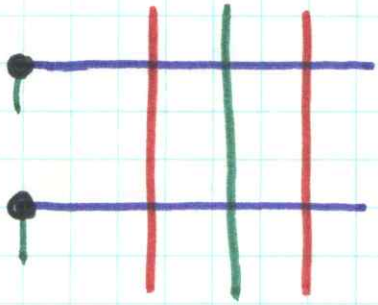
  Rather than just attacking the whole thing at once!

- We note that the most important region to do REGULARLY & COMPACTLY is probably the "PLANES"!

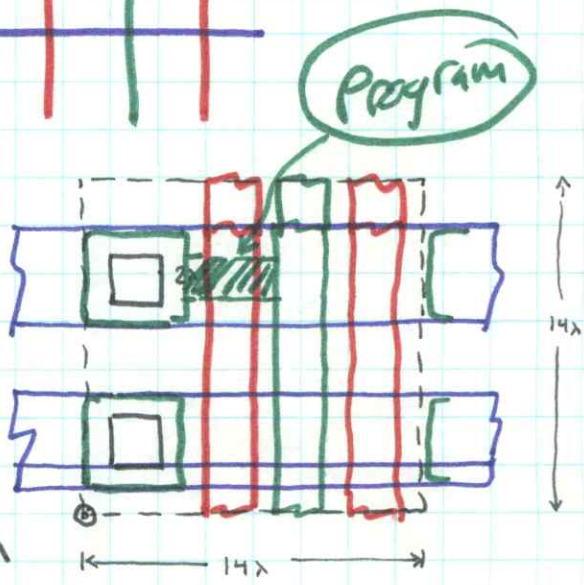
[THE PLANES WILL OCCUPY MOST OF THE AREA OF A LARGE PLA, EVEN IF THE PULLUPS, DRIVERS, etc., are not minimized.]

- LOOK AT A PIECE OF THE STICK DIAG OF THE "AND" PLANE

(TILT FOR "OR")



- This just rejects.
- How to lay out. Well, First put vertical reds and green as close ASPOSS:
- Then put metal across this, As close As poss: But contacts gets us, so use 4λ

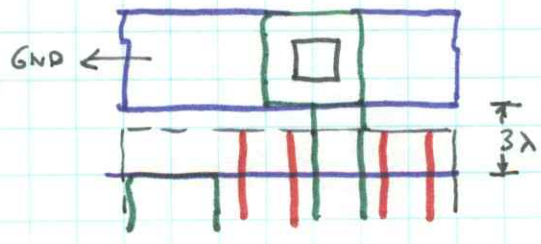


- PLACE DIFF FLASHES, DETERMINES PITCH OF CELL.
- So weve layed out a "PAIR of PLA Cells" For the AND/OR Planes

ITS SQUARE!

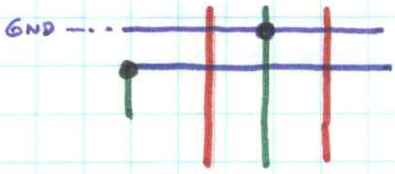
- WHAT ABOUT GND Return at edges of PLANES:

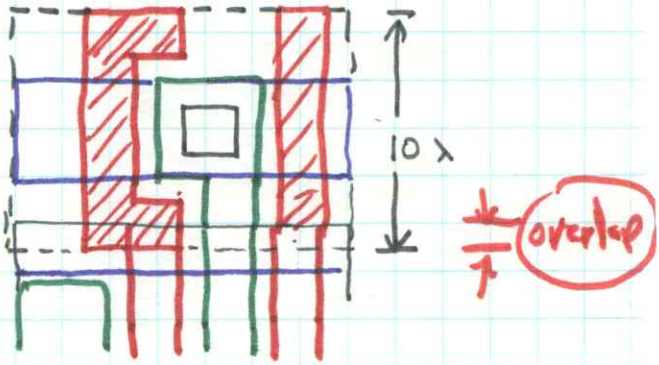
Could just use:



- BUT LETS BE COOLER! IN A GIANT PLA, WELL NEED MORE GROUND RETURNS.

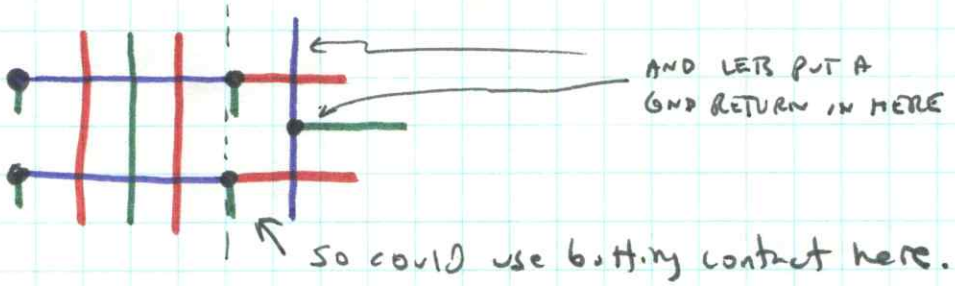
LETS MAKE A CELL THAT WE COULD USE AT INTERVELS, INSERTED AS ROWS, IN "AND" PLANE (Coli in "OR" Plane).



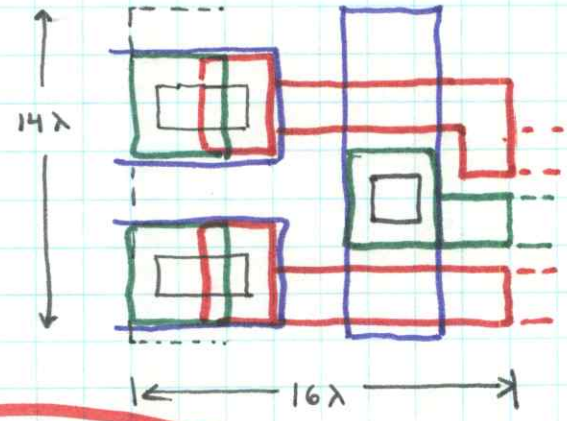


- LETS DESIGN CELLS TO CONNECT THE PLANES:

COULD JUST MAKE A BLUE to RED contact. But what if Rightmost PLA CELL DIDN'T HAVE A CONTACT? HOW WOULD WE MAKE TRANSISTORS THERE



- MUST LAYOUT 14 λ HIGH, WITH RED WIRES GOING OUT AT PROPER PLACE TO ENTER A "TILTED" PLA CELL PAIR:

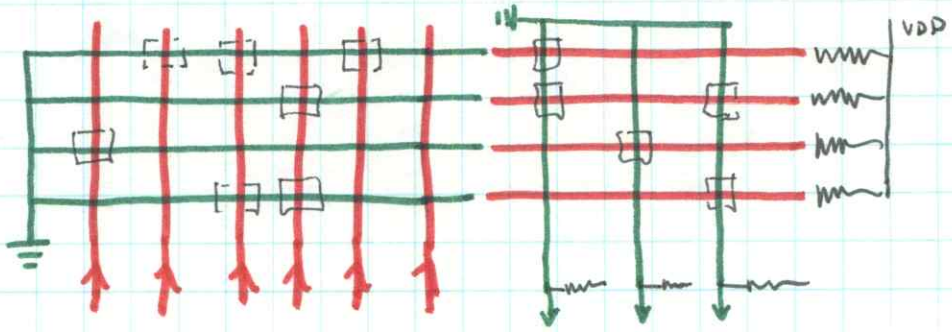


WHILE SEEM SIMPLE, NOTE, THESE ARE FATHER TRICKY, MINIMUM LAYOUTS

SHOW SLIDES CELLS, OVERALL, MANIPULATE CELLS

(COVER IF TIME)

- REMEMBER, WE COULD HAVE MADE A NAND-NAND PLA RATHER THAN NOR-NOR:



- What would be advantages of this type of PLA?

  (much smaller area)
  except watch out for pullups!

- What would be disadvantages of NAND-NAND PLA?

  > much slower

  > speed is a fcn of size

  > WORST: PULLUP SIZE IS A FUNCTION OF SIZE!
  (So can't use same cell for all such PLAs)

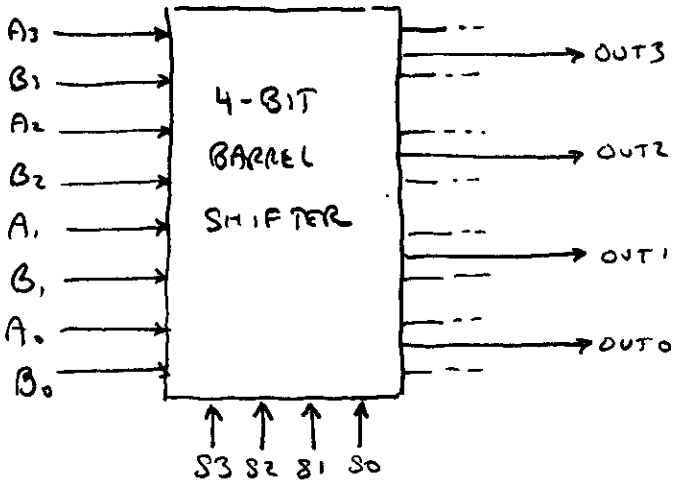
MORAL: DON'T USE NAND-NAND PLA UNLESS IT IS SMALL, YOU ARE TIGHT FOR SPACE, AND YOU CHECK YOUR DELAYS & PULLUP/PULLDOWN RATIOS CAREFULLY.

THE BARREL SHIFTER

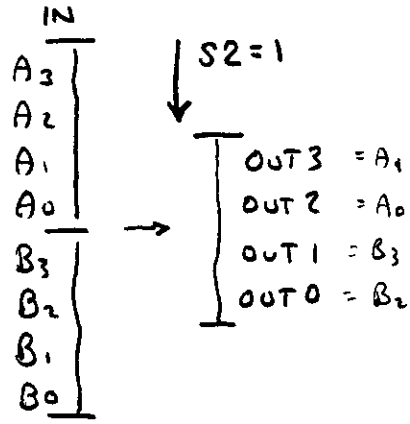
- ONLY C/L, in fact can be done with a switch array, but is so useful and so clever, that I consider it an important "SUBSYSTEM".
- This maps nicely into silicon, and is kludgey using purchased parts.

SHOW OM SLIDE THE OM uses a 32 input, 16 output Barrel shifter. Used for all sorts of field extraction & alignment operations prior to inputting data to the ALU.

- Let's Describe and Design a FOUR BIT Barrel Shifter 8 in 4 out so we can keep the details in control & visualize what is happening.
- However, this design is directly extensible to a 16 bit Barrel Shifter.
- Block Diagram: Context: Embedded in a system. Two 4-bit Buses run right thru it!

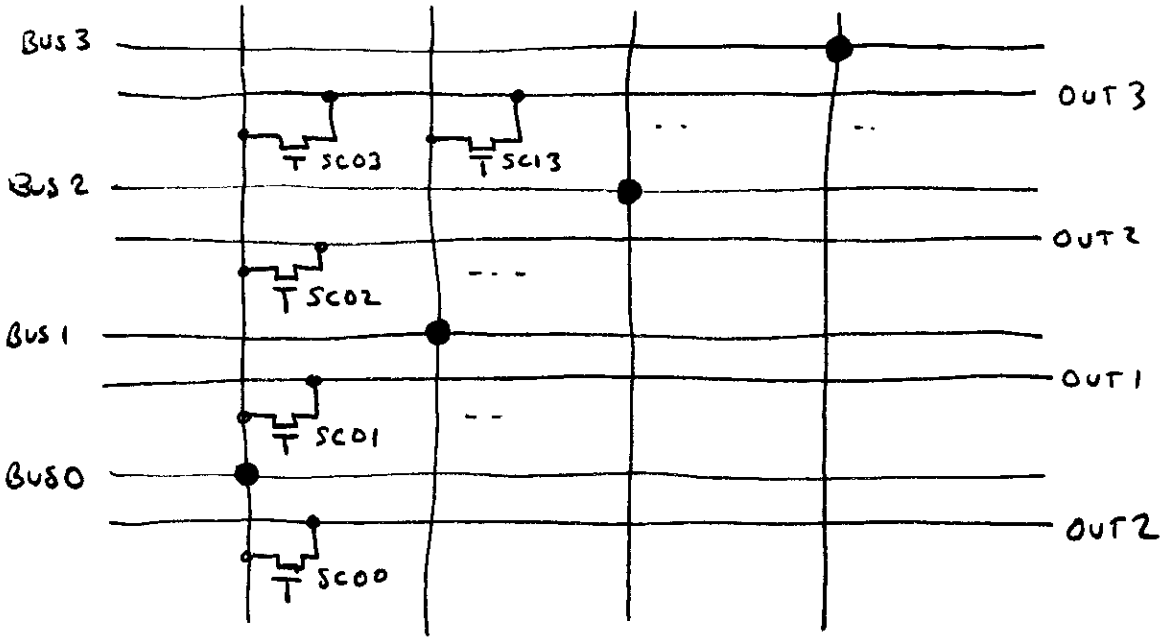


CONCEPTUAL PICTURE OF FUNCTION:



- NOW, WE NEED A WAY OF CONNECTING ANY BUS BIT WITH ANY OUTPUT BIT. THUS WE MUST HAVE DATA PATHS RUNNING VERTICALLY.

- A SIMPLE CIRCUIT IS A 4x4 CROSSBAR --- THIS GIVES US SOME STARTING IDEAS:



- HERE, SWITCH SC_{ij} connect BUS_i to $OUTPUT_j$

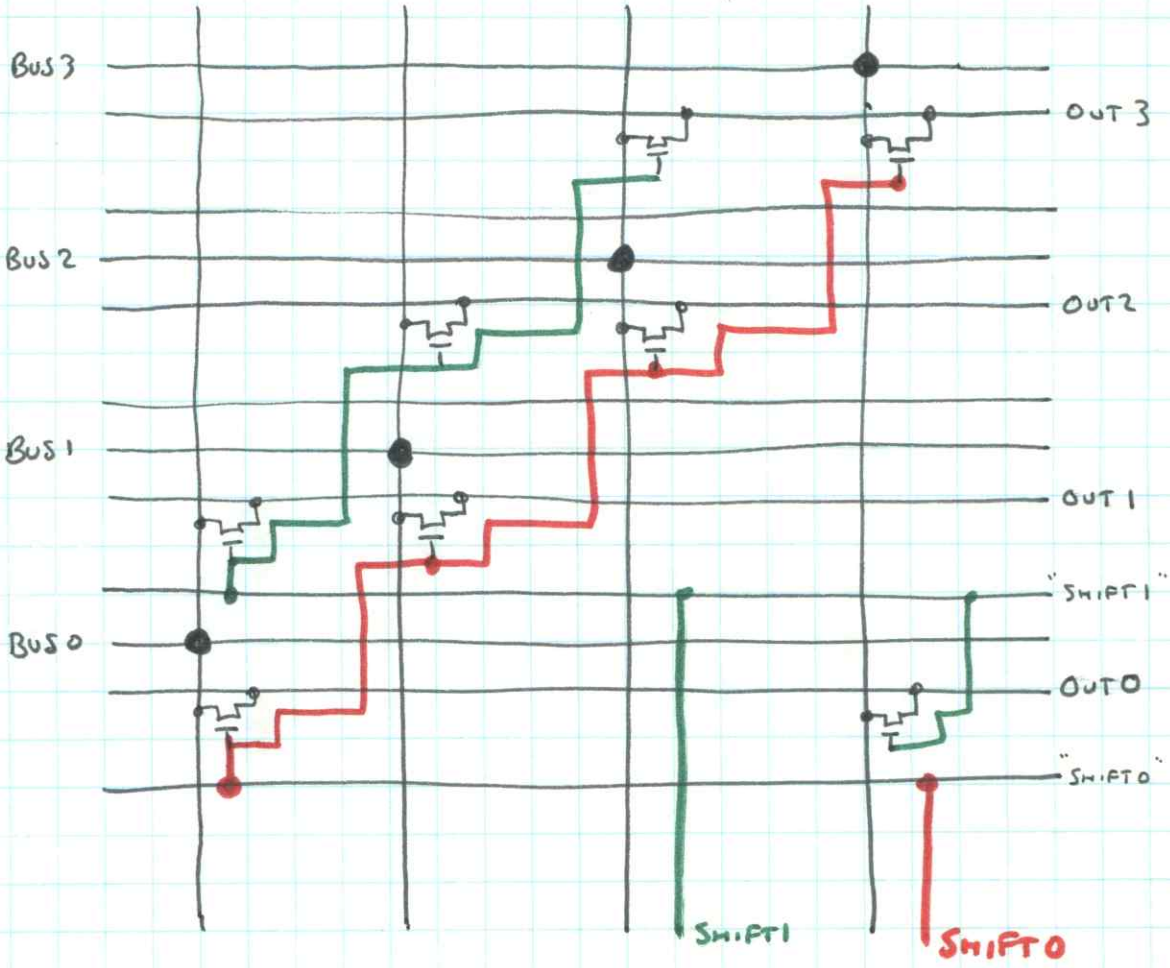
- WE COULD DO ALL SORTS OF SHIFTING, INTERCHANGING WITH THIS STRUCTURE.

- Ah! BUT IT HAS N^2 control lines that we must get into it. This might not be too bad for small N.

- BUT THERE IS A WAY TO CONNECT SOME SUBSETS OF THESE SWITCHES TO FORM A SIMPLE BARREL SHIFTER.

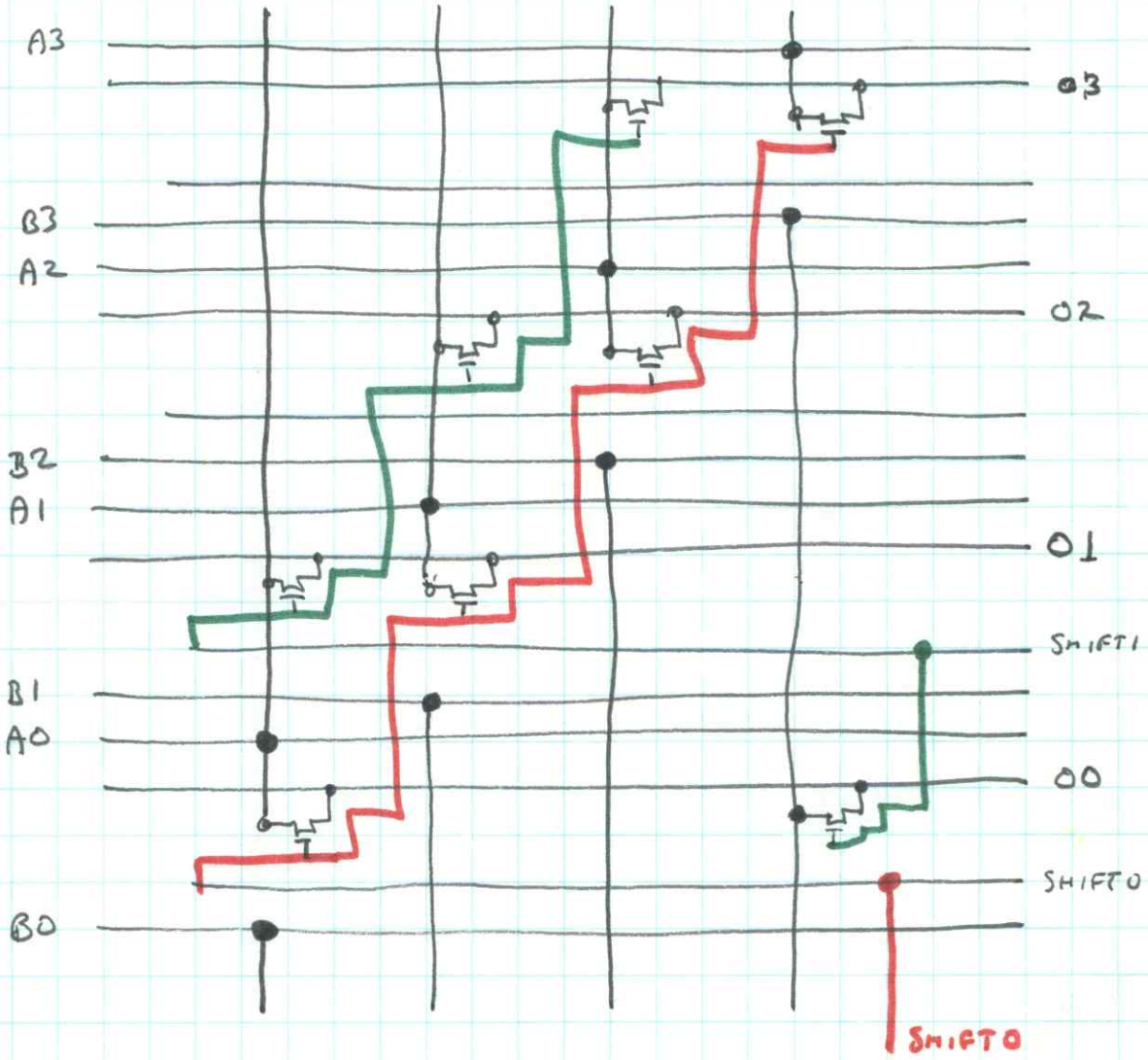
(cont.)

DRAW SAME DIAGRAM: WITH A FEW MORE LINES:



- DRAW IN ALL FETS TO CONNECT BUS i TO OUTPUT i (SHIFT = 0) THEN HOOK THEIR GATES TOGETHER, i, TO A LINE CALLED SHIFT 0
- DRAW IN ALL FETS TO CONNECT BUS i TO OUTPUT $i+1$ (SHIFT = 1) THEN HOOK THEIR GATES TOGETHER TO LINE CALLED SHIFT 1
- ETC.
- ONLY ONE OF THE SHIFT LINE MAY BE ON AT ANY ONE TIME.
- THUS WE HAVE A FOUR X FOUR BARREL SHIFTER

NOW, HOW DO WE GET 2 4-BIT #'S BARREL SHIFTED TO ONE 4-BIT # OUT, WITH ~~A~~GRACEFUL CROSSING OF THE WORD BOUNDARY? JUST ADD IN ANOTHER LINE FOR THE OTHER BUS, AND: SPLIT THE VERTICAL WIRES



- As Before, PLACE
  - > ALL FETS CONN. BUS; to OUT; . CONNECT GATES TO SHIFT0
  - > ALL " " BUS; to OUT;+1 . " " " SHIFT1

, etc.

- Note How Now on SHIFT1, ~~B0~~ ~~B3~~ ~~A0~~ ~~A3~~ ~~shift~~ (DK)

  A0 → 01
  A1 → 02
  A2 → 03
  B3 → 00

  ? ~~Hmm looks Strange~~ ! OK!

LAYOUT: The Barrel Shifter is one structure that seems easier to think of in circuit form. Try sticking it and you'll see what I mean!

- But, once the Fig 14 structure is developed, you can see that the layout in Fig 14a is equivalent.
- Busses Run thru in POLY
- Outputs Run thru in DIFFUSION
- ~~Add~~ Vertical Lines in Metal
- Shift Constants run Horizontally in POLY, cross FET GREEN, THEN VERT IN METAL
- SO, A PARTICULAR SHIFT CONSTANT CONNECTS A SET OF BUS LINES TO A SET OF OUTPUTS. **BASICALLY A SIMPLE TO CHECK LAYOUT**

LAYOUT NOT MIN-ON PITCH OF ALU

[THIS IS A VERY CLEVER SUBSYSTEM. I DONT KNOW WHO ALL WORKED ON IT, BUT MOST OF THOSE NAMED IN THE SECTION "OM PROJECT AT CALTECH" HAD SOMETHING TO DO WITH IT.

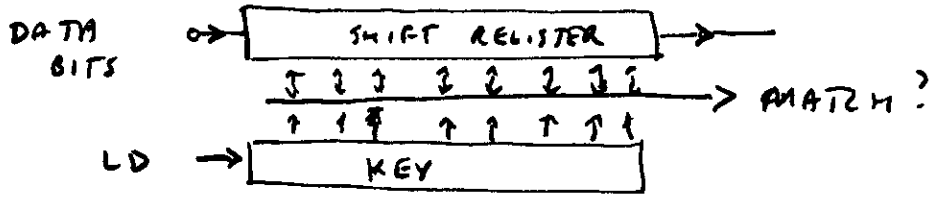
(A COUPLE OF MONTHS AGO I TALKED WITH BILL LATTIN WHO HEADS UP INTELS NEW ARCHITECTURE GROUP IN OREGON)--

(BOTH HE): I THINK WE ARE WITNESSING A PERIOD OF CLASSIC INVENTION. THINGS ARE HAPPENING REMINISCENT OF WHEN CONSISTENTLY GOOD STEEL + THE BOOTSTRAP OF GOOD LATHES & MILLS BECAME AVAILABLE IN THE 19TH CENTURY: A PERIOD OF GREAT MECH INVENTIONS: COLTS REVOLVER, THE IDEAS ASSOC. WITH INTERNAL COMBUSTION ENGINES, ETC.

SO, WHILE MAYBE MUCH OF THIS WILL BE AUTOMATED, IT WILL BE (AT FIRST) AT HIGHER OR LOWER LEVELS. BUT AT THE DAWNING OF SYSTEMS WITH THIS NEW TECHNOLOGY---. I THINK WERE GOING TO SEE A LOT MORE INTERESTING INVENTIONS

A SERIAL BIT STREAM COMPARATOR:

- LET ME POSE A PROBLEM: WE WANT TO MAKE A FAST/DENSE INTEGRATED SUBSYSTEM FOR DOING BIT STRINGS SEARCHES:

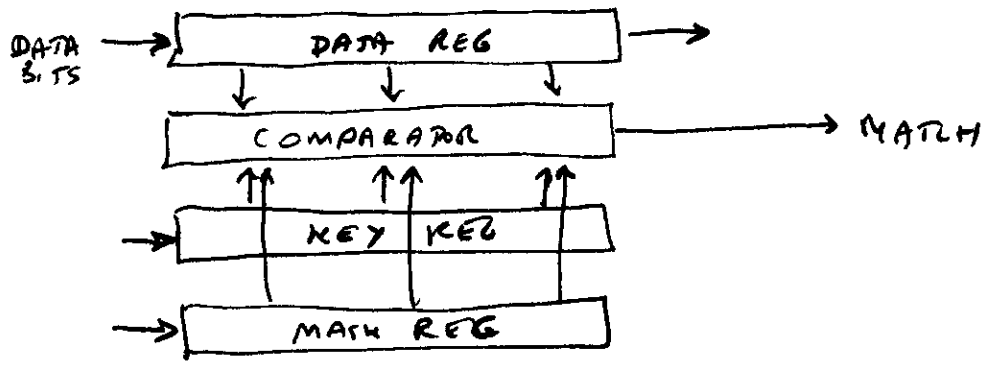


- MIGHT THINK OF LOADING A KEY STRING INTO A REGISTER, AND RUNNING DATA THRU A REG. NEXT TO IT AND SOMEHOW COMPARING ALL THE BITS, WHEN THEY ALL MATCH → GET A TRUE MATCH OUTPUT.

- WHY WOULD WE WANT TO DO THIS? LOTS OF SYSTEM APPLICATIONS. SEARCHING FOR DATA PATTERNS IN TEXT EDITING. COULD MAKE A SMART DISK SUBSYSTEM WITH A CHIP OUT THERE THAT COULD SEARCH FOR DATA BEFORE READING/WRITING. etc

- LETS BUILD UP MORE OF THE BLOCK DIAGRAM! MIGHT WANT TO BE ABLE TO SEARCH UNDER A MASK, SO WERE NOT LIMITED TO FIXED STRING LENGTHS:

So:

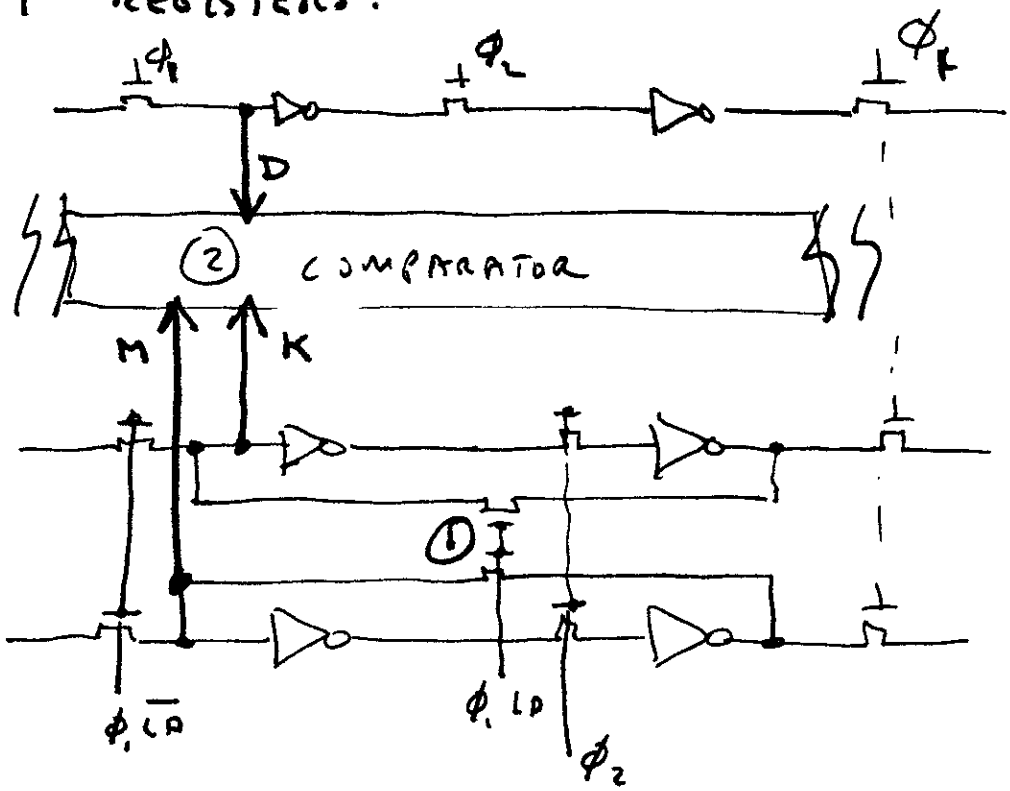


I.E., ONLY TRY TO MATCH THE SUBSET MARKED BY THE MASK BITS.

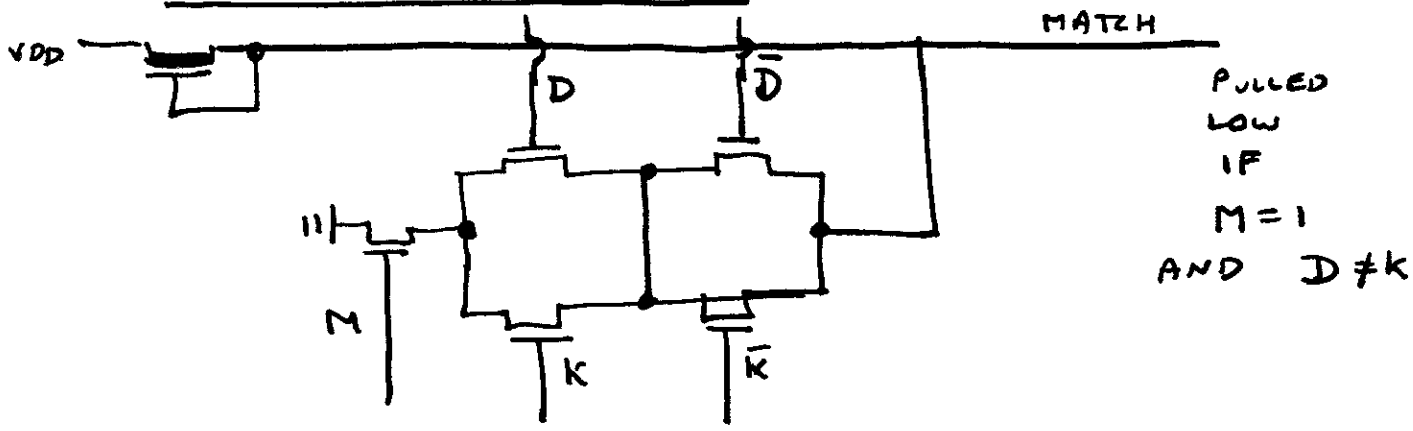
HOW WOULD WE DESIGN THIS:

- IN NON-INTEGRATED FORM, IT WILL BE EXPENSIVE: BECAUSE WE'D NEED SERIAL IN, PARALLEL OUT SHIFT REGISTERS – CAN'T MAKE DENSE BECAUSE OF LARGE PINOUTS.

- BUT WHAT IF WE USE OUR FAMILIAR nMOS SHIFT REGISTERS:

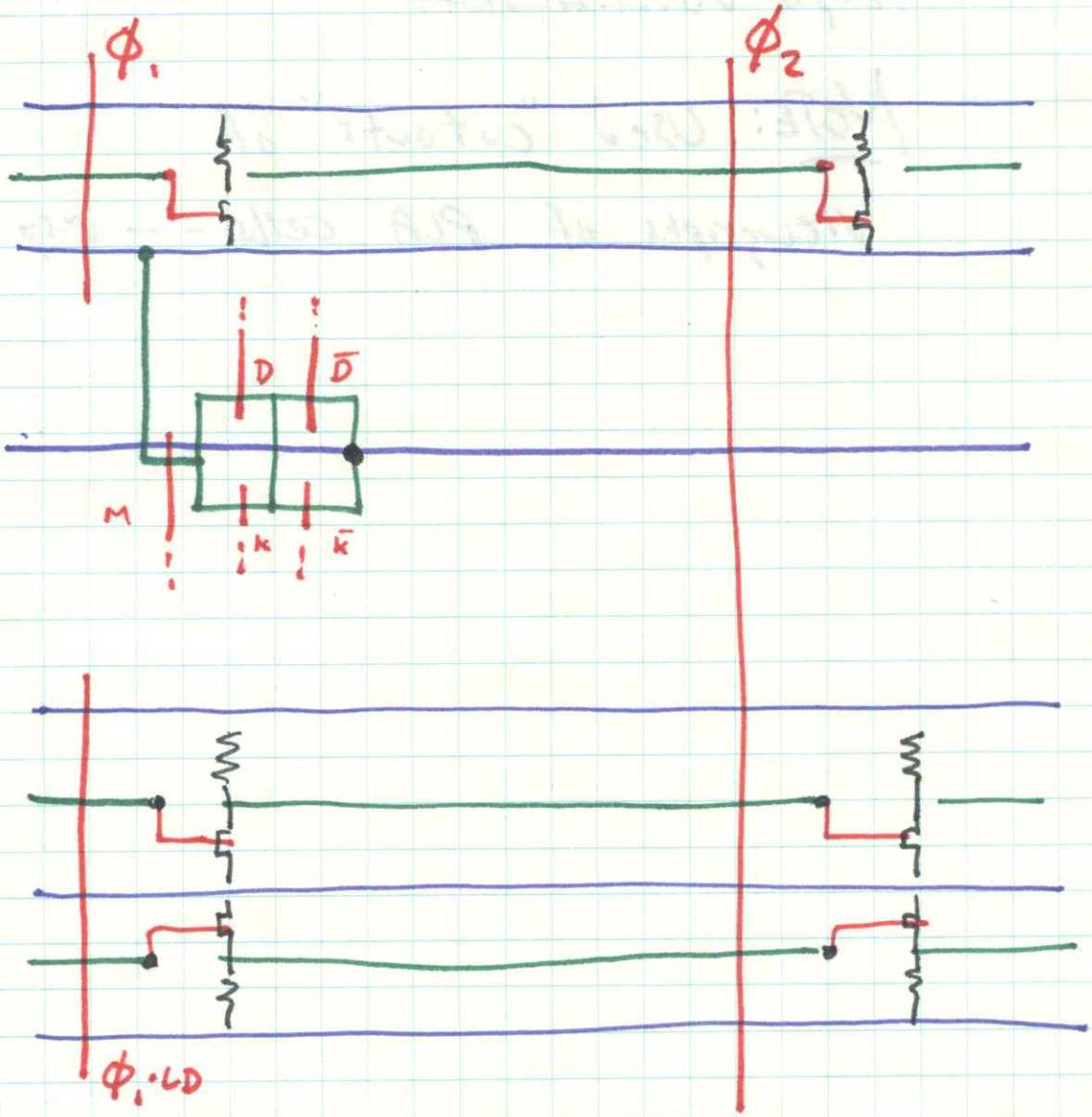


- MAKE LOWER TWO LOADABLE (1)
- BRING MASK, DATA, KEY TO COMPARATOR (2)
- NOW DESIGN COMPARATOR:



HAND OUT H.W. #4

- NOW, HOW MIGHT WE STICK DIAGRAM THIS?
- THINK A LOT ABOUT HOW TO ROUTE VDD, GND, AND THE CLOCKS/CONTROLS
- POSSIBILITY: START TO WORK PROBLEM:



- GOT TO GET FEEDBACK TO $\phi \cdot LD$ IN LOWER TWO SOMEHOW.
- MANY ALTERNATIVES FOR WHOLE STRUCTURE. DO WE WANT TO RUN CLK/CTL VERT? OR HORIZ. CONTEXT? YOU DEFINE

Lecture #9 — October 12

- **COURSE REGISTRATION**: TO GET LAB ACCT $, NEXT BOOK, MUST BE ON LIST
- Variety of things today: Introduction to digitizing/encoding layouts by use of a symbolic layout language.

  Introduce another example subsystem (sorter). Talk over your first project assignment, & prepare to begin the lab work.

- Briefly discuss HW#3: (a soln on whiteboard?)

Many got reasonable solutions to problem #9. Quite a few didn't. The lesson: Word descriptions don't define state machines in real cases. State diagrams or other formal things do. There were things unspecified in the problem: actually how to start the machine. Are one-bit messages allowed? Those interact. There are many solutions. Be sure however to note that you must get a double error at MSG ending in 0110 to satisfy the part that was stated.

The layout problems. I guess most of you internalized the process and design rule material quite well: most of the layouts had no design rule violations. As usual, some of the really compact layouts raised questions about how to really interpret the design rules, especially around the butting contact between gate and source. (Most common errors: incorrect red/green in BC)

Some compact solutions: (subst. smaller than most)

Problem 10:	Guy Steele	$304\lambda^2$
	Dean Brock	$324\lambda^2$ (right #'s only)
	Lynn Bowen	$324\lambda^2$
Problem 11.	Gerald Roylance	$1302\lambda^2$
	Michael Coln	$1357\lambda^2$
	Guy Steele	$1404\lambda^2$
	Dave Otten	$1408\lambda^2$

BRIEF REVIEW OF IMPLEMENTATION:

- How do we get our designs implemented?
  GO THRU SLIDE SEQUENCE:

  PATTERN GENERATION, STEP & REPEAT TO GET MASKS, PROCESS, & PACKAGE

- KEY FIRST STEP IS PATTERN GENERATION. "FLASHING" BOXES ONTO A "PHOTOGRAPHIC PLATE", WITH A SORT OF "PROJECTOR"

- PATTERN GENERATOR DRIVEN BY A MINICOMPUTER, AND SIMPLY FLASHES SEQ OF RECTANGLES EACH HAVING [X, Y, h, w, a] VALUES, AS FED TO THE MINI BY A TAPE CONTAINING THE "PG FILE" FOR THE DESIGN, IN AN APPROP. FORMAT

- NOTE: This thing isn't particularly smart: If you had one simple cell repeated over and over, you'd still have a huge PG file.

- THE PG Files for project sets may contain Hundreds of Thousands of rectangle items. LOTS OF DATA.

- So that's an output file. We sure don't want to directly encode our designs in such a way. We need a way of encoding cells as "symbols" and then being able to call and repeat them in some reasonable way.

Symbolic Layout Languages:

~~what we're trying to describe is kind of like two dimensional (relatively) object code~~ ... We could use something like a macro-assembler where we define Macro's and then call them to insert code in place.

Let's Look at an informal example:

SLIDE OF SRCELL SHOW CELL ENCODING.

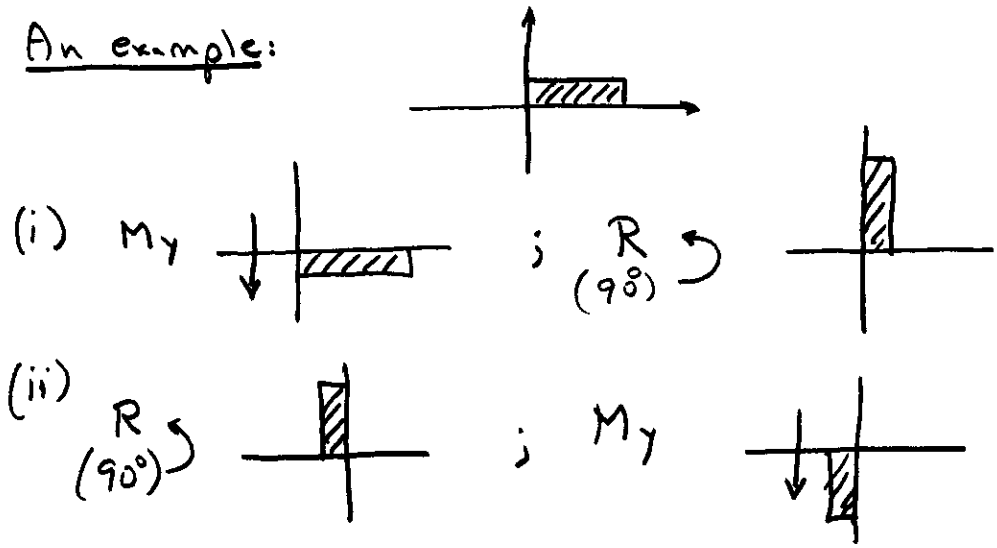
SLIDE OF CODE SHOW ITERATION, TRANF. TO GET AN ARRAY ...

Mention PLA code. **SHOW FIG 15 SLIDE**

Ah, but before we get carried away:

- There are dangers lurking that you might not anticipate:

An example:



- In general, sequences of MIRRORINGS, ROTATIONS, & TRANSLATIONS produce an overall effect dependent on the order.

- It is non-trivial therefore to specify the semantics of even the simplest layout languages: i.e. avoid continual misinterpretations of what particular encodings mean.

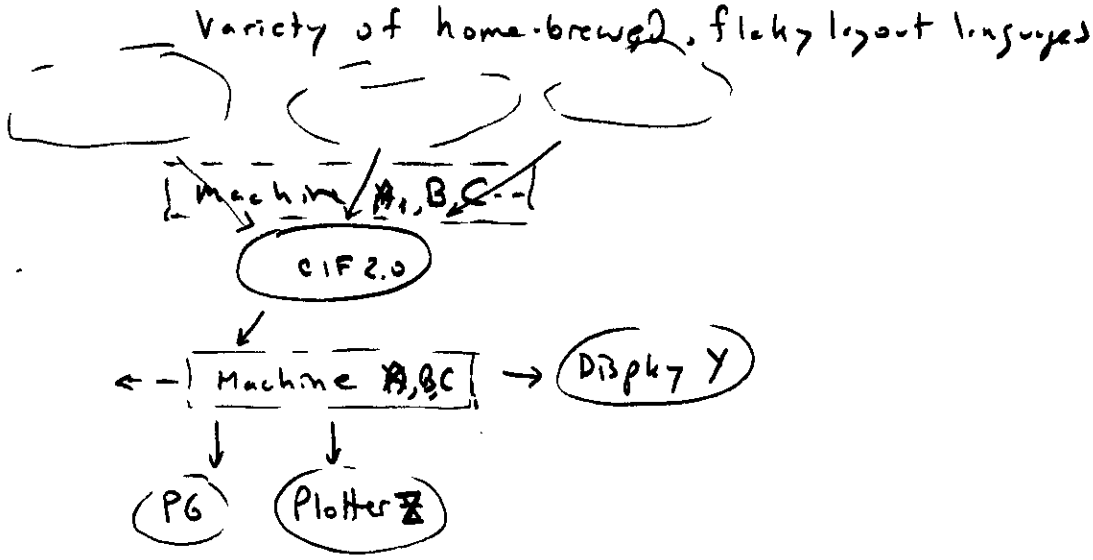
BACKGROUND:

There is no generally accepted, defined, documented layout language. We're about back in the "early 50's" phase of software history.

BUT:

There has been an effort to define a common design interchange language -- to avoid the problem of everybody having to keep writing new file conversion programs ...

So: Where are we?



MORE ABOUT THE INDUSTRY: MOST "ADVANCED" PLACES USE INTERACTIVE LAYOUT SYSTEMS. WE'LL SEE MORE ABOUT THAT LATER: BUT UNDERNEATH THOSE ARE DATA STRUCTURES AND OPERATIONS SOMETHING LIKE THOSE OF SYMBOLIC LAYOUT — ONLY MANIPULATED DIRECTLY BY COMPUTER. ~~CRAIG~~ ACTUALLY, THE SYMBOLIC LAYOUT APPROACH IF DONE AT A HIGH ENOUGH LEVEL, HAS ADVANTAGES NOT DIRECTLY AVAIL. IN STRAIGHT INTER-GRAPHICS SYSTEMS. (SYMBOLIC MANIP CAN BE VERY POWERFUL)

- WE'LL SEE AN EXAMPLE OF A RELATIVELY SOPHISTICATED SYMBOLIC LAYOUT LANGUAGE IN THE NEXT BOOK
- MEANTIME WE'LL GO BACK TO BASICS AND LEARN HOW TO CODE IN THE INTERMEDIATE FORM.
- This will be supported in the LAB: i.e. Text files of CIF2.0 code can be plotted on the plotters. Also, all the software necessary to generate/sort PG files is ready to go at PARC — CIF files thus can be shipped off for actual implementation
- HOWEVER, ANYONE WISHING TO EASE THEIR LAYOUT ~~ENCODING~~ TASK COULD WRITE A PREPROCESSOR TO TRANS. X INTO CIF

CIF 2.0 [A Human Readable Intermediate Object Code]

- On reading at this time, not necessary to go into details unless you wish to. We will use only a small subset - to get our projects going. We'll learn mainly by examples.

- **But first Correct ERRORS/HANDOUTS (2)**

The Subset: You should learn how to specify (interpret) in CIF2.0:

- Distances. Integer values in units of hundredths of a μm
- Coordinates: Right handed coord. system, incr. y up, increasing x right. Interpreted as front surface of chip. (not intermediate artifacts)
- Directions: Specified by 2 integers: a direction vector

  First is component along X, second along Y.
  Thus: (1 0) is in +X direction.

- BOXES: Box Length 25 Width 60 Center 80,40 Direction -20,20; (in X)

[Note: COMMANDS FOLLOWED BY ;
Note: CIF file is Sequence of Commands terminated by an END marker E]

- Recommend using shorter encoding:

BOX: B 25 60 80 40 -20 20;

[Note: Direction defaults to +X. We will use only Boxes at right ∠s. So don't need direction]

LAYER SPECIFICATION:

- A MODE IS SET WHICH APPLIES TO ALL SUBSEQUENT GEOMETRIC PRIMITIVES (BOXES) UNTIL SET AGAIN.

Layer ND;
(etc.)

or

L N D;
L N P;
L N C;
L N M;
L N I;
L N B;
L N G;

- **SYMBOLS:** This facility in CIF provides a means to greatly reduce the size of most intermediate form files compared to the PG file to be generated.

- The symbol facility in CIF is deliberately limited in order to avoid mushrooming difficulties of implementing programs that process CIF files. For example:

  > Symbols Have no parameters.
  > Calling a Symbol does not allow the symbol geometry to be scaled up or down.
  > There are no direct facilities for iteration. This is primarily due to the difficulty of defining a standard method of specifying iterations without introducing machine dependent computation problems.

- However, it is still possible to achieve much compaction by defining several levels of symbols:

  i.e. cell, row, double-row, array, etc.

DEFINING SYMBOLS:

- Precede the symbol geometry with a DS command;
  Follow " " " with a DF "

- `Definition Start #57 A/B = 100/; ---; Definition Finish;`

  > OR: `DS 57 100 1; ---; DF;`

  > The first argument is the symbol's identifying number.

  > The DS provides a way of scaling distances using literal values of A & B:

  As the form is read, each distance (position or size) is scaled to:

  $$(a * \text{distance})/b$$

  > Thus if the designer wished to use a grid of 1 micron, the symbol definition might cite distances in microns, and specify a=100, b=1 to convert to integers in units of 1/100ths μm. Or, use λ = 3μm, grid. Be careful: Integer distances only. This reduces the number of characters in files, and may improve their legibility. This isn't scaling. A symbol is defined with absolute distances.

- DS's may not NEST.

- However, DS's may contain CALL's of other symbols, which may in turn CALL other symbols.

- A Symbol must be defined before it is called.

  (Put symbol defs first)

CALLING SYMBOLS:

- The CALL command takes a specified symbol, and specifies transformations (TRANSLATE, MIRROR, ROTATE) to be applied to it to "place an instance" (instantiate) of the symbol at a particular location in a particular orientation.
- Call Symbol #57 Mirrored in X Rotated to -1,1 then Translated to 10,20.
- alternatively: C 57 M X R -1,1 T 10,20;
- NOTE: C 1 T500 0 MX adds 500 to the X coords, then mirrors in X.
  C 1 MX T500 0 Mirrors in X, Then adds 500.
- The order is important. Intuitively, each transformation is applied in sequence.
- SYMBOL CALLS MAY NEST. A symbol DEFINITION may contain a CALL of another symbol. However, no direct or indirect recursion (calling itself)
- WHEN CALLS NEST, it is necessary to "Concatenate" the effects of transformations specified in the sequence of CALLS.

- LAYER SETTINGS PRESERVED ACROSS SYMBOL CALLS & DEFS.
- WITHIN A SYMBOL DEF, LAYER MODE IS IMPLICITLY RESET BY THE DS, DF COMMANDS: DS to NULL (must be specified), DF to previous value.
- BUT I'd use simple procedure of closely coupling LAYER SPECIFICATIONS & THE ENTITIES THEY ARE FUNCTIONALLY ASSOCIATED WITH, TO AVOID ERRORS.

TRANSFORMATIONS: The primitive transformation are:

- T point: Translate the current symbol origin to this point (translate and place an instance of the symbol (origin) at this point)
- M X : Mirror in X: Multiply X coordinates by -1.
- M Y : Mirror in Y: Multiply Y coordinates by -1.
- R point: Rotate Symbol's X axis to this direction.

- Transformations are applied in sequence. However, don't need to do them all separately. Can compute the effect of a concatenated sequence as follows:

- Each point (X Y) is transformed to (X' Y') in the chip coordinate system by a 3x3 transformation matrix:

$$[X' \; Y' \; 1] = [X \; Y \; 1] \; T$$

- The transformation matrix T is simply the product of all the primitive transformation specified by the call: i.e. $T = T1 \; T2 \; T3$, etc.

- The primitive transformation matrices ~~are~~ obtained by using the following templates:

(Cont.)

Primitive Transformation Templates:

$$T\,a\,b \qquad T_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix}$$

$$MX \qquad T_n = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$MY \qquad T_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R\,a\,b \qquad T_n = \begin{bmatrix} a/c & b/c & 0 \\ -b/c & a/c & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{where } c = \sqrt{a^2+b^2}$$

Transformation of Direction Vectors: (X Y)

We Form the vector [X Y 0] and transform it by T into [x' y' 0].

The new direction vector is then simply (X' Y').

Read Section on transformations carefully.

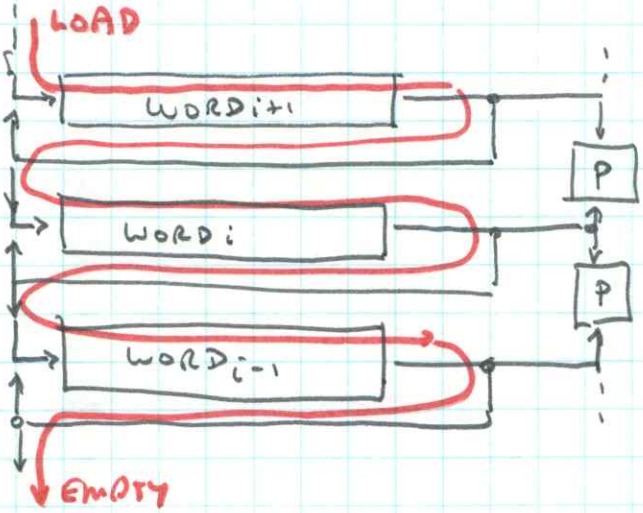
COMMENTS: Enclose in parentheses: (comment);

END COMMAND: E signals the end of the CIF file.

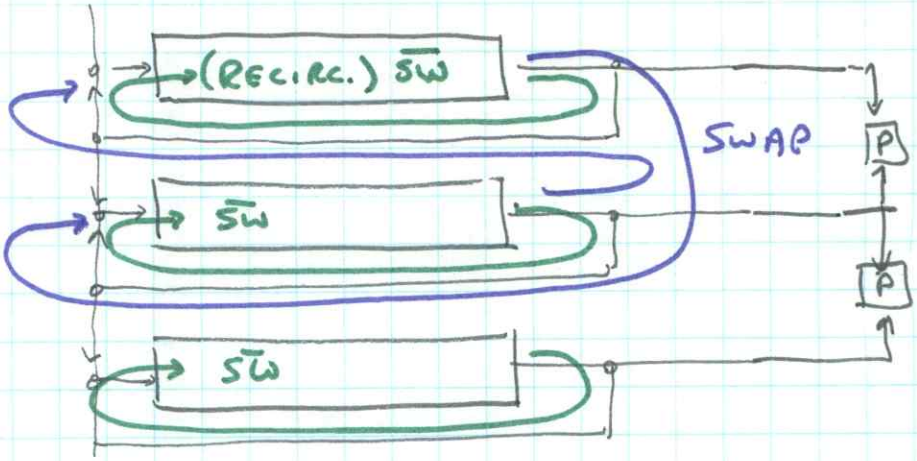
DISCUSS NEXT HW ASSIGNMENT:

- Problem 13 is a CIF coding exercise. Keep a copy of your solution. You'll need it for a lab exercise.
- THE SORTER SUBSYSTEM. Problem 14

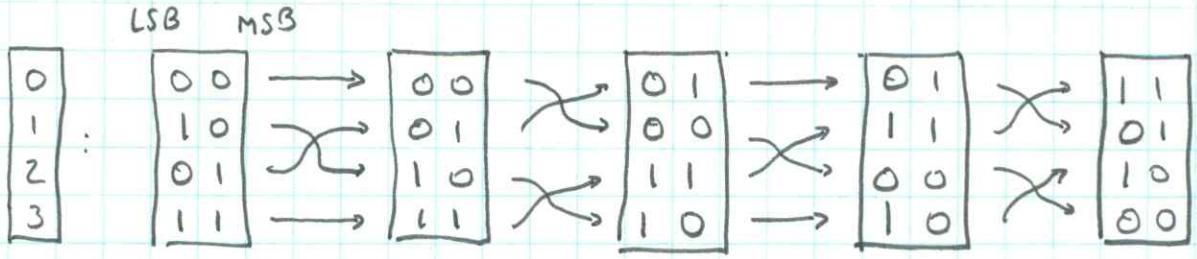
USE HW SET AS NOTES. DEVELOP STRUCTURE:



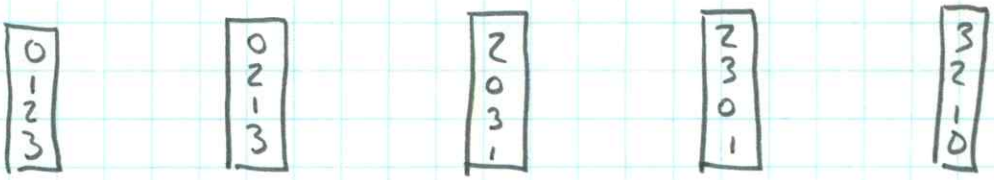
"A SMART MEMORY"



THE CONCEPT OF THE BUBBLE SORT: SUPPOSE LOADED:

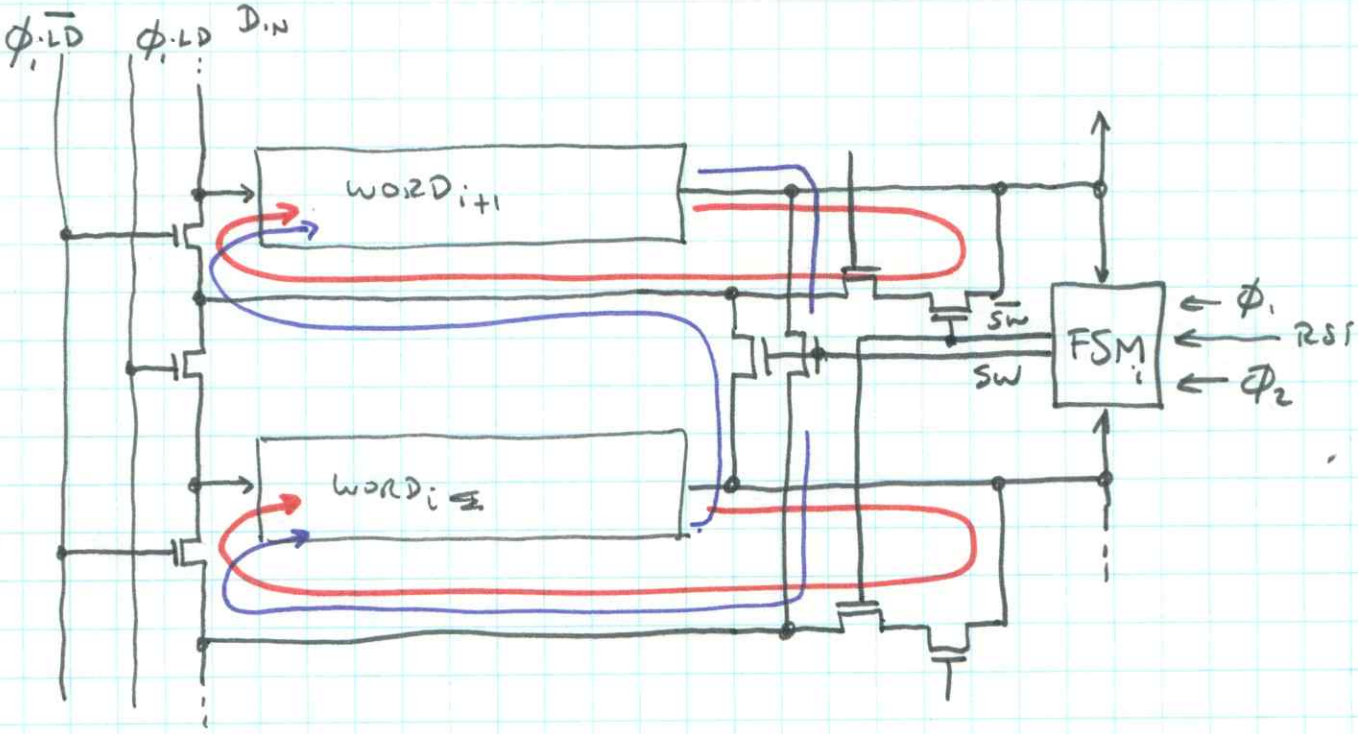


i.e:



ALGORITHM: SHIFT OUT AT RIGHT. START SWAPPING ADJ. PAIR OF ROWS IF DIGITS ARE DIFFERENT, AND LOWER = 1, UPPER = 0. COMPLETED IN N "CYCLES"

MORE DETAIL OF THE SWAPPING SWITCHES:



Think this FSM, in some detail.

But all you should do for this assignment is the circuit design of FSMs.

Be sure to carefully apply the 2-ϕ clocking methodology, doing the right things on the right phases.

- **PROJECT ASSIGNMENT #1**
  (See Assignment sheet)

- **PROJECT LAB STATUS/QUESTIONS**
  (Jon Allen)

6.978 LECTURE #10. OCT. 17, 1978

- TODAY: THE SORTER (CONT.). YIELD STATISTICS.

- SEMINARS: AM PLANNING TO HAVE SEVERAL CURR. ACT. RES. VISIT. THESE ARE OF GEN. INTEREST TO CLASS. WHERE POSS. WILL SCHED DURING CLASS HOURS. (OPEN)
- FIRST SEMINAR: DOUGLAS FAIRBAIRN, ON ICARUS & INT. LAY. SYS. HOPEFULLY TUES OCT 31. IF SO MIDTERM WILL BE MOVED TO THURS. NOV 2. (WILL KNOW NEXT TIME)
- MIDTERM: WORTH 1/3 OF FINAL. ON BASICS. THE HW ASSIGNED THIS TIME IS LAST BEFORE MIDTERM.

HW #4: SOME REALLY ELEGANT SOLNS. THOSE WHO GOT 10 or 10+ ARE WELL PREP. TO BEGIN PROJECTS.

MULTIBIT COMP 1 BIT VERT SLICE IN λ^2

~4600	Dean Brock	5280	Guy Steele
~4600	Andy Boughton	5300	Alan Snyder
5160	Randy Bryant	5328	Siu Ho Lam
5202	David Ottlen		

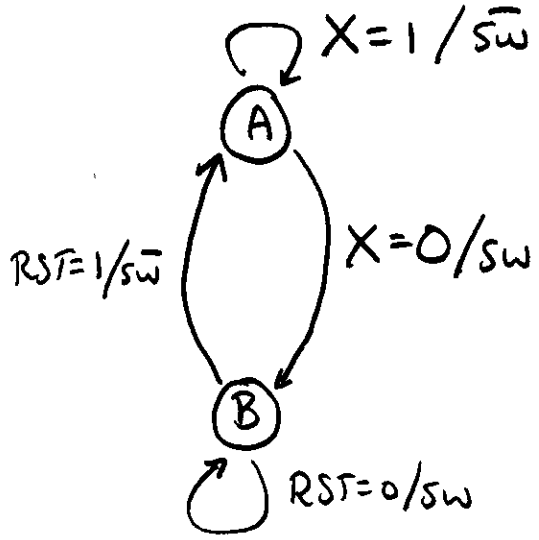
LAB: Glen Mirocker (alum) will open today ~3:30 to begin testability
Bill Henke: present more on CIF subset will use.

HW#5: The sorter is a good example of a major project ... The ideas we're covering will help you anticipate the work involved, the pitfalls you may encounter.

I tried to be clever on HW#5! Leaving some pitfalls for you to discover. Unfortunately I made a blunder which probably caused you unnecessary confusion: SLIDE -- ERROR

Continue with SORTER today. If you've made an effort to understand its function, you'll find the material today easy to follow and perhaps rather interesting!

- HAVE THE SORTER FIGURE (IN HW #5) ON SLIDE OR (ON BLACKBOARD.)
- USE WHITEBOARD(?) FOR SOME CIRCUIT FIGURES AND FOR CLARIFYING RECIRC VS LOADING.
- Review Sorter Fcn: LOAD, SORT.
- In HW #5, wanted to show use of very simple 2 state FSM ... showing that other than PLA might be best way to implement in simply cases. Unfortunately there was a superficial error in the transitions listed.
- Lets correct that error and look at a simple circuit implementing that FSM.

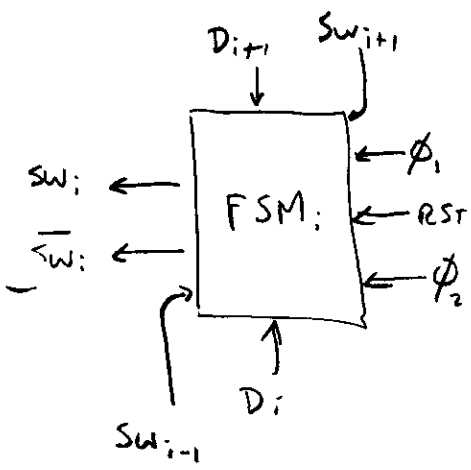


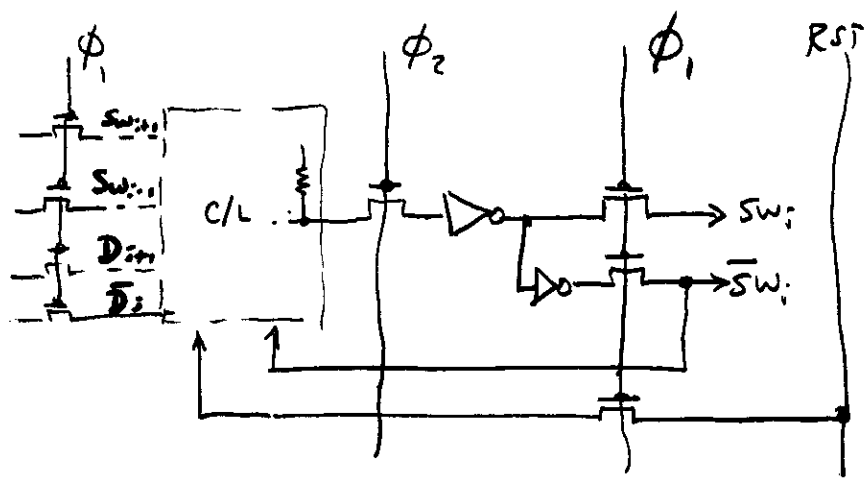
$$X = SW_{i+1} + SW_{i-1} + D_{i+1} + \overline{D_i} + RST$$

STATE A: NOT SWAPPING

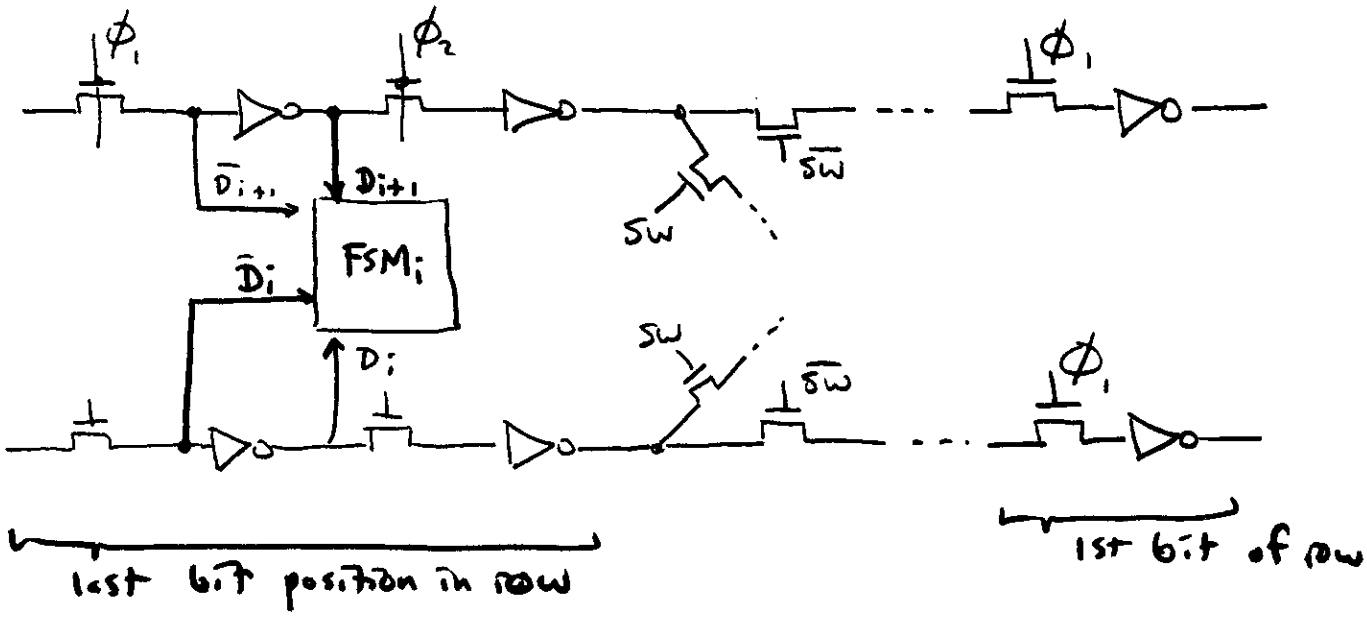
STATE B: SWAPPING

- FORM FOR INSERTION INTO THE SORTER:





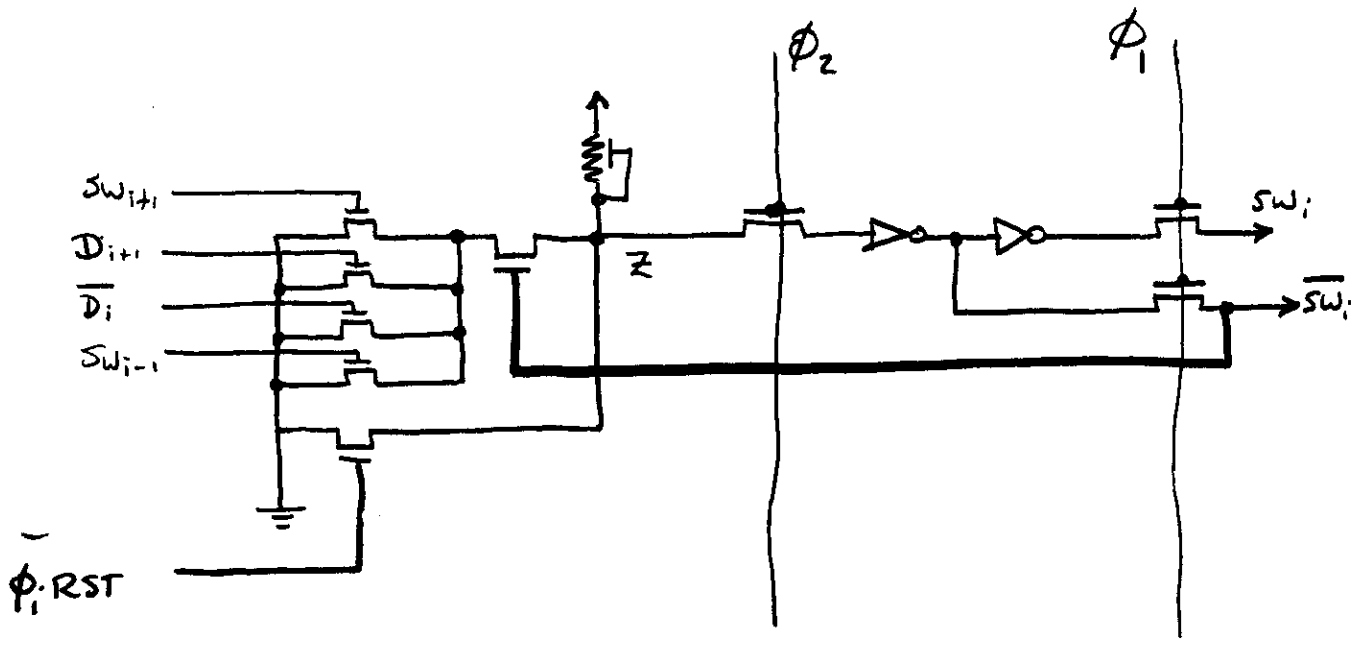
- ALL INPUTS SW_{i+1}, SW_{i-1}, D_{i+1}, D_i ARE TO BE TAKEN FROM POINTS "SET UP DURING ϕ_1": FOR EXAMPLE



- NOTE ALSO THAT THE OUTPUTS $\overline{SW_i}$, SW_i ARE SET UP INTO THE SWITCHES DURING THE FOLLOWING ϕ_1. ALSO, THEY INPUT ADJACENT FSM's DURING " " ϕ_1.

NOTE THAT A ROW SHOULD CONTAIN WORD$_i$ + 1 BITS

- HERE IS A SIMPLE CIRCUIT IMPLEMENTING FSM$_i$:



- EXPLANATION: > Reset PULLS DOWN Z, SETS OUTPUT TO $\overline{SW} = 1$.
  > THIS ENABLES NAND PART OF GATE.
  > THEN, IF RST OFF, AND ALL OTHER INPUTS = 0, Z IS PULLED UP AND SWAPPING IS INITIATED.
  > ONCE SW = 1, GATE IS DISABLED FROM PULLING DOWN AGAIN.

- NOTE: USE OF $\phi_2 - \phi_1$ CLOCK SCHEME MEANS WE DO HAVE TO WORRY ABOUT RELATIVE DELAYS CAUSING HAZARDS / RACES.

OK, THAT WAS THE HW PROBLEM: BUT WHAT WAS THE REAL PROBLEM? SHOULD WE JUMP IN AND START STICK DIAGRAMMING, & THEN BEGIN LAYOUT?

AH! NO! Because there is a fatal error in the algorithm!

We need a third state: (A) Not swapping (yet)
(B) Swapping
(C) Dont SWAP

Otherwise, we could have two rows in proper order, surrounded by other rows in proper order, that improperly begin swapping on first encountering a $\binom{0}{1}$ even though they previously encountered a $\binom{1}{0}$.

- So, we must add a state to "Remember Not To Swap" if we encounter $\overline{D}_{i+1} = 0$, $D_i = 0$.

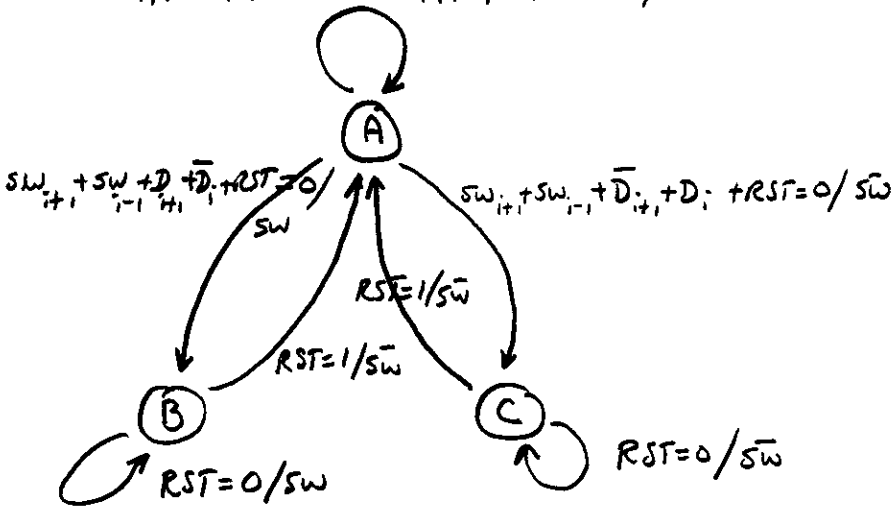
- A STATE DIAGRAM FOR THE CORRECT THREE STATE FSM:

and an extension of the previous circuit diagram to implement the three state FSM: is given in the next HW assignment.

$SW_{i+1} + SW_{i-1} + D_{i+1}D_i + \bar{D}_{i+1}\bar{D}_i + RST = 1/S\bar{W}$

STATE DIAGRAM:

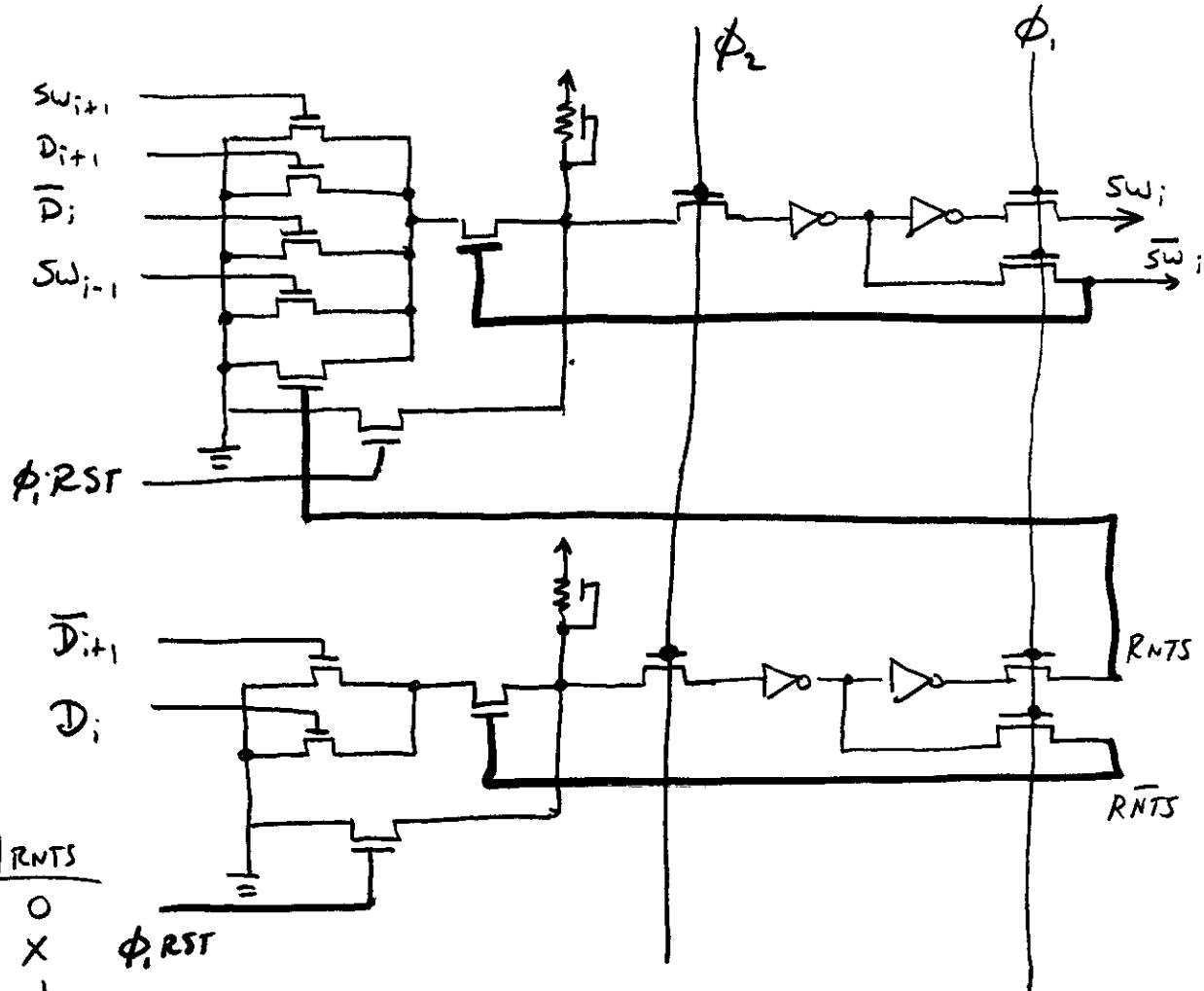
A: NOT YET SWAPPING/RESET
B: SWAPPING
C: REMEMBER NOT TO SWAP



i.e. If $SW_{i+1} + SW_{i-1} + RST = 1$, STAY IN (A)

If $SW_{i+1} + SW_{i-1} + RST = 0$, THEN IF $D_{i+1} = D_i$ STAY IN (A)
IF $D_{i+1} > D_i$ GO TO (B)
IF $D_{i+1} < D_i$ GO TO (C)

ONE POSSIBLE CIRCUIT IMPLEMENTING FSM:

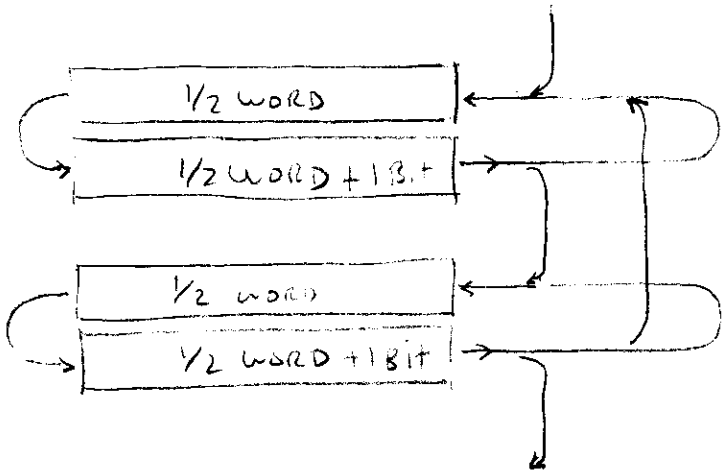


ST	SW	RNTS
A	0	0
B	1	X
C	0	1

- FOR HW #6, I'd LIKE TO FIND A GOOD, EASY TO LAYOUT CIRCUIT. NOTE THE PROBLEM OF THE PITCH OF ONE FSM VS THE SHIFT REGISTER.

  ALSO, DON'T WANT TO HAVE LONG LINES FOR FEEDBACK. MIGHT HAVE TO DRIVE THEM: SOLUTION:

  WRAP-AROUND EACH SR, BRINGING THE TWO ENDS TOGETHER. OF COURSE THIS STARTS TO KLUGE-UP THE SWAPPING OF FSM, BECAUSE NOW THE LOAD CIRCUITRY IS THERE ALSO:



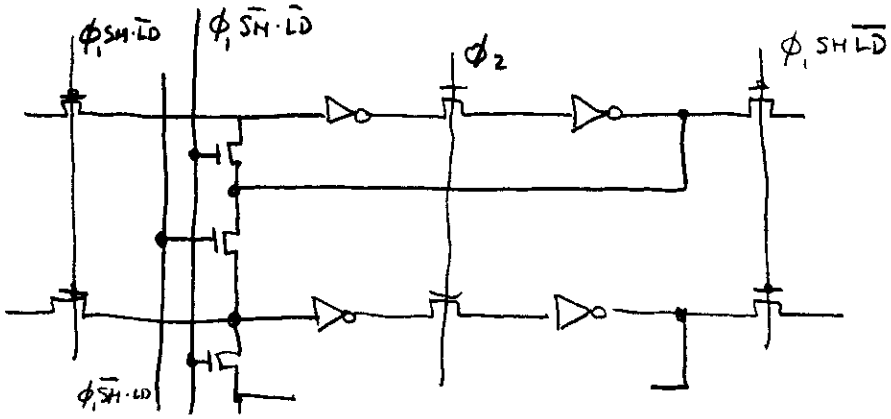
[CONSTRAINT: FSM; SHOULD LIE ON SR ROW PAIR PITCH.]

[MAYBE USE 8:1 MIN PULLDOWN SR HOWEVER]

- TO EXPLORE TENTATIVE STICK DIAGRAMS: STICK DIAGRAM ON A GRID ITS A GOOD WAY TO GET A FEELING FOR DENSITY. (LET RED OR GREEN RUN RIGHT BY BLUE, CTR.).

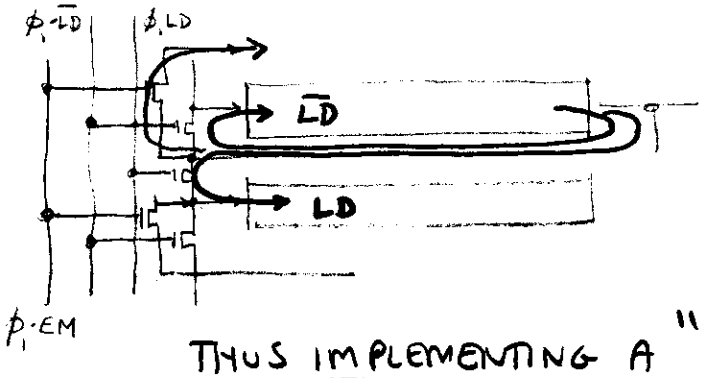
- FOR OPEN ENDED PROBLEM: TRY TO FIND A SIMPLER FSM; CIRCUIT, OR ONE THAT MIGHT LAYOUT MORE EASILY.

- OTHER EXTENSIONS: EASY TO MAKE //LOAD/EMPTY. PUT FOLLOWING INTO THE SR AT EVERY BIT:



[THIS MAY SPEED UP LOAD/EMPTY, BUT REALLY KLUDGES UP THE SR ARRAY]

- NOTE ALSO THAT IF WE ADD ANOTHER PATH AT THE "LEFT" (CONCEPTUALLY) AS FOLLOWS:



THUS IMPLEMENTING A "SORTING STACK"

- IN THIS CASE WE NEED TO HAVE A ZERO FILL-IN AT THE BOTTOM. NOTE THAT WE CAN BE SORTING AS WE FILL. THUS THE DATA CAN BE WITHDRAWN AS SOON AS THE STACK IS FULL!

- THUS THE STACK & TOP/BOTTOM SORTER BOTH HAVE A REP TIME OF 2N WORD SHIFTS. BUT THE STACK HAS A DELAY OF ONLY 2N WHILE THE TOP-BOTTOM HAS A DELAY OF 3N.

> ANOTHER SYSTEM LAYOUT PROBLEM WITH THE SORTER:

WE'D LIKE TO MAKE REALLY BIG SORTERS.

BUT THE THING IS TALL & SKINNY. HOW DO WE SNAKE IT AROUND ON THE CHIP? HOW DO WE ROUTE POWER AND GROUND? ETC. What about MULTIPLE CHIP SORTERS?

> WE'LL COME BACK TO THE SORTER IN AWHILE!

LECTURE #11. 19 OCTOBER 1978

- Course Sched / Project Sched:

> I'll hand out a revised schedule next week sometime. Probably no more significant homework. But total of ~4 project assignments:

26 OCT: Tent. Project Selection
9 NOV: Detailed Proj. Desc: Blk Diag, Alg., Stick Dag of subsections.
21 NOV: Tentative Layout
7 DEC: Project Report

> I'll make selection for inclusion by 28TH NOV. Files Sent ~5 December.

> Selection: Will exclude: Those progressing slowly
Any with any obvious defects in des/layout.
Those not carefully subsetted for testing.

- RATE OF PROG.
- LACK OF SIGN. ERR.
- QUAL./NOV. OF APPL. OR DES.
- EFF. SUBS. FOR TEST

Will favor: Those completing early, well checked.
Interesting design or appl. concepts
Well subsetted for first testing

- I can arrange later to get full-size checkplots of entire project layouts for selected projects.
- Estimate that about 1/2+ the projects will get into the chip set (*)

- Would like to have a team formed to take on tentative design study for an important subsystem that will def. be done in LSI here later on: The Chaosnet interface. Like ~4 or 5 students to participate. This COULD be a very exciting project ... but also might be a hard one. Tom Knight & Jack Holloway of the AI Lab will be able to meet with a group next week to describe it, give handout on current MSI version. See Me After Class if interested. Personal Computing; Coax Comm. net; ---

---

- MISC: Go over errors in Figs. in Ch. 5.

YIELD STATISTICS: HOW BIG SHOULD A CHIP BE?

- Empirically we find that as chip area increases, yield goes down; dramatically down

- There are no general, simple models. All we usually have is empirical data. But to get a feeling for the problem:

  [Suppose that defects are simple point defects, randomly scattered over the wafer, and that any single defect will kill a chip.]

  > Suppose there are N defects per unit area on average.

  Then the probable number falling within an area A is given (approx) by the Poisson probability: The prob that there are exactly n defects

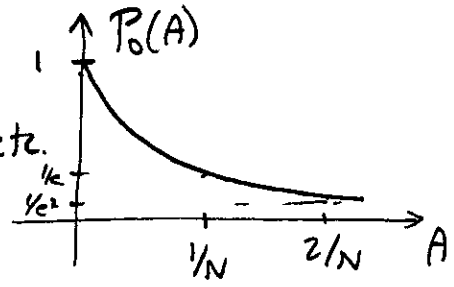
$$P_n(A) = \frac{(NA)^n}{n!} e^{-NA}$$

Data Analysis for Sci & Eng Meyer, Wiley Pub. '75

  In particular, $P_0(A) = e^{-NA}$ = probability of zero defects in an area A.

- Thus chips with $A >> \frac{1}{N}$ will almost never be found to work.

  chips with $A \sim \frac{1}{N}$ will have a yield of $\approx 37\%$, etc.



- In the real world of manufacturing, chip size thus is a highly complex matter, interacting with the cost of testing, the defect density, the part-type, etc., etc.

- Some Actual Values: Roughly, for big chips ~ 5mm x 5mm, yield will be ~ 10% or 50.

  A visual inspection finds 2/3+ of the defective chips. Testing finds the rest.

- Note: While our project chips are big, the projects are small, and yield will be very high, especially if inspected visually. If it doesn't work, it's probably a design error.

- We've Begun to Make analogies between integrated systems as hard patterned code and software systems as loaded code - i.e. the design, code generation processes, problems are really quite similar.

  BUT THERE IS ONE CRUCIAL DIFFERENCE: The integrated system compiler/loader doesn't produce exact identical copies of code: each chip may be slightly different due to defects!

  IMAGINE PROGRAMMING IN AN ENVIRONMENT where the loader Always tossed some random errors into your code! That's what we have to deal with in integrated systems.

- Maybe some clues about how to approach these issues could be gained by trying experiments with different ways of coping with such a software environment.

- Of course in big software systems, all the bugs are never out & people have learned some ways of structuring/testing to cope with complexity/errors. But in integrated systems you face this at the outset.

- AS WE SCALE DOWN TOWARDS VLSI, THESE ISSUES MAY BECOME VITAL. WE MAY BEGIN TO TRADE OFF FUNCTIONAL DENSITY FOR IMPROVED TESTABILITY... IF WE CAN GET ECONOMIC LEVERAGE IN SOME WAY -- i.e. MORE FCN FOR LESS OVERALL COST. COUNTING COST OF TESTING!

- EVEN BEFORE WE GET X100 DENSITY, WE CAN EXPLORE:
- WE CAN GET >X100 COMPONENTS INTEGRATED TOGETHER: HOW?
  USE THE WHOLE WAFER
- THIS WOULD NICELY "SIMULATE" VLSI (EXCEPT FOR T, POWER).
- **WHY ISN'T THIS DONE?** FIRST, I DON'T THINK ANYONE HAS REALLY TRIED. BRIGHT STUDENTS OF COMP. SCI/COMP. ARCH HAVEN'T STUDIED THE TECH UP TO NOW. MEANS OF IMPLEMENTING FULL WAFER DES. JUST NOW BEC. ACCESS (E-BEAM)

  Also, the industry doesn't take risks. It's always working on next years real product, and the process for the year after that.

- LETS TAKE A SPECIFIC EXAMPLE: YOU GUESSED IT: The Sorter!
  HOW DO WE COPE WITH DEFECTS. WITH TESTING:
  If we put many sorter sections on a wafer, each with some self-test circuitry, and with some type of bypassing interconnection form — we could **TEST THEM ALL IN PARALLEL, & STRING TOGETHER JUST THE GOOD ONES!**

  IDEAS EMERGING FROM SUCH EXPERIMENTAL ARCH/TESTING WORK COULD BE VERY IMPORTANT IN FUTURE VLSI.

- SO LETS THINK BIG! NOT ONLY WILL WE GET MORE DENSITY (MUCH MORE), AND FASTER DEVICES, AND LOWER POWER PER DEVICE. BUT WE CAN GET BIGGER CHIPS! IF WE COULD PARALLEL TEST/CONFIGURE

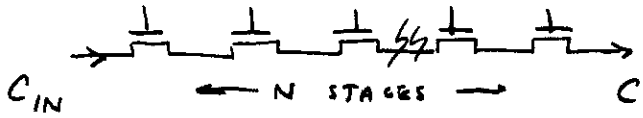
AH! BUT DON'T GET TOO CARRIED AWAY: BE CAREFUL: YOU'VE GOT TO GET LEVERAGE:

EXAMPLE: At a module yield ~ 1/2, For area A, if double A to get self-test + connection net, then yield → ~1/6.

So if start with 100, get only 50 (@ 2A per) of which only 1/6 th now work ~ 8 equiv chips. [Play with the number of sections]

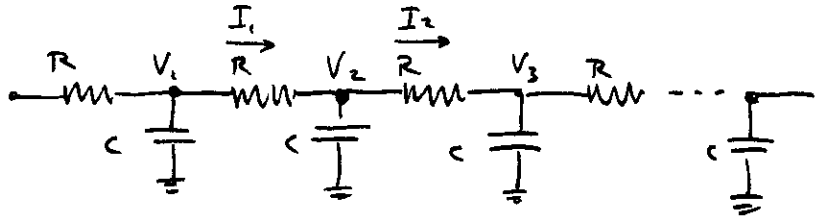
ANOTHER TOPIC: DELAYS IN PASS TRANSISTOR LOGIC

- WE KNOW THAT DELAYS IN INVERTING LOGIC GO AS $O(N)$
- BUT WHAT ABOUT "SWITCH ARRAYS" OR "CARRY CHAINS" IMPLEMENTED USING PASS TRANSISTORS? HOW DO THESE COMPARE?
- CONSIDER:



QUESTION: HOW DOES DELAY GO AS FCN OF N?

- APPROX EQUIV CKT:



[MIN R = res of on pass-trans, min C = min C_g. Parasitics increase these values, especially C.]

- Consider $V_2(t)$: $C\frac{dV_2}{dt} = I_1 - I_2 = \left[\frac{V_1 - V_2}{R}\right] - \left[\frac{V_2 - V_3}{R}\right]$
- If we considered R & C per unit length, then this reduces to differential form:

$$RC\frac{dV_2}{dt} = \frac{\Delta V_{1-2} - \Delta V_{2-3}}{\Delta x} = \frac{\Delta^2 V}{\Delta x^2}$$

ΔV_{1-2} = change in V for change in x; ΔV_{2-3} = change in V for change in x; $\frac{\Delta^2 V}{\Delta x^2}$ = change in $\frac{\Delta V}{\Delta x}$ for change in x

$$\boxed{RC\frac{dV}{dt} = \frac{d^2V}{dx^2}}$$

Where: R = res/length
C = cap/length

Slide

- This is the well known Diffusion Equation.

  Its solution are complex, but in general the time required for a transient to propagate in such a system is proportional to X^2.

- This can be seen qualitatively: Doubling N doubles both R & C in the system, multiplying an inherent time constant by 4.

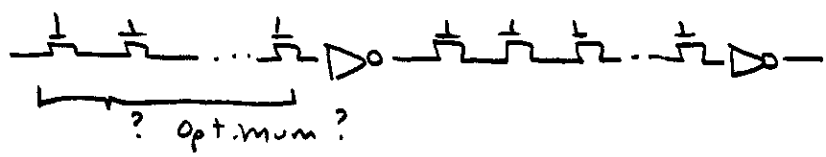
- ∴ One extra pass transistor: Adds little delay to a small chain

  But may add a lot of delay to an already large chain!

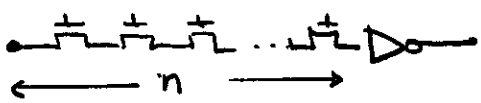
- WHAT TO DO! Aha! **Break up long chain into sections Put inverters in between**

Accept/Trade the added fixed delay for big reduction in the overall total delay!

- HOW OFTEN?



- SUPPOSE HAVE:



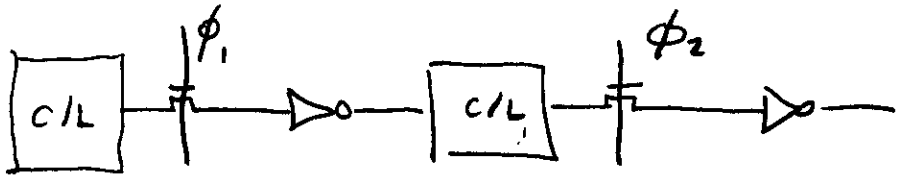
- TOTAL DELAY $\cong RCn^2 + T_{inv}$
- AVERAGE DELAY/STAGE $\cdot RCn + T_{inv}/n = f(n)$
  we find that minimizing f(n), the min occurs when: $\boxed{RCn^2 \sim T_{inv}}$ So thats how to choose n
- Right now: $T_{min}\, n^2 \sim T_{inv} = \frac{9}{2} \cdot 2\, T_{min}$, ∴ $n^2 \sim 9$

  and $\boxed{n \sim 3}$. We'll usually use $n \sim 4$
- IN CLOCKED STAGES: (IT'S AN ARCHIT. QUESTION. CAN USE UP TO THE MAX DELAY ON A Φ WITHOUT PENALTY)

TRANSIT TIMES & CLOCK PERIODS

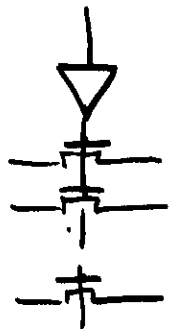
- What is the shortest clock period we might be able to use in 1978 for synch. digital system in nMOS?

Consider:
R-R Transfer stage



- NORMALLY THE C/L + CLOCK GATE: IN PASS TRANSISTORS.
- Level Restoring will be inverters: $k \sim 8$. Max dly $\sim 8\tau$
- NORMALLY THESE will be MATCHED:
  C/L + CLOCK: $\sim 8\tau$, INV. MAX $\sim 8\tau$
- THUS TOTAL OF $\sim 16\tau$ per phase $\times 2$ for STRAYS.
- THUS $\sim 30\tau$ per clock phase.

> BUT CTL lines typically drive 10 to 30 pass transistors.



- Even if SUPER BUFFERS, if $Y \sim 30$, Driver delay is $\sim 9\tau$
- MUST ADD $\sim 8\tau$ to operate drivers

> $\therefore$ Total Clk $\phi \sim 50\tau$, $\therefore \boxed{T \sim 100\tau}$

- In 1978: $0.3 < \tau < 1.0$ ns so $T \sim 30$ to 100 ns.
  (For compact synch. d.j. syst)

However, if some signals must run for long distances, drive very large loads, or are gated by longer pass-T chains, the corresponding ϕ_i and thus T will be longer.

For example in OM2, ϕ is 50τ but ϕ_2 has a delay of 100τ for each 4 bit ALU block, so $\phi_2 \sim 400\tau$.
$\therefore \boxed{T(OM2) \sim 450\tau}$ or (135 ns to 450 ns = few MIPS)

(*) UPDATE ON RULES OF THE GAME:

Area Estimates: Negotiating for Area (AKA "SPACE WAR")

36 people will participate in projects

Possibly: 4 - 3 pers, 6 - 2 pers, 12 - 1 pers = 22 tot. projects.

likely ~ $1/2^{+}$ to $2/3^{-}$ will get done/look o.k. to go on chip set.

So probably: ≈ 12 - 15 projects will go on chip.

Chip set likely to be 6 mm x 10 mm (2 chip types).

Area ~ 60 mm^2. Probably ~ 10 mm^2 will go for scribe lines, alignment marks, test patterns, etc. Maybe a bit more for "packing inefficiency".

So: maybe 45 mm^2 / 15 projects = 3 mm^2 / project.

So: A large project will be 2 x 2 mm = 4 mm^2.

If we put many of these, or some bigger ones, they'll have to be compensated for with some little projects.

Try for 2 mm^2	Let me know early if you'll want to go over 4 mm^2

> USE SYMBOL # 100 or greater
> The Lib. will use symbols 1-99

[redacted]: Priorities? CIF Cell?

6.978. LECTURE # 12. 24 OCTOBER '78

- Handouts: HW, CIF Tran Guide, Guide to LSI Implementation
- Announce: Lab is running. Software & Plotters working.
  2 Lab assistants: Charlie Davis, Philip Ngai.
  Rm 36-561. Open ~3 to ~8+, Mon-Fri.
  PROJECT IDEAS
- TODAY: Design & Implementation: More on how actually done at present. How we'll do it. Details that aren't in Text.
- Impl. Guide cont. much of this mat'l. Background: Was prep. for instr., TA's, Lab Assistants, for courses starting this year ---. But there is a lot of gen. useful info, so I had copies made for all of you. ---
- So, today we'll skim thru a wide range of practical topics to get feeling for how things are really done now. Mostly we'll skim thru the Impl. Guidebook.
- Some of this will be useful reference mat'l for projects
  [For example, there is a cell library in CIF in GuideBook]
- All of this will set stage for looking ahead into the future: What would we do differently? ---
- In Later Lectures: We'll study future patt/fab techniques which'll enable higher density, & the effects of this ---

BUT THERE IS MORE TO THE COMING CHANGE: We'll consider at least 3 areas:

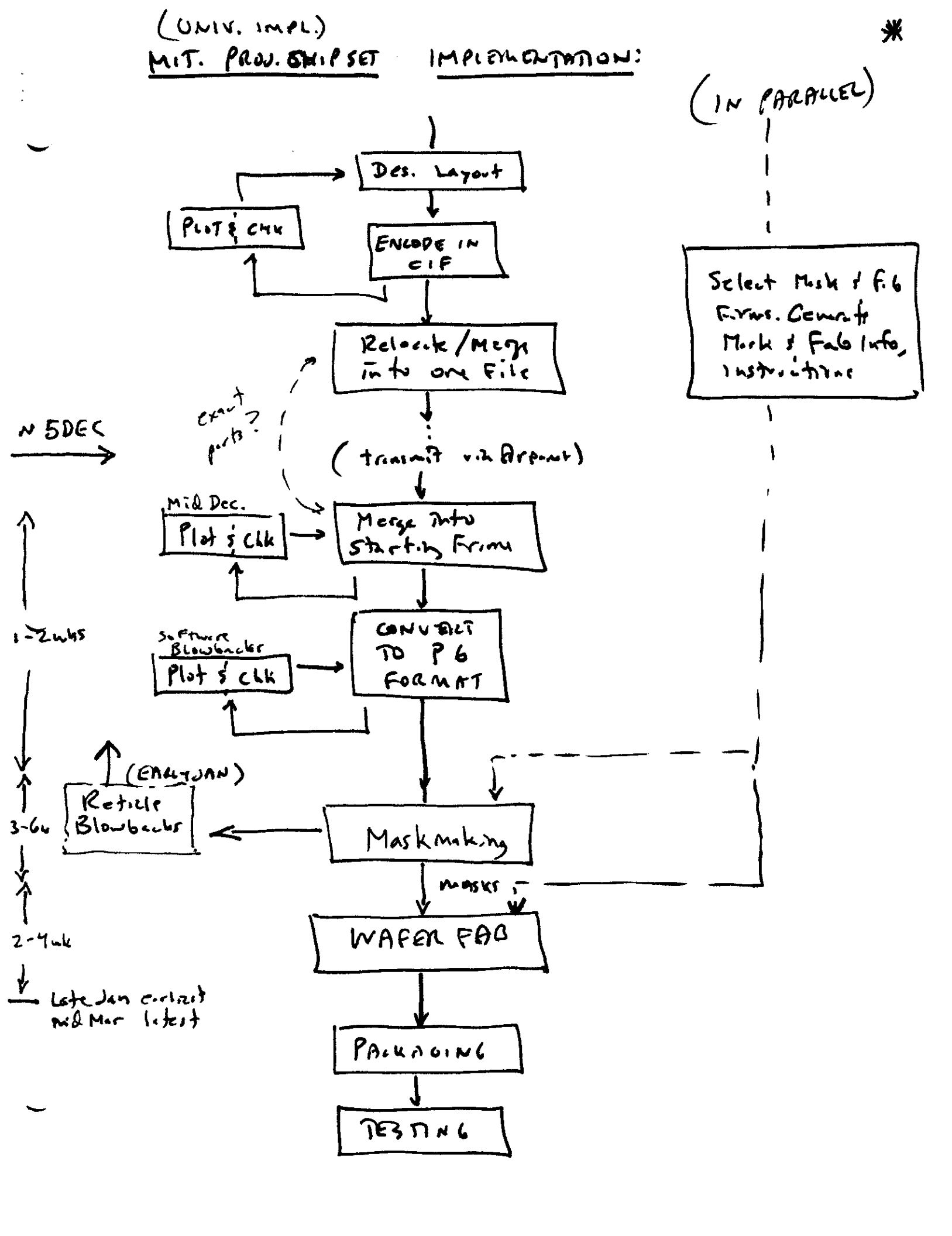
1. IMP. DES AIDS: FASTER/EASIER DESIGN
2. MORE PROC. AUTO: FASTER IMPLEMENTATION
3. HIGHER DENS.: SCALING EFFECTS

- OVERVIEW/REVIEW OF DES & IMP (SLIDES)
  "ARTIFACT FLOW"
- NOW, HOW ARE WE ACTUALLY GOING TO DO ALL THIS?
- DRAW FLOWCHART ON BOARD *

MANY OF THE ANSWERS, IN DETAIL, ARE IN THE GUIDEBOOK.

(UNIV. IMPL.)

MIT. PROJ. CHIPSET IMPLEMENTATION:



- I'd like to go thru the Guidebook in "LOGICAL ORDER"

  Skimming some sections, just indicating contents, Going into more detail on other sections.

  All this is to raise your awareness about the range of topics. You'll know where to look to begin to get answers. Also recommend the Sep'77 SCI AM issue for more "LORE". But it and most other ref. are written to describe what "Someone else" does rther than teach you how to do it.

- TOPICS IN LOGICAL ORDER:
  - > DESIGN AIDS: BASIC IDEAS, PRAC THINGS YOU MIGHT USE.
  - > PRESENT MASK & FAB PROCEDURES
  - > MULTI-PROJECT CHIP & STARTING FRAME
  - > INTERFACING MASK & FAB FIRMS
  - > WHEN WAFERS COME BACK: PACKAGING
  - > TESTING: ELECTRICAL, FUNCTIONAL

- DESIGN AIDS: SKIM THRU, IDENTIFY A FEW SECTIONS:

- INTRO SECT (P 5-8) S.TRIM., "AUTOMATED DESIGN AIDS"
  - > Desc. Basic Des.Sys (CIF → Plotter as here)
  - > Symb. Layout > Inter. Graphics
  - > Ideas regarding Future ADV. Des. Systems

- More Detail on Symbolic Layout: (p 9-15) M. Stone. Discussion of what symb. layout languages might do. Later, a SPEC of a proposed language (ICLIC) is included (p. 79-99)

- Design Rule Checking: Short but interesting discussion (p 15-18) by w. wilner. Not trivial problem. Industrial programs exist purporting to "check design rules". Note that till now no formal description of any design rules has existed. ALL AD HOC

  We will include in publ. text a recently completed formal desc. by Irene Buchanan of Edinburgh of our design rules.

In long run: Prob best to Inst. Layout from sticks as few rules, R.T. CHEN

One Design Aid You'll Find Useful: A Cell Library

(p 100-144) Bob Baldwin, Dick Lyon

- A bit about managing a Cell Library. 100-101
- Table of Contents 102-103

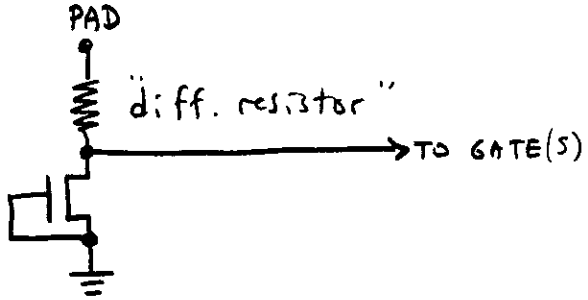
- **SOME USEFUL CELLS:** Pads, PLA cells, Logic/Arithmetic, & STARTING FRAME (disc. later)

- **PADS:** Input w lightning Arrestor, Output with Driver, VDD, GND, Blank. p 104-111. SLIDES

Cleverly Designed so they all fit together on same pitch. VDD runs along top & sides. GND along bottom.

EXAMPLE: **PADIN:**

Normally, FET is OFF, so circuit behaves as resistor connected to pad.



But, if Voltage becomes large (i.e. Pad struck by lightning) then FET will punch trough, current will Flow dropping the over voltage across the resistor.

- Punch through occurs at a lower voltage than gate oxide breakdown, so gates are protected.

Example: **PADOUT:** Inverter Chain: 3 stages,

each larger than preceding: Input 4x minimum, Next: (superbuffers) 4x preceding, Next: wide Enh. Mode FETS 8 times preceding.

Traded off max speed (xe) for simplicity, less area, lower power than the OM drivers/pads.

- **PADS: SIZE:** 126 μm (114 if overglassed).

BIG enough to be easy to Bond.

ELECTRICAL SIMULATION: Another Useful Design Aid.

p19-25 DICK LYON

- Very common use of computers in traditional unstructured IC design.
- However, for a small % of cells in our structured designs, electrical simulation can be very useful - to reduce delays and improve performance by varying design parameters & observing effects. EXAMPLES LONGEST CHAIN OF PASS-TR, NODE WITH HIGHEST FANOUT, CONTROL DRIVERS, OUTPUT DRIVERS.
- TWO WIDELY USED SIMULATORS: MSINC (SIMAG.) SPICE (BERKELEY) Tend to suffer from univ. Batch "card oriented" culture of design. Data Prep. is awkward. User interface poor. Not easily integratable in design systems.
- EXAMPLE: In Guidebook: Dick Lyon presents worked out Example of PADOUT OUTPUT DRIVER (SLIDE OF LAYOUT)
  - > Circuit Diagram of PADOUT is given (SLIDE) containing node #s, element names, W & L values for FETs.
  - > SPICE INPUT FILE, CALLED A "DECK", (SLIDE) indicates sorts of parameters required, EX: AS = area of source, AD = Area of drain. [Note: 1st inv not part of PADOUT. SIGNAL SOURCE 3.5v 20MHz SQ WAVE with 2 nsec Rise/falls + inverter combined external]
  - > OUTPUT PRODUCED IN LINE PRINTER FORM. (SLIDE) Node 1 to node 6 delay ~ 11 nsec @ T = .24 nsec [certainly optimistic] But helped very much to reduce delay relative to T.
  - > AT best such simul. only as good as their FET models & input parameters.

- GREAT NEED FOR & GEN. ABSENCE OF HIGHER LEVEL SIM.
  - > LOGIC TRANSFER FUN TESTS > R-R TRANS. SYS SIMULATION.
- All should be within same integrated design system, operating off same or machine generated variants of same data base, with att. paid to user interface. [BUT TRADE-OFFS VS IMPL TIME]
- Const. of such int. des syst. important requirement if [illegible] design is to be made more like "programming". [i.e. DON'T USUALLY SIM. PROGRAMS]

PRESENT MASK & FAB PROCEDURES
& HOW THEY AFFECT OUR DESIGN FILES / PREPARATIONS.

- (p 32-37) Contain Some More Details on Mskmaking.

Most Mask houses use GCA-Mann PG & Photorepeat Equip. The photorepeaters yield a limit to field exposable ~1cm²

Thus, we normally make 10X reticles, and at most cover a 1x1 cm area on the wafer with these.

Now the package we use is std.d 40 pin - which will hold ~ 6mm x 6mm chip. So if use 1cmx1cm, must be able to scribe within it. (more later).

We must provide artifacts for:

> alignment marks used in fab sequence
> CD's used in mask making (lines of known width)
> scribe lines
> maybe "fiducials" used in photorepeating
> maybe "parity marks" used in photorepeating

We must also provide information regarding plate polarities, dep. on whether neg or pos resists used in fab line. etc. etc. etc.

- (p28-31) Contain More Details on Process

In particular: Quite a bit now precedes the first "oxide patterning" in our previous simple model of the process. Now, a so-called "Channel Stop" region is ion-implanted (pt) under areas where thick oxide is to be grown (rather than cut). This means we can use 2X DIFF-DIFF. It also cuts down on parasitic C's. (self-aligned)

But: the overall effect of the process is the same as that we've previously studied.

Note also: The substrate will be grounded. In packaging we'll use conducting epoxy to glue the chips into their package. Then connect gnd lead to package ground.

- ON: Now how do we deal with all these artifacts and procedures, etc. If each designer had to do all this, it would be an enormous overhead per design.
- Solution: Share the overhead via the Multi-Project Chip
- (CH 4 in text + p 51-67) TALK ABOUT BACKGROUND.

  SHOW SOME SLIDES OF PAST MPCS
- Only the "Coordinator(s)" have to know all the details. Individual designers just supply their design files.

- THE STARTING FRAME: If we exclude all the actual projects, we're left with the starting frame: all those artifacts which convey the projects thru MASK, FAB, & PACKAGING, & ELECTRICAL TESTING.

  > alignment marks > CDs > Fiducials > parity marks
  > scribe lines > electrical test patterns.

> SHOW SLIDES & DISCUSS

(see also cell library for CIF code of some of these artifacts)

> SHOW ALSO BLOWBACKS ON STAND

(PLOTS ALSO IN LAB)

INTERACTING WITH MASK & FAB FIRMS

- Assuming we will run GCA Mann PG, we must convert our CIF code to MANN PG format. This format is described on p. 74-78
- We must also provide the MASK firm with quite a bit of info, some of which is FAB line dependent. These issues discussed on p 38-40 and particularly,

  on p. 68, 69: [Copy of Specs sent to Mask house for the project set discussed in Guidebook.]

- **Discuss this SPEC sheet.**
- An Index of Manufacturers is Given p 72-73. Please use judgement here. Probably shouldn't contact unless you've got design file for project chip plus MONEY in hand.
- A large project set will cost ≈ 6k for masks, (mostly PG time), and ≈ 2k for FAB of ≈ 20 wafers. ~20 is a minimum run. If run more, less /wafer.
- We'll get enough chips so that every participant will have many chips to bond to just their project.
- Time: 3-6 wks masks, 3-4 fab, 1-4 misc.

MORE ABOUT TIME LATER: WE'VE GOT TO STREAMLINE ALL THIS, MAKE MORE LIKE PROGRAMMING. IF EVERYTHING WENT AS FAST AS POSSIBLE, WITHOUT QUEUES: DESFILE-CHIPS IN ≈ 3 days.

[ONE OF THE VERY IMPORTANT CHANGES COMING IS THE GENERAL AVAILABILITY OF FAST TURNAROUND IMPLEMENTATION]

6.978 LECTURE #13. 26 OCTOBER '78.

- TODAY: IMPLEMENTATION (CONT.); THE STORED PROGRAM COMPUTER

- HANDOUT: Project Assignment Schedule:
  26 OCT: Select; 9 NOV. Descr; 21 Nov. Layout; 7 Dec. Report.
  MY Selection: 21 - 28 NOV.; Files Sent ~ 5 Dec.

POLL ON PROJECT SELECTION

- CONTINUE WITH A BIT MORE re Impl: THE PROCESS:
- BASIC NMOS Process desc. in text. (SLIDE)

- Process we'll use will have a # of steps to reach the first patterned oxide: these are described in GUIDE BOOK

- Basic idea is to get a p^+ region underneath the Thick field oxide region: (SLIDE)

  This reduces parasitic C's and allows DIFF-DIFF to be 2λ
- WE WILL USE GNDed SUBSTRATE (EXPLAIN)

- INTERACTING WITH MASK/FAB FIRMS: (p.72-73)

  An index of firms is listed in the guidebook. Note, this was intended primarily for instructors & those actually running project chips (i.e. those with money).

  Costs: Making Masks for large proj. set ~ 6 K. Depends primarily on the # of flashes (P.G. Time).
  Min run of Wafers: ~ 2K to fab ~ 10 to 20.

  [Note that Maskmaking generally takes longer, more expens.]

  Specs: Sample SPEC sheet for project given on p 68-69

WHEN WAFERS COME BACK:

There's a discussion of dicing, chip mounting, wire bonding on p 40-42.

TESTING: Were concerned with Two different types:

(a) Electrical Testing: Could be done by probing wafers or by bonding up test patterns in packaged chip:

Special Patterns are included in the starting Grome to allow us to measure τ and also the $R/\square$ of the various layers, to extract the MOSFET characteristics, and determine the quality of the process by measuring resistances of long chains of contacts, etc.

A number of such patterns are illustrated and described in p 58-62 by Rick Davies

Show Slides

(b) Functional Testing:

Assuming the process worked, and we've measured τ, and assuming that projects are small and thus have high yield,

We now will perform functional tests on our projects. Procedures for this are discussed in p 43-50 by Peter Dobrowolski.

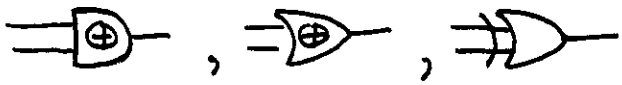
Show Slides and Discuss.

[COULD USE ANY OF THE CURRENTLY POPULAR MICROPROCESSOR DEVELOPMENT KITS. ESPECIALLY IF BUILT UP SOME OUTBOARD HARDWARE]

IF TIME: [DO AT END]

ANOTHER TOPIC WE'VE USED MOSTLY NAND, NOR, INVERT GATES.

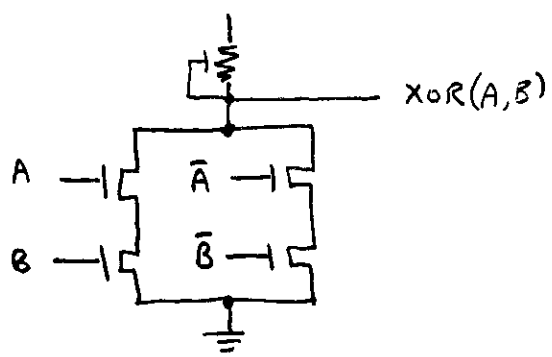
ANOTHER IMPORTANT GATE EASILY IMPL. IN MOS IS THE **XOR** GATE "EXCLUSIVE-OR"

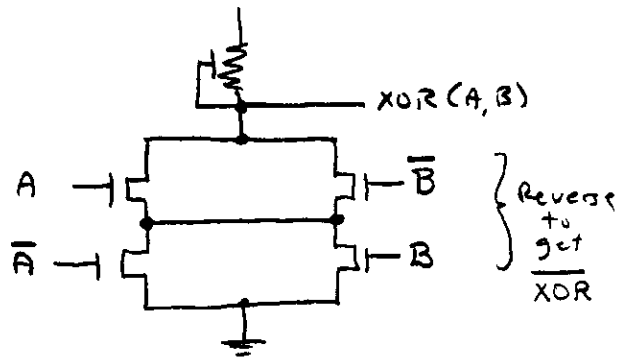


	(XOR)	($\overline{\text{XOR}}$)
A B	$A \oplus B$	$\overline{A \oplus B}$
0 0	0	1
0 1	1	0
1 0	1	0
1 1	0	1

XOR $\Leftrightarrow$ "MOD 2 SUM"

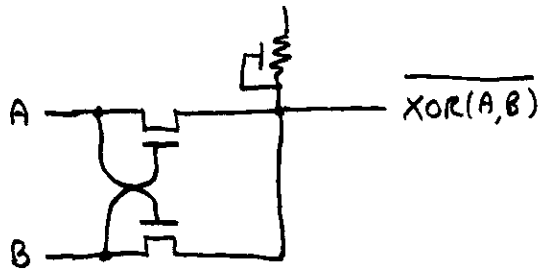
AS IN MULTI COMPARATOR:





ANOTHER VERY SIMPLE CIRCUIT, REQUIRING ONLY A & B (NOT COMPL)

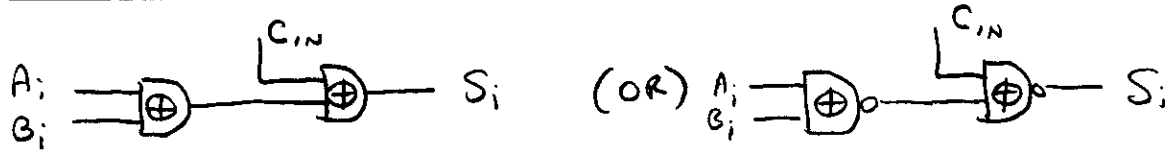
IS:



Describe fcn:
If A = B = 1, OUT = 1
A = B = 0, OUT = 1
If $A \neq B$, then 'ON' FET connects to zero input.

(NOTE: CANNOT BE FED BY CLOCKED PASS-TRANSISTORS)

EXAMPLE USE: IN ADDERS:



[NOTING: $(A \oplus B) = \bar{A} \oplus \bar{B}$]

THE STORED PROGRAM MACHINE:

- CH 5 & 6 Desc. an LSI comp.sys. des & impl. at Caltech.
- Provides many ex. of LSI circuit & subsystem design. Partitioning of 2 chips of system: DATA PATH, CONT. (SLIDES)
- In later lectures I'll be describing this system in some more detail.
- It is architectually a general purp, stored program computer, using microprogrammed control.
- So that we can share a common terminology, and really understand the details of this system, I'd like today to review the basic ideas of the stored program computer --- the classical general purpose computer sometimes referred to as the von Neuman machine (as a concept).
- What is a general purpose, St. prog. computer? [First ½ CH6]

  i.e., what are the key ideas, from which we synthesize and instantiate such machines.

[> By the way, that is an often embarrassing question to ask of "computer architects". Try it sometime!]

- Consider the OM Data Path: (SLIDE)

  It is claimed that this regular looking structure can perform a rich variety of operations on data stored within it.

  How can we visualize this — I tend to think of it using a piano analogy --- control lines as keys that are struck --- sequence of notes and chords can over time build up a very complex, abstract, piece of music.

  DATA FLOW vs CONTROL ... So the data chip is only at best ½ a computer. We must generate the control sequence somehow.

- **WHAT'S THE SIMPLEST FORM OF CTL. SEQUENCER:**

Fig 1 A Finite State Machine. Here, no inputs to FSM just runs thru a fixed seq of o-puts indep of act. in Data Path. Could be used for impl. a digital filter - data in at left, fixed set of op's, data out at right.

- **Enhance this a bit:** Fig 2

To make control sequencing possibly be fcn of some event in data path, some LOG FCN of data called FLAGS are fed as inputs to the FSM.

Note: These diagrams "cover" large sets of possible FSM's, and overall structures. They are meant to clarify the KEY IDEAS which enhance functional capability as we move up thru a hierarchy to the SPM.

Any one of these can "cover" machines of great complexity in detail. But the key ideas are few in number

Note: For simplicity, will not bother to show the clocks and PLA regs. Assume everything is Synchronous

- **Enhance Further:** Fig. 3

Fig 2 is quite general --- but we can make improvements such as adding or dedicating a register to hold the flags, loaded by output from the FSM.

Thus, the flags can be used as control inputs for many cycles after their generation.

> A basic limitation however is the small amount of information provided by the few flags generated by the data path operations.

- THE STORED PROG. MACH. (FIG 4a)

A very powerful & general arrangement is shown in Fig 4a.

FSM sequencing not only controlled by last state & flags but also by data coming from a memory. This provides a completely new dimension of possibilities:

- The fundamental idea is:

Rather than have the FSM perform one predefined operation (no matter how complex), we design it to perform any of a set of predefined operations

Called the "Machine instruction set".

The Mach. Inst. Set is carefully defined so that with the (CONT, DATAPATH, MEM) we can mechanize any of a number of different algorithms of interest to a number of diff users.

These algorithms are ~~implemented~~ encoded as programs composed of sequences of machine instructions loaded into the MEM.

Programs operate on data also located in the memory.

- The Machine functions as follows

> One register in DP is selected to hold pointer into the program. Call this the PROGRAM COUNTER (PC).

> In Particular FSM state, called Fetch Nx Inst (FNI)

the FSM causes the memory to be read at location PC.

The resulting fetched inst. then places the FSM into first state of sequence to execute that instruction type.

- The FSM sequences thru # of states to execute that inst., at some point incrementing / calc. next PC value, & finally returning to the FNI state.

- Problem with Fig 4a: Most steps of INST need as FSM input some details of inst. or its encoded fields. We'll get more effective use of PLA if we dedicate a register to hold the instruction:

- **FIG 4b** In Fig 4b, an Inst. Reg. (IR) holds the inst. fetched during FNI from mem locn PC.

- NOW LETS FURTHER STRUCTURE THINGS BY NAMING the STAGES of EXEC. OF INSTRUCTIONS:

  Suppose have mach inst set incl. ALU ops, BR ops, MEM ops. & Data Path like OM. What must controller do to execute typical machine instructions?

  Typically Six Stages: **ON BOARD**

  ① Fetch Next Inst — inst at PC fetched into IR

  ② Decode Inst — "branch to start. state" for OP type

  ③ Fetch Operands — state seq. to fetch opnds ...

  ④ Perform Op — ex: Add Reg 1, Reg 2 ---

  ⑤ Store Result — to dest: Regs, Memory

  ⑥ Calc Next PC, go to FNI — Most incr., branches do more.

- HOW DO WE DESIGN SUCH A CONTROLLER?

We construct a state diagram and impl. in PLA FSM. But, many states, 50 - 100+, can be very complex.

- To structure this problem: Fig 5

Form state diagram as matrix. Vertically we stack up the processing stages FNI, Decode, etc.

Then, we have one column for each instruction type, as many columns horizontally as instructions.

[While many states, transitions are usually single and local.]

- Figure 5 shows INFORMAL EXAMPLES

of control sequencing to execute some different instruction types, to give a feeling for the details.

(TALK THRU THESE A BIT)

- IN TEXT THERE ARE CORRESPONDING "TABULATED" SEQUENCES.

THE # CYCLES TO DO THESE DEP. ON DATA PATH CAPABILITY, i.e. HOW MANY THINGS CAN GO ON IN PARALLEL.

(SHOW THESE 3 SLIDES)

- YOU'LL FIND IT INSTRUCTIVE TO EXAMINE THESE SEQUENCES.

- Note that they look like "programs" written in a very low level "machine language."

- This anticipates the concept of "Micro-Programmed Control"
- SOMETIMES: > Entire Mach. Inst. Set not Defined by the machine is being designed.
  > Some users might want to run prog. for another machine type. Could sim., but inefficient.
- Thus, often wish we could have some sort of writable controls ... so we could change them. Would help in debugging also.
- So, computer designers have often used MEMORY to hold control sequences ... i.e. implemented the FSM with a memory (FIG 6).

  This is inefficient: F flg bits, N next state lines, & I-bit instructions, then need $2^{(I+F+N)}$ words.
  But can usually easily reduce by inserting logic in feedback path:
- In figure (Fig 7), some logic will call the "Microprogram counter Path" is inserted in the path between the (two) memory and decoder.

  This type of control is generally referred to as MICRO PROGRAMMED control, whether wr. table or Read only memories are used.
- Now the design of the control logic is reduced to encoding the sequences of control bit patterns to be stored in the MICRO-CODE memory.

- THE "MICRO-PROG CNTR PATH" similar to the data path:

  IT IS CONTROLLED BY OUTPUTS OF THE μ-CODE MEMORY.
  ITS PURPOSE: TO REDUCE THE AMOUNT OF μ-CODE MEMORY REQ'D.
  DOES THIS BY:
  - MAPPING THE F+N bits of state into smaller #, then decoded to address μ-code memory.
  - Reduces N by allowing complex fcns within μPC to be specified with just a few bits of control info.

- THE CONTROLLER CHIP DESC. IN CHG IS THE μPC PATH PORTION OF A MICRO-PROG. CONTROLLER FOR OM2. (GO BACK TO OV. SLIDE)

- ALTERNATIVE WAY TO VIEW FIG 7:

  > EXAMINE LOOP FORMED BY μPC, Decoder, μ-CODE MEM

  > VIEW μ-code mem address as an "Inst ADDRESS" and wires from μ-code mem to μPC as "INSTRUCTION"

  > This alternative view is shown in FIG 8

- What we've really done is create another STORED PROGRAM machine within our STORED PROGRAM machine,

  So as to put as much fcn capability as possible in the path between machine inst and decoder of the state machine.

- ONE NOTE OF WARNING: VERY LITTLE IS EVER GOING ON AT ONE TIME WITHIN THE SPM --- THINK OF ALL THAT MEMORY, THE FETCH, EXEC, STORE, etc.

6.978 SEMINAR — TUESDAY OCT 31

- TODAY: Douglas Fairbairn, XEROX PARC,

  Seminar: Interactive Graphics Aids for Integrated System Design

---

- Next Time: Midterm Exam: Bring your colored pens/pencils Books, Notes, Scratch Paper

---

- Projects: Returning Project Assign #1.
  - > Not grading separately. If you do #1, 2, 3, they will be checked and counted as 10's.
  - > The overall project incl. the final report will be graded, and count as an equal to the exams.
  - > Handing Out List of Projects so far: Discuss. Many very ambitious - But that is O.K if you have contingency plans for smaller pieces to complete for projects. Actually, that helps provide the CONTEXT for smaller steps
  - > The List may help you Identify possible collaborators.

- TODAY IM VERY PLEASED TO NAME DOUG FAIR. OF XEROX PARC JOIN US TO PRESENT A SEMINAR ON —.

  DOUG IS A MEM. OF STAFF AT PARC, AND PRES. IS AGT. MGR OF THE LSI SYSTEMS AREA.

  HIS RES. INTERESTS RANGE FROM COMPUTER ARCH - ESP OF PERSONAL / DISTRIBUTED COMP SYS — TO INTEG SYS ARCHITECTURE & DES. METHODOLOGY.

  DOUG WAS ONE OF THE FOUNDERS OF XEROX'S LSI SYS. LAB, AND THE WORK HE'LL DESCRIBE TODAY WAS ONE OF HIS FIRST PROJECTS WE UNDERTOOK IN COLLAB W. THE UNIV Researchers / STUDENTS.

(10-30-78)

	PROPOSED PROJECT	TEAM	COLLAB*	SIZE	STATUS
Sandra Azoury	CRT CONTROLLER	A		M	FIRM
Moshe Bain	PROG. CLK GENER (OR) PROG. WORD GENER.		D/J	L	TENT.
Bob Baldwin	LCS Net "NAME SOLDIER"		C	M	FIRM
Andy Boughton	SERIAL DATA MANIPULATOR, ARBITERS	B		L	FIRM
Lynn Bowen	CRT CONTROLLER	A		M	FIRM
Jarvis Dean Brock	SER. DATA MANIP., ARBITERS	B		L	FIRM
Randy Bryant	SER. DATA MANIP., ARBITERS	B		L	FIRM
Jim Cherry					
Michael Coln	D TO A CONVERTER	C		M	FIRM
Martin Fraeman	PROGRAMMABLE INTERVAL CLOCK			L	TENT.
Steven Frank	WRITEABLE PLA			M	FIRM
James Frankel	(BIT SLICE MICRO PROCESSOR)			L	TENT.
Nelson Goldikener					
Tak Hiratsuka			J		
Siu Ho Lam	AUTOCORRELATOR			L	FIRM
Clement K. C. Leung	SER. DATA MANIP., ARBITERS	B		L	FIRM
David Levitt			J		
Rae McLellan					
Glen Miranker					
Craig Olson					
Ernesto Perea	BIT SLICE M-PROGRAM SEQUENCER			M	FIRM
Robert Reynolds	D TO A CONVERTER	C		M	FIRM
Gerald Roylance	FREQ. SYNTHESIZER		D/J	L	TENT.
Jorge Rubinstein	CRT CONTROLLER	A		M	FIRM
David Shaver					
Alan Snyder	ASSOCIATIVE MEMORY			M	FIRM
Guy Steele	LISP MICROPROCESSOR		C	VL	FIRM
Richard Stern	FIR FILTER			M	FIRM
Robert Todd					
Paul Toldalagi					
Scott Westbrook					
Runchan Yang					
Prof. Jonathan Allen					
Prof. Dimitri Antoniadis	PROJECT SET TEST PATTERNS		D		FIRM
Prof. Fernando Corbato			J		
Johan De Kleer					
Prof. Clifton Fonstad					
William Henke					
Thomas Knight					
David Otten	(BUS INTERFACE CLOCK)				FIRM
Prof. Paul Penfield					
Prof. Richard Thornton					

* Collaboration: D: Want Design Collaborators
C: Want Checking Collaborators
J: Possibly interested in Joining or Forming a Team Project.

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 36-575

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

Memorandum

To: IC Group (and others interested)

From: Lynn Conway

SEMINAR ANNOUNCEMENT

Douglas Fairbairn
Member of the Research Staff
Xerox Palo Alto Research Ctr.

Will be showing a film and giving a seminar on:

Interactive Graphic Aids for Integrated System Design

Date/time: Tuesday, October 31, 1978

Room 39-400

1.30 - 3.00 p.m.

LECTURE #14 NOVEMBER 7

- TODAY: REVIEW EXAM; EFFECTS OF SCALING
  GLEN: CELL LIBRARY
- NOTE: PROJECT ASSIGN #2 DUE THURS. PLEASE HAND IN XEROX COPY SO I CAN KEEP YOUR DET. PROJ. DESC. FOR REFERENCE

  TIME IS MARCHING ON. ANYONE WHO HASN'T GOT A PROJECT IDEA BY NOW SHOULD MAKE AN APPT TO SEE ME – I'LL HELP SELECT AN IDEA.

BY THE WAY: GOOD NEW RE PROJ. SET IMPL.

- NOTE: ON NOV 21 (TUES) DICK LYON OF XEROX PARC ~~[illegible]~~ WILL PRES A SEMINAR ON VLSI IMPL. OF SPEECH PROCESSING FUNCTIONS.

  HE WILL PRES. SOME BASIC BUILDING BLOCKS ADDERS, MULTIPLIERS, MEMORY SUBSYSTEMS, OUT OF WHICH ONE CAN BUILD A DIG. SIGNAL PROC. SYSTEM. HE WILL THEN DISCUSS AN EXAMPLE PROJECT CHIP NOW IN DESIGN: A N30 CH ~~ALL~~ DIGITAL BANDPASS FILTER BANK WHICH WILL FIT ON ONE CHIP EVEN IN PRES. TECHNOLOGY.

  HE WILL GIVE A PRES. OF THE OV. SYS. DES OF AN ISOLATED UTTERANCE RECOGNITION SYS. WHICH SHOULD FIT ON ONE OR A FEW CHIPS WITHIN A FEW YEARS. (FOR VOICE INPUT TO COMPUTERS)

- NOTE: CALVERT MEAD OF CALTECH WILL PRESENT A SEMINAR ON HIGHLY CONCURRENT SYSTEMS (CHAP 8 MAT'L) ON TUESDAY DEC 5.

[TRY TO HAVE LEVITT, TODD, TAKE EXAM DURING CLASS.]

- LETS REVIEW THE EXAM: HAND-OUT GRADED EXAMS

- THE MEAN = 80, MEDIAN = 86
  HIGHEST GRADE = 96 (2 people: GUY STEELE, MOSHE BAIN)

- 11 between 90-96
  9 between 80-89
  10 below 80

MOST DID VERY WELL. CONSIDER OVER 80 AS A GOOD GRADE.

REFER TO EXAM SHEETS:

I GRADED FAIRLY HARD. A FEW WHO HAVE DONE VERY WELL ON HW DID POORLY. MAYBE HAD A BAD DAY. DON'T BE DISCOURAGED. BUT DONE LIKE DID...

Problem 1 (a) $C_L = 0.8$ pf

$C_{g(min)} = 4 \times 10^{-4}$ pf $\times (6 \times 6)\mu m^2 = .0144$ pf

$$Y = \frac{C_L}{C_g} = \frac{0.8}{0.0144} = 55.6$$

For min delay, ratio of successive sizes (f) = e.

$f^N = Y$, $N = \ln_e Y = \ln(55.6) = 4.017$

$\therefore N = 4$ stages

Grading: Calc. Err: ~ −4 ; Incorr. form ~ −8

(b) Trickier than it looks: Array contains 256 pairs of inverters.

3 SLIDES

i.e. 16 rows of 16 cells × 2 inv. per cell (4 per cell pair)

- However, only half can be on at one time Maximum.
- First of each pair is approx 5 □'s long when "on"
- Second is " 6 1/6 □'s " " "
- So worst case is when all "1st" inverters "on".
- In this case, $I_{total} = 256 \times \frac{5^V}{5 \times 10^{-4}} = 25.6$ ma
- Wires can carry max 1 ma/mm²; But are 1μm thick, so:

$\therefore$ WIRES ≥ 25.6 μm wide −3

Grading: Calc errors −3 to −4; Invalid Assump: −4 to −6; Forgot pulldown R

Problem 2: Most people did O.K. Most PLA's matched the logic equations used.

Errors were in manipulation of logic equations, or in not getting minimum # p-terms.

Grading: −4 to −6 for log. eqn error(s)
−3 for not minimizing # p terms
−3 for PLA coding error

Problem 3:

(a) Yellow implant must be 1½ λ from neighboring enh. mode transistor gate region (SLIDE)

Some said metal not over cuts, but look closely, O.K.

(b) Several Major Errors: (SLIDE)

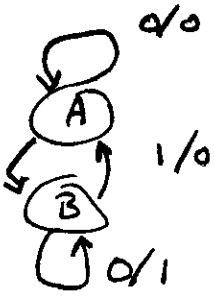
> Ratio should be 8:1, not 4:1.
> No implant over pullup to make depl. mode.
> Metal over Butting contact is 2λ not 3λ from GND METAL.
> Some DIFFS ARE 2λ sep. not 3λ.

(c) EXAMPLES OF REAL IMPROV. IN DESC. LAYOUTS OVER WHAT CIF CAN DO:

> DIG AT CORNERS OF BOXES / NOT CTR)
> DIRECT ITERATION (AS IN INFORM. DESC. LANG IN CHAP 4)
> PASS PARAMETERS TO CALLED SYMBOL: USE FOR EXAMPLE: SCALING
> USE #'S OTHER THAN JUST INTEGERS, ALLOWING EASY SPEC OF SIZES IN λ.
> CALL SYMBOLS BY NAME INST OF #.

, etc.

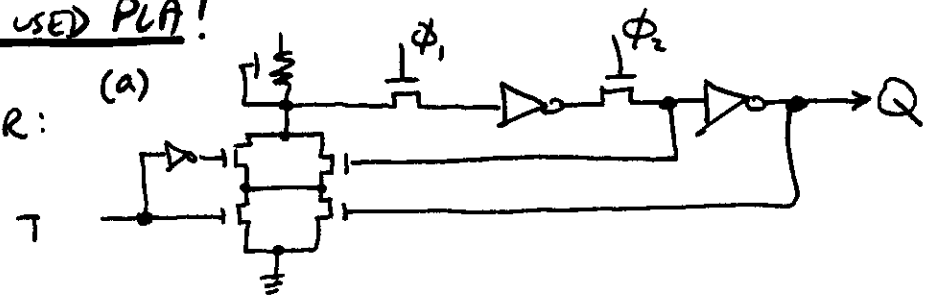
PROBLEM 4: T → TFF → Q



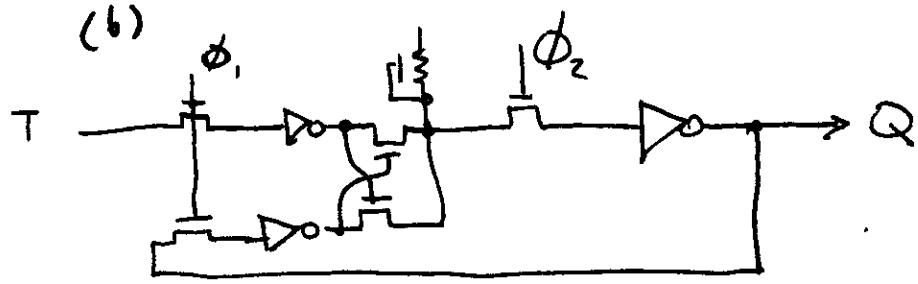
MANY SOLNS: A FEW USED PLA!

- SOME BASED ON XOR:

(a)



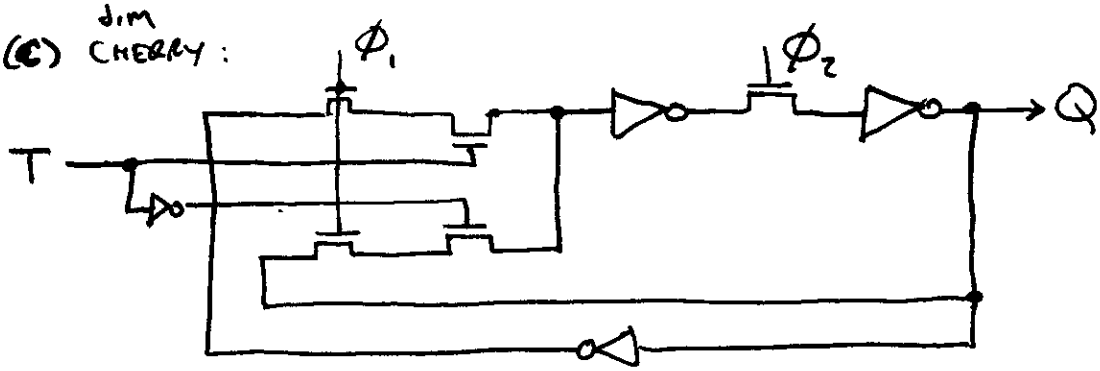
(b)



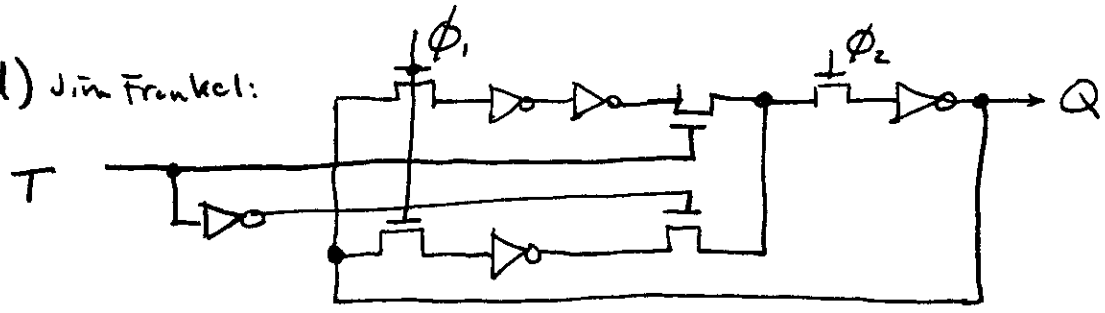
- OTHERS:

Based on Polarity Selecting inverted or non inverted feedback based on value of T

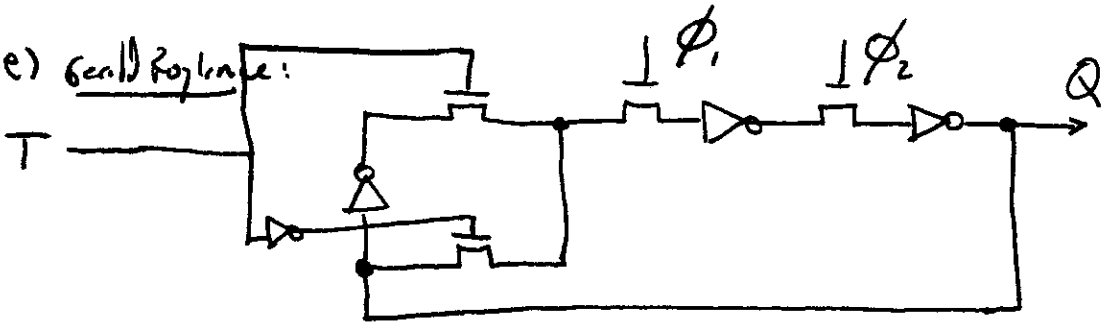
(c) Jim CHERRY:



(d) Jim Frankel:



(e) Gerald Roylance:

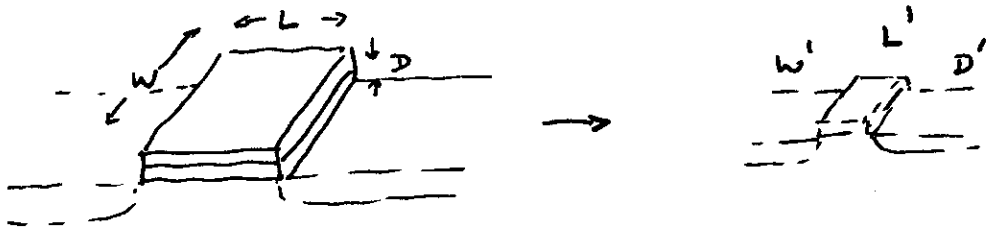


WHAT HAPPENS AS WE MAKE THINGS SMALLER?

i.e. as $\lambda \rightarrow \lambda/\alpha$ IN FUTURE

THE EFFECTS OF SCALING DOWN DIMENSIONS OF INT. SYSTEMS:

SUPPOSE WE USE A NEW PROCESS IN FUTURE, SO THAT ALL DIMENSIONS PARALLEL TO SURFACE & VERT ARE REDUCED BY A FACTOR α. WHAT WILL BE EFFECTS ON ELECTRICAL PROP OF DEVICES?



- Distances divided by α: $W' = \frac{W}{\alpha}$; $L' = \frac{L}{\alpha}$; $D' = \frac{D}{\alpha}$
- Also, to keep all Electric Fields same as before, we reduce VDD and set all thresholds to be reduced by same factor α.
- This is a particularly simple form of scaling to analyze. The actual forms used will differ as we approach extremely small device sizes, but this form may take us down to or under 1 μm feature sizes ($\lambda = 1/2\,\mu m$).
- Now what happens to τ, to C_g, to I_{ds}, etc?
- Remember Eqn 1(CH.1): $\tau = L^2/\mu V_{ds}$

$$\therefore \frac{\tau'}{\tau} = \frac{(L/\alpha)^2}{(V/\alpha)\cdot L^2/V} = \frac{1/\alpha^2}{1/\alpha} = 1/\alpha$$

$$\therefore \boxed{\tau' = \tau/\alpha}$$

Transit time of min devices, & thus circuit speeds all scale down linearly with α

- Gate Capacitance: $C_g = \epsilon WL/D$

$$\frac{C_g'}{C_g} = \frac{(L/\alpha)\cdot(W/\alpha)}{(D/\alpha)}\cdot\frac{1}{WL/D} = 1/\alpha \qquad \boxed{C_g' = \frac{C_g}{\alpha}}$$

- Note, however, that Capacitances in general scale up by α if given in terms of $C/\mu m^2$.

  Since for given absolute W & L, D gets thinner.

- Note: Resistances/$\square$ in general scale up by α

  since lines get thinner vertically. However, the R/$\square$ of transistors remains ~ same.

- Because of this, and because of other problems with POLY (if clumps of crystals too big R gets large, and this gets worse as get smaller), the ratio of POLY res to FET res may get worse as get smaller.

- FROM EQN 3: (CH.1) $I_{ds} = \frac{\mu \varepsilon W}{LD}(V_{gs}-V_{th})(V_{ds})$

  So $I_{ds} \propto WV^2/LD$

  $$\frac{I'}{I} = \frac{WV^2/\alpha^3}{LD/\alpha^2} \cdot \frac{1}{WV^2/LD} = \frac{1}{\alpha}$$

  $$\boxed{I'_{ds} = \frac{I_{ds}}{\alpha}}$$

- So, current per device goes down by $\frac{1}{\alpha}$
- However, # devices per unit area goes up by α^2
- $\therefore$ Current Density over the chip (if uniform design or repeated design sub-blocks) goes up linearly with α

  This is another problem: We may need METAL wires which are proportionally wider compared to previous designs as we scale down other features, if the wires are near the 1 ma/μm^2 current limit.

  One solution: Increase H/W of wires by α^2 as scale down. Some of this can be done, but impract. to do completely. So some wires get wider!

SCALING OF POWER DENSITY:

- **DC POWER DISSIPATION:** Per device $P_{dc} = I \cdot V$

  Since $I' = I/\alpha$, $V' = V/\alpha$, $\boxed{P'_{dc} = \frac{P_{dc}}{\alpha^2}}$ per device.

  But the number of devices increases as α^2. So, power density (average) over the chip (if regular pattern scaled) remains constant.

- **SWITCHING POWER:** The drivers which operate pass gates, charging & discharging capacitances, dissipate switching power. We can estimate this roughly ~~with dc power~~ by ~~calculating~~, estimating I, V during transients. But more directly:

  $P_{sw} \propto \frac{CV^2}{T}$; i.e., P_{sw} is the energy stored on the capacitance of a device, divided by the clock period or time between succ. charging/discharging. And, we know that $T \propto \tau$.

  Thus, $P_{sw} \propto \frac{WV^3}{DL}$, $\boxed{P'_{sw} = \frac{P_{sw}}{\alpha^2}}$ per device

  So both dc & sw power remain cont. per unit area.

  Note: the average dc power for most systems can be approximated by adding total P_{sw} to 1/2 the dc power resulting if all level restoring logic pulldowns were turned-on.

  [P_{sw} dis. in control drivers to pass gates].

- RULES OF THUMB FOR POWER DENSITIES:

While things don't get worse as we scale down, be careful:

> over areas of dimensions ≫ larger than wafer thickness (i.e. macroscopic dimensions), the following rough rules of thumb apply:

< 1 watt/cm²	no problem at all
2 watt/cm²	} somewhere in here, you begin to need special heat sinking
4 watt/cm²	}
> 8 watt/cm²	begin to need forced cooling of some kind to remove heat.

> As in all thermo problems, all depends on next higher context. One bolted chip in well heat sinked package may do well in still air at 4 w/cm², while a whole board packed full of them would need to have plumbing and be freon-cooled.

- What can I say to summarize: Be careful!

Calculate power dissipated by large projects or full chips. If > 2 watt/cm² you may get in trouble if want to use a lot of them.

> Ways out: There are often many ways to trade-off power vs time.

Remember, by using longer pullups/pullDowns we can keep same ratio in shift register, but use less power, at price of longer delays.

- LET'S CALCULATE PERHAPS A WORST CASE FROM AMONG OUR DESIGNS:

SHIFT REG IN FIG. 8b (CH4): THIS HAS WIDE PULLDOWN, SHORT PULLUP, EVEN THOUGH 8:1.

CELL AREA = $19\lambda \times 21\lambda = 57\mu m \times 63\mu m = 57\times10^{-4}cm \times 63\times10^{-4}cm$
$= 3591\times10^{-8}cm^2 = \underline{3.6\times10^{-5}cm^2}$

Power = $I \cdot V = \frac{5v}{60K\Omega} \cdot 5v = \underline{4.2\times10^{-4}\ watts}$

(ACTUALLY THE 10KΩ/□ IS FOR FAST TRACES ∴ IS OPTIMISTIC)

PWR/AREA = $\frac{4.2\times10^{-4}}{3.6\times10^{-5}} = 12 w/cm^2$. BUT IN SERIES ONLY 1/2 ON AT A TIME

$\therefore$ PWR/AREA $\approx$ 6 w/cm²

SO IF YOU FILLED A CHIP WITH THESE, YOU'D GET IN TROUBLE!

- AS AN ASIDE, ONLY ≈ 2K of ~~these~~ in pairs would fit on a ≈ 5mm × 5mm chip.
So how do we get 4K, 16K etc. memories, but stay within power limits?

> We don't use static cells for such memories. We store charge on capacitors. More next week

- Also, if want denser cells and have power problems, there are a variety of circuit tricks that can be used. Such as "clocked pullups". More next week.

IN GENERAL, WITHIN OUR DIG SYS, USING DES METH IN TEXT, WONT HAVE PROBLEMS. REGIONS OF WIRES AND PASS GATES DONT DISS. DC POWER, & COMP. FOR REGIONS OF DENSER ~~STUFF~~. LEV. REST. CELLS.

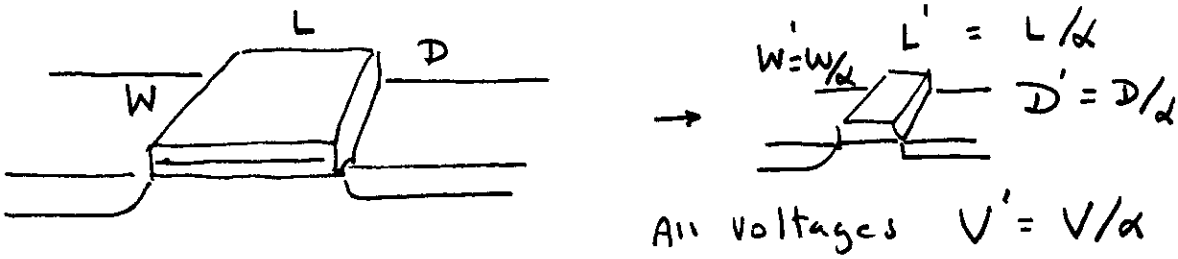
BUT IF MAKE REALLY BIG ARRAYS OF SR'S, STACK CELLS, ETC. CALCULATE PWR DENS. TO BE SAFE.

- IN ANY EVENT: SCALING DOES NOT MAKE THIS WORSE

LECTURE #15 NOVEMBER 9

- Collect Proj. Assign. #2
- LAB/PROJ: Use only BOXES at right λ's. Later software won't support more.
  Also: We have priority in Lab > 3 P.M.
  POINT OUT $ COSTS: ONCE DO RIGHT, STAMP OUT LINE COOKIE AT ~$1 OR SO A PIECE.
- TODAY: CONTINUE SCALING.
  DISCUSS LIMITING FACTORS.

- SUMMARIZE SCALING SO FAR:



All voltages $V' = V/\alpha$

Using this simple scaling by α of all dimensions including vertical, and all voltages (V_{DD}, V_{Tn}, V_{Dep}) we found:

$$\boxed{\tau' = \tau/\alpha} \quad ; \quad \boxed{I_{ds}' = I_{ds}/\alpha} \quad ;$$

PROBLEM ENCOUNTERED!

[ex.: EQ 1: $\tau = \frac{WC}{\mu V_{ds}} \therefore \downarrow$]

But # of devices per unit area goes up by α^2
So mean current density: CURRENT INTO AREA OF CHIP goes up by α. This is real problem, since in our scaling wires would get thinner. Current limited wires would either have to have aspect ratios increase by α^2 (which can't go on for long) or get proportionally wider.

$$\boxed{C_g' = C_g/\alpha} \quad ;$$

But capacitances in general scale up by α if given in terms of $C/\mu m^2$ since vertical dimensions shrinking (oxide getting thinner).

Resistances/□ scale up by α since lines getting thinner.

(Except note that we can't really do that with curr. lim. metal.)

However, if calculate it out, find that R/□ of FETS will stay about the same.

So another problem: R/□ of poly, diff getting proportionally larger while R/□ FET staying same. This is aggravated by crystal clumping in POLY --- makes effect worse as scale down.

DC Power Dissipation: Per Device $P_{dc} = I \cdot V$

$$P'_{dc} = \frac{P_{dc}}{\alpha^2}$$

; # Dev./Area goes up as α^2

So (whew!) Power dissipation/unit area stays approx constant.

We discussed power dissipation limits: Is much more diff. to pin down to single constraint as in current density in wires. Dependent on next level context

Tabulated:

< 1 W/cm^2	no prob.
2 W/cm^2	} begin to need
4 W/cm^2	} reasonable heat sinking
> 8 W/cm^2	need way to remove heat: forced cooling

We examined ≈ worst case in our methodology:

Large array of shift registers are in Fig 8b CH 4.

Found power/unit area ≈ 10 W/cm^2.

BUT NOTED THAT WE USE PASS-T LOGIC BETWEEN THESE AND USUALLY DON'T USE SUCH SHORT BUSSES, ETC.

SO NORMALLY WE DON'T NEED TO WORRY. BUT SHOULD CALCULATE IF IN DOUBT. TRY TO STAY < 2 W/cm^2 IF CAN. (Cover large areas of chip)

Scaling of

SWITCHING POWER: The drivers which operate pass gates, charging & discharging capacitances dissipate switching power.

The power is dissipated at the drivers, but we calculate the amount based on the capacitances & voltages & clock period:

P_{sw} = energy stored on capacitance divided by the clock period or time between successive chargings / discharging.

But $T^2 \propto T$. Thus, $P_{sw} \propto \frac{CV^2}{T}$; $T \propto \frac{L^2}{V}$

$$\therefore P'_{sw} \propto \frac{WL}{D} \cdot V^2 \cdot \frac{V}{L^2} = \frac{WV^3}{DL} \quad \therefore \boxed{P'_{sw} = \frac{P_{sw}}{\alpha^2}}$$

So: Since P_{sw} per device goes down by α^2, and # dev./area goes up by α^2, the switching power also stays constant per unit area as we scale things down.

NOTE: AVERAGE DC POWER DISS. IN MOST SYSTEMS CAN BE APPROXIMATED BY ADDING TOTAL P_{sw} TO 1/2 DC POWER RESULTING IF ALL LEVEL RESTORING LOGIC WERE TURNED ON.

SWITCHING ENERGY

We've noted before that there are various ways to trade off power vs delays. We can often use less power if we can tolerate longer delays, and vice-versa. This can be done by building it into the design, or sometimes can be controlled dynamically. This reflects an important metric of device performance: **SWITCHING ENERGY per DEVICE**

E_{sw} = power consumed by device at max clock freq, multiplied by the delay: i.e., it is a "power × delay" product.

Rationale: In a sense, to do any computation, we must switch a large collection of switches, switching at minimum (Think of toggle switches) them in a particular order, and some number of times.

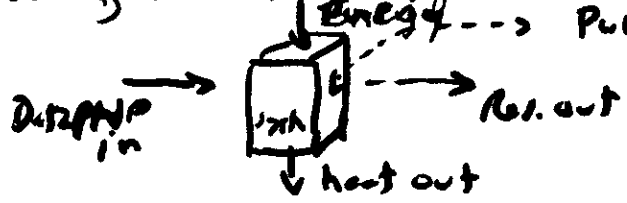
Switches have some switching energy which measures the work done to throw the switch. We often have the option (by design or control) to choose to put the energy into a system slowly, (and thus less power) taking longer for a calculation. Or – put it in faster, flipping the switches faster. (SEE CHAP 9)

However, there are usually constraints on both speed and power, and these limit ultimate performance:

No matter how much power we put in, we can't reduce delays below the minimum value of τ.

Also, if we put in too much power, we may reach a power density limit in a particular design even before reaching the fastest speeds.

Min Energy of comp. = Min Power × Time : # Switchings × E_{sw}/dev



How does switching energy scale:

Our basic FET switches have $E_{sw} \propto CV^2$

and $\therefore$ $$\boxed{E'_{sw} = \frac{E_{sw}}{\alpha^3}}$$

so this crucial metric of device performance scales incredibly favorably. This is why scaling down sizes is so important.

Summary so far: Suppose we

Scale down an entire system by $\alpha = 10$.

> Resulting system will have 100X as many devices/unit area. (Or take only 1/100th as many chips)
> Power Density remains constant.

> All voltages reduced by factor of 10
> Current/Area (per chip) increased by factor of 10
> Time delay/stage decreased by factor of 10
> Power-Delay Product (per device) decreased by " " 1000

This is very attractive scaling except for the current density problem. The delivery of the required average dc current presents an important obstacle to scaling. Even in today's systems, many wires are operated at near their current limit. So, wires must become relatively wider, or have much higher aspect ratios, or both.

(& Don't forget the problems of Poly, Diff res/□ rel to FET)

[Forgetting possible design mechanisms, with problems:
CAN WE SIMPLY SCALE DOWN WHOLE DESIGN? Yes but: Consider]

- Delays to Outside World: (Read SPACE vs TIME in CH 1)

What is effect of scaling on output driver design & delays? We can't just scale them down: the outside world stays big. Remember the result in chap 1:

Min Delay when use factor of e, and $N = \ln Y = \ln \frac{C_L}{C_{g_{min}}}$

In this case: Min Tot Delay $\approx \tau e \ln\left[\frac{C_L}{C_g}\right]$

Now, scale everything down by α, including Voltages. (This we do scale even in the external world)

$\tau' = \tau/\alpha$; $C_g' = C_g/\alpha$; $\therefore Y' = \alpha Y$

DERIVE ↓ ⌜→ $\therefore$ * $\boxed{t'_{min} = t_{min} \cdot \frac{1}{\alpha}\left[1 + \frac{\ln\alpha}{\ln Y}\right]}$ below for deriv. if necessary

So, as inverters get smaller, more stages are required to obtain minimum offchip delay.

The relative delay to outside world increases, But the absolute delay decreases!

ALSO: At Least Driver Designs must change & can't be just scaled down like rest of system (cont.)

Derive *: $t_{min} = \tau e \ln Y$

$$t'_{min} = \tau' e \ln Y' = \frac{\tau}{\alpha} e \ln(\alpha \cdot Y) = \frac{\tau}{\alpha} e[\ln\alpha + \ln Y]$$

$$t'_{min} = \frac{\tau e \ln Y}{\alpha}\left[1 + \frac{\ln\alpha}{\ln Y}\right] = \frac{t_{min}}{\alpha}\left[1 + \frac{\ln\alpha}{\ln Y}\right]$$

ALSO, BRIEFLY MENTION BIPOLAR TTL, I^2L, OTHER MOS: CMOS

SO, SCALING PRODUCES SOME GREAT EFFECTS. → ↓ CONT.

BUT WE SHOULD ASK: HOW SMALL CAN WE MAKE THESE DEVICES AND STILL HAVE THEM WORK?

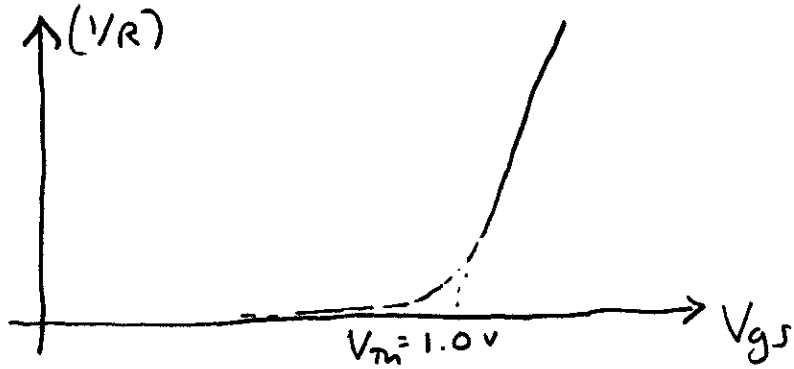
WE MUST SUSPECT THAT THERMAL, STATISTICAL, QUANTUM EFFECTS ARE ULTIMATELY GOING TO MESS THINGS UP!

QUESTION: IF PATTY'S FAB WERE NOT LIMITING US, HOW SMALL COULD WE MAKE FETI AND STILL HAVE THEM WORK?

[MANY FACTORS TO CONSIDER. I'LL DISCUSS SEVERAL OF THE MAJOR ONES, ONE IN SOME DETAIL. IF YOU'RE INTERESTED IN THIS: I SUGGEST READING THE SURVEY PAPER BY KEYES, AND ALSO BROWSING IN CHAPTER 9]

- SUBTHRESHOLD CONDUCTANCE:

IF WE REALLY PLOT DETAILS OF COND. VS V_{gs}, IT IS NOT SIMPLY A STRAIGHT LINE RUNNING DOWN TO V_{TH}, BUT HAS AN EXPONENTIAL TAIL:



Below V_{TH}, the conductance $1/R$ is not zero but depends on V_{gs} and temperature:

$$\frac{1}{R} \propto e^{(V_{gs} - V_{TH})/(kT/q)}$$

T = absolute temp

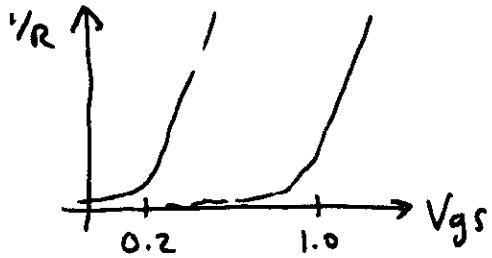
k = Boltzmann's constant

q = charge on electron

At room temperature, $\frac{kT}{q} \sim 0.025$ volts

Thus, at present threshold voltages, an off device, below threshold by perhaps 0.5 volts, is below threshold by $20\,kT/q$. Thus its conductance is decreased by a factor of $\approx 10^7$ over that when on near threshold. Said another way: if used as a pass-T_g Q taking T to pass thru on device will take 10^7 T to pass thru off device.

BUT suppose scale down by factor of 5:



[Curves keep ~ same form. But shifted left.]

Now the OFF FET IS DOWN ONLY BY $4kT/q$, ∴ may have as much as $\frac{1}{100}$ conductance when off as when on.

Use of dynamic storage especially in memories, where stored for many T_g will be increasingly harder, espec. below 1 μm. Of course trying to do things statically causes us power dissipation problems. Ah --- you can now see how we're going to get boxed in.

We could scale without continuing to scale voltages say when VDD reaches about 1V. But this also causes us power dissipation problems!

EVEN NOW VOLTAGE SCALING SHOULD BE DONE TO MF. MOS PERF. sig to 3V

Could reduce prob by Reducing T. See interesting paper by Gaensslen, Rideout, Walker CRL where very small MOSFETs were op. at liquid N temps and measurements confirm improvements.

[But the TTL comp. used at 5V]

ALL ENERGIES CAN BE SCALED AS $\propto kT$

everything as ratios of #kT

Ah, related to low temp operation: Side point:
So → Reducing temperature also reduces $E_{sw_{min}}$. Figures quoted were at room temperature. But be careful!

CH 9 shows interesting comparison: FETs vs Josephsons:

Won't go into full detail, read if you are interested:

Although switching energy at Device is lowered, you must put in energy into the refrigerator to keep it at the low temp, at least as much as difference resulting from lower $T_{operation}$.
NOW, A COMP DIFF TECH: USE FLUX NOT Q TO ST. INFO.
IBM CS's quote the low E_{sw} of Josephson Junctions at the low temperature environment. It turns out that if you scale FETs down to ~ 1/2 μm, gaining the d^3 improvement in E_{sw}, and operate them at the same temp as J-J's — They will have the same E_{sw}!

But to calculate the energy requirement for a computation, must use T at the heat sink temperature. Refrigerating devices to reduce energy of computation is the logical equivalent of constructing a perpetual motion machine.

So: Viewed as a system: FET system and JJ system will have similar ~~energy~~ switching energy requirements. FET will be simple (operate ≈ room temp.). J-J has advantage of trading the power-delay trade off to lower values at delay (~ 1/50 best FETs). J-Js at quantum limits at ~1μm sizes. can't be scaled down smaller. ---

MAYBE DON'T BOTHER WITH THIS

So the factor of 100,000 quoted by IBM CSs of switching energy is wiped out by X1000 (possible) improvement in FETs by scaling, and X100 due to the perpetual motion machine error.

JUST MENTION THESE BRIEFLY:

OTHER LIMITING FACTORS: (SEE REF BY KEYES, SEE CH 9)

- STATISTICAL VARIATIONS IN THRESHOLD VOLTAGE:

  AS WE SCALE DOWN, WE'LL FIND THAT $\frac{\Delta V_{Th}}{V_{Th}}$ is proportional to Scaling factor α.

  Results from granularity & statistical distribution of substrate impurity charges which determine the threshold voltages.

  At same time, # devices increasing. If pullup threshold goes one way, and pulldown another, may end up with inverter which doesn't work. As shown in Ch. 9, this may also limit how small supply voltages can be made. In VLSI system contain 10^7 inverters, if we require probability (that all FETs being within threshold limits) = 0.9, may require VDD $\approx$ 0.7 v.

- Quantum Effects. Gate oxide is already only 1000 Å = 0.1 μm thick. Positional uncertainty for electrons is related to uncertainty in momentum by

$$\Delta p \, \Delta x \approx \hbar$$

For energy barrier of ~1ev, calculating corresponding Δp, we find Δx is about .001 μm. Gate oxides and junction depletion layers must be many times this or electrons will "tunnel" through. Thus we are near a fundamental size limitation due to quantum phenomena.

SUMMARIZING HOW THINGS MAY GO:

	1978	MID-80's	19XX
MIN FEAT. SIZE (2X):	6μm	1μm	0.3μm
T :	0.3 to 1.0 ns	~0.05 to 0.15 ns	~0.02 ns to 0.04 ns
E_{SW} :	$\sim 10^{-12}$ J	$\sim 5 \times 10^{-15}$ J	$\sim 2 \times 10^{-16}$ J
LOCAL SYNCH SYS CLK. PERIOD (~100T) :	~30 to 100 ns	~5 to 15 ns	~2 to 4 ns

> The mid 80's column we'll probably reach without major hassles. Voltage will be scaled to 1½ or 1V, and power density won't be too much of a problem.

> Subthreshold current will be emerging as a problem, but not within our digital processing structures where "refresh" occurs every 50 ns or so.

> Current density will be a rapidly emerging problem, but will be handled with more area devoted to power lines, and higher-aspect ratio wires.

- Getting the last order of magnitude out of the technology before fundamental physical limits are finally hit will, however, be a major hassle. It will require close collaboration of researchers spanning the range from CS, to Arch, to E.E., to Device phys., to materials, in order to provide the overall context to help narrow down/select the alternatives to explore.

ON ORDER OF WAVELENGTHS OF LIGHT! .4 to .7 μm — OF COURSE, WE ARE STILL LEFT WITH THE PROBLEM: HOW DO WE MAKE SYSTEMS THIS SMALL? MV ≈ .3μm

6.978 LECTURE #16. NOVEMBER 14.

TODAY: MORE ABOUT THE FUTURE: HOW THE PATT'S FAB TECHNOLOGIES MAY BE IMPROVED TO ACHIEVE SCALING TO LIM. DIM.

FIRST: PROJECTS: NOTE: NO OFF. HRS. TMRW. FRIDAY INSTEAD

- Prof. Antoniadis is looking for several students to collaborate on designing & laying out electrical test patterns. These will be acceptable projects and any who haven't started work on a project yet - I suggest you speak with Prof. Antoniadis right after class. DIMITRI - perhaps you could say a bit more about this: ---

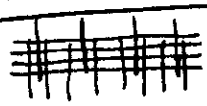
- There are some students who haven't yet turned in enough of the project assignments for me to get a clear idea of your project selection. Please see me after class if you are on this list. (I'll finish a bit early):
  Rae McClellan, Dave Levitt, Dave Shaver, Scott Westbrook, Moshe Bain, Martin Freeman.
  (FIFO) (multi-function memory) (test patterns) (clock generator)

- You may have noticed the OVG cuts in the PAD cells over the contact pads. We will be producing an OVG mask, which could be used in later processing to pattern protective OVG. However, in the fab of the project set, no OVG will be placed. You will therefore (may) be able to probe any large (> 25μm x 25μm) metal features as test points. Sometimes people put these in as contingencies (rather than bringing out all test points to contact pads.) But be careful of long C's AND gates can't be tested!

- A word about Functional testing: Testing uncovered chips requires reduced light levels. The operation of dynamic circuits using pass transistor input to a gate can be severely affected by light. Light induces leakage currents in the n-p junctions between source and drain and substrate. At room T, charge may be stored for a number of milliseconds on dyn. nodes IN ABSENCE OF LIGHT. However, in normal room light, this time may be reduced to tens of microseconds. Avoid light when long clock periods are used. Dyn. Mem. Chips packaged in Blk Opaque Packages because of this effect.

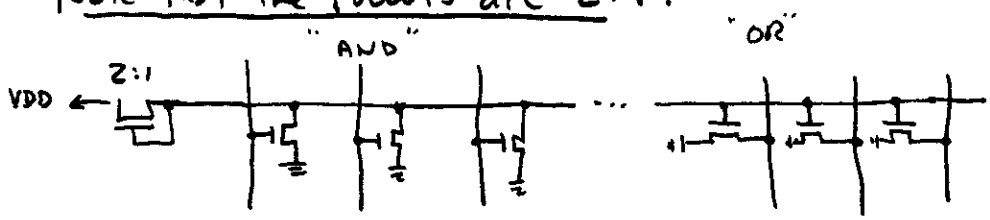
- ~2 weeks to get to prelim layout of entire subsystem. I want them to check IF want to get on fast chip out.

- **COMMENT ON WIRING STRATEGIES**



- **The PLA Library cells**: Several of you are using the PLA cells in your projects. You've already noted there are differences in the PULLUPS & the IN & OUT registers. The other cells look the same as those in the book. BUT, there is a key difference - I want to thank Glen Mitsubori for bringing this to my attention:

Note that the PULLUPS are 2:1.



It was designed so that the PULLDOWNS could be 1:2. This make the ratio 4:1 as required. BUT NOTE, if you programmed it with min width diffusion lines, you would make pulldowns that were 1:1, and overall ratio would be only 2:1.

Reason for 1:2 PullDowns: In large PLA, the limiting factor will or might be the FANOUT into the OR plane. The larger pulldowns and shorter pullups can source or sink more current and can thus drive this larger capacitive load twice as fast as the design in the book.

The 1:2 PullDown doesn't enlarge the basic PLA cell pitch area. We use a 4λ x 4λ DIFF box to place a transistor rather than a 2λ x 4λ box. Sketch on SLIDE

- **FINAL**: I had originally planned to give a Take Home Final Exam. However, I now think that such a final would be anticlimactic after the projects. I'd much rather that you put your energies into your projects and not worry about a final. So I don't plan to give a final.

Those who didn't do well on the midterm: please remember that I will be giving the PROJECT much more weight than the M.I.T. Exam. If you are still concerned — see me.

PATTERNING & FABRICATION IN THE FUTURE:

IN the previous lectures, we examined the quantitative effects on performance of scaling down the dimensions of devices in our systems, i.e. The effect of making things smaller. We also studied factors which limited such scaling ie. HOW small the transistor can be made and still function.

Now: How in fact can We make things That small? Recall that the limiting feature sizes were on the order of 1/4 micron feature sizes. The wavelength of visible light is ~ 0.4 to 0.7 μm. UV is ≈ 0.2 to 0.3 μm

- LETS EXPAND OUR VIEW OF THE FUTURE FROM THE SIMPLE TABLE GIVEN LAST TIME OF SIZE/PERF. as fcn time. TO INCLUDE HOW THESE CHANGES CAN BE BROUGHT ABOUT. NOTE: ALTHOUGH AS SYSTEM DESIGNERS WE WON'T NEC. BE INVOLVED IN THESE PATT/FAB TECH. CHANGES, WE MAY NEED TO KNOW ABOUT THEM, TO UNDERSTAND HOW THEY MAY AFFECT SYSTEM DESIGNS OR DESIGN FILE PREPARATION. THIS IS ESP. TRUE FOR THOSE COORDINATING PROJECT SETS.

- FOR THE DESIGNER WHO DOESN'T GET INVOLVED IN IMPL., OUR FILM PROCESSING ANALOGY WILL HOLD, EVEN THOUGH THE PROCESSING TECHNOLOGIES ARE RAP. CHANGING: WE'LL KEEP GETTING FASTER & FINER GRAIN FILM EACH YEAR - - -

AS WE PROCEED FROM 6μm to 0.3μm Feature sizes,

1. CIRCUIT & DEVICE TECHNOLOGY & DESIGN RULES & CONSTRAINTS
2. PROCESSING TECHNOLOGIES
3. PATTERNING TECHNOLOGIES

} WILL ALL BE CHANGING.

LETS EXAMINE THESE IN ORDER:

DESIGN:

(1) nMOS will very likely be a standard technology for high density/high performance LSI/VLSI for the next 4 to 6 years at least. When feature sizes go under 1μm, we will be forced into new device, circuit, system design techniques because of getting boxed into the current density vs power density corner as we saw last time. A likely candidate for the 1μm to 0.3 μm scaling is CMOS, which is similar as a medium for design.

However, we needn't be too worried about these changes. During the next 4-6 years, computer aids for design will be rapidly improved. Design will be done at a higher level - more like arranging the floor plans of subsystems and compiling stick diagrammed cell arrays from libraries into target design rules - with aids helping check for current/power density limits, etc.

So, as fast as changes in the way we design or constraints on design occur, we will get improved design aids - design should get much easier.

---

(2) PROCESSING TECHNOLOGIES: Even if we could pattern resist with features smaller than the wavelength of light, current processes couldn't "develop the film":

Independent of any particular technology,
(a) diffusions produced by placing wafers in gases at high temperature, and (b) wet etching techniques, are not sufficiently controllable to achieve fine feature sizes.

Some solutions:

(a) Ion implantation is even now replacing earlier techniques for diff. of impurities into the substrate. Offers high degree of control over dosages, and high uniformity.

Basically, the wafer with patterned resist is simply exposed to ions accelerated in a big ion accelerator. The ion implanters are expensive, but in principle are simple. Also, easily automated. Wafers can be delivered on a truck and flipped into position in a lock, then exposed, then continue on.

(b) Wet etching gradually replaced by etching with Plasmas: i.e. by using glow discharges of gases to produce Free ions of great chemical activity. Again, here, can achieve great control over the etching process.

(c) How might we achieve high aspect ratio wires? Techniques are evolving. One possibility is called ion milling. Ions accelerated to modest energies sputter away metal not covered with resist. Can yield sides much steeper than with wet etching.

PATTERNING: O.K., maybe improved design systems will help us cope with changing technology & more design constraints, and process technology may evolve to etch or implant fine details. But how do we pattern resist with features finer than wavelength of UV light?

Patterning Technology is undergoing a rapid evolution. It is important that we have a feeling for this because a lot of information handling is involved in patterning & patterning technology imposes many constraints on designs and design files.

Also, great reductions in impl. turnaround time can be made by system designers working to take advantage of new patterning technologies — which provide opportunities for more automation of the overall process.

- LETS SKETCH THE COMING EVOLUTION IN PAT TECH BY BEGINNING WITH WHATS HAPPENING RIGHT NOW --- THEN WORKING FROM THERE.

(SLIDES) MOST FAB LINES USE "WORKING PLATES" TO MAKE A SORT OF "CONTACT PRINT" EXPOSURE TO PATTERN RESIST. THE PLATES GET DIRTY & WEAR OUT.

- SO: TREND IS TO PROJECTION EXPOSURE OF 2X (or higher) MASK ONTO THE WAFER. MASTER MASKS CAN BE USED (DON'T WEAR OUT). LARGER FEATURE SIZE ON MASK MAKES IT A BIT EASIER TO PRODUCE MASKS OF A GIVEN ON-CHIP FEATURE SIZE.

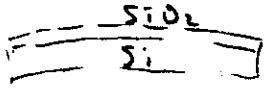
SUCH PROJECTION EXPOSURE WILL BE GOOD DOWN TO ½ μm FEATURE SIZES. BELOW THAT WE'LL NEED AN ALT TO UV light. BUT MORE THAN feat size to worry about. λ also result of ~1 μm

- BUT EVEN AT ≈ 2 μm Feature size ($\lambda = 1$ μm) WE

ENCOUNTER A PROBLEM WITH RUNOUT: So far we've always thought of exposing whole wafer at once. Can we continue to do this? Possibly not. Consider:

When bare wafer is heated, it expands. Now suppose SiO_2 is grown. Thermal coeff of exp of SiO_2 is ≈ 1/10 that of Si:

As wafer cools, Si shrinks more than SiO_2. So, wafer won't be flat, but will be convex on the SiO_2 side



Now if cooled slowly, may be possible to relieve the stress induced by the diff in ~~exp~~ contraction.

- But — while wafers might then be flat — they are of a slightly different size than originally. Unfortunately this change is pattern dependent, and so we can't resort to simply making successive mask layers larger.

- As wafers get larger, and feature size, smaller, at some point full wafer exposure must be abandoned because won't be able to align a successive layer over whole chip.

Alternatives to Full Wafer Exposure:

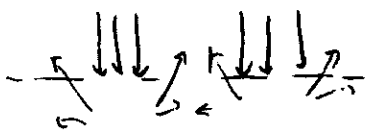
(i) Direct exposure of resist on wafer using an Electron Beam. The beam can not only expose, but can using a variety of techniques, sense a pattern previously produced, thus doing local alignment corrections.

(ii) Exposure using masks, probably projection of mask, but of less than full wafer. Step & Repeat with local alignment on the wafer to cover entire wafer.

When $\lambda < 0.5\mu m$, Feature Size $< 1.0\mu m$, we no longer can use UV light (wavelength $\geq 0.3\mu m$), since can't resolve features.

- What are the alternatives available? First, we could use an electron beam. Electrons accelerated to moderate energies will expose various resists. We can produce very narrow E-beams ~~< 250 Å~~, but there are problems which will likely limit use of Direct writing with E-Beams to feature sizes $> \sim 0.5\mu m$. Problems:

  > Main problem is scattering of electrons in resist and silicon. The exposure latitude narrows as the spatial period of a pattern is reduced:

  SHOW SLIDE OF CALC EXP.

  USING AS 250Å, 10KeV, resist 0.4μm.
  AT SPACINGS OF 2μm, 1μm, 0.5μm, 0.3μm.

  > Associated Problem: Exposure at any point depends on exp. at neighboring points. This proximity effect requires pattern dependent exposure corrections at small values of λ.

  > As λ decreases, the time to "write" the whole wafer rapidly increases. Time (expense) per wafer gets very large at small λ.

- However, although time/wafer may be long, the direct writing E-beam exposure offers possibility of shorter turnaround than when using masks. No masks to pattern and develop. The E-beam machine can be viewed as a COMPUTER OUTPUT DEVICE. We could have a different chip design in every chip position. IDEAL FOR MULTI-PROJ. CHIP SET IMPLEMENTATION. Especially if integrated into a relatively automated fabrication facility.

- O.K. But how do we do better than ≈ 0.5 μm, and how do we get high wafer throughput in manufacturing at < 0.5 μm feature sizes?

- ONE POSSIBILITY IS TO USE X-RAYS TO EXPOSE THE RESIST.
(MUCH OF THE PROGRESS IN THIS AREA IS RESULT OF WORK BY HANK SMITH AND HIS COLLEAGUES AT MIT LINCOLN LABS.)

WE GET AROUND THE WAVE LENGTH PROBLEM BY USING SOFT X-RAY RATHER THAN UV. TECHNIQUES FOR OPTICAL ALIGNMENT (INTERFEROMETRIC TECHNIQUES) ARE NOW KNOWN WHICH CAN ALIGN TO ≈ 0.02 μm.
XRAYS OF ≈ 100 to 1000 eV range (WAVELengths ≈ 0.01 to 0.001 nm)
SO, WE CAN USE THE STEP & ALIGN TECHNIQUE TO ULTIMATELY SMALL DIMENSIONS USING ORTICAL INTERFEROMETRIC ALIGNMENT AND X-RAY EXPOSURE.

- WE ARE THUS BACK TO MAKING MASKS:

OK on next pg

X-RAYS REQ. A VERY THIN MASK SUPPORT: e.g. MYLAR, UPON WHICH A HEAVY METAL SUCH AS GOLD OR TUNGSTEN IS USED AS THE OPAQUE MATERIAL.

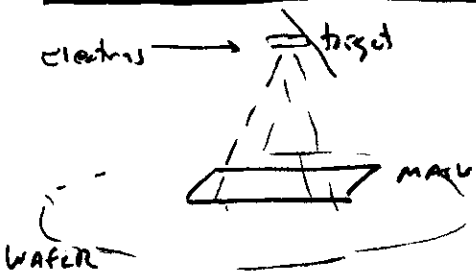
NO BACKSCATTERING OF XRAYS OCCURS. INTERACTIONS OF X-RAYS WITH MATTER TEND TO BE ISOLATED, LOCAL EVENTS. ANY ELECTRONS PRODUCED WHEN XRAY ABSORBED ARE LOW ENOUGH IN ENERGY SO RANGE IS ONLY SMALL FRACTION OF MM.

- SO, PATTERNS PRODUCED BY X-RAYS IN RESIST ON SILICON ARE MUCH CLEANER & BETTER DEFINED THAN THOSE PRODUCED BY ANY OTHER KNOWN TECHNIQUE (SLIDE) SLIDE SHOWS PAD WITH PERIOD OF ≈ 0.3 μm.

- HOW DO WE MAKE THE X-RAY MASKS?

  ANS: USING AN E-BEAM MACHINE!
  (O.K. IF IT TAKES A WHILE TO EXPOSE, SINCE WILL USE MANY TIMES)

- A Problem with X-RAYS:



If use traditional method of producing soft x-rays, have problem of not a "point source" and beam is not collimated. So, if get close to source to get higher intensity, feature resolution is poor. If far away, Exposure times too long.

- A solution: Use a Storage ring (synchrotron) to produce the X-rays. A 500 to 700 MeV electron storage ring shaped as a many sided polygon. Beam deflected at each "corner" using superconducting magnets. Deflection results in a centripetal acceleration of electron and hence in intense tangential emission of synchrotron radiation. The most important component of such radiation is soft X-rays. Could fit one exposure station to each vertex.

  Alignment would be done by an automated optical interferometric technique, on a per chip basis.

  X-ray intensity in such a system is high enough that one layer of our chip could be exposed at each vertex every few seconds. Achievable values of λ in both the feature size and alignment sense are down to $\lambda \approx 0.1\,\mu m$.

- AN OVERVIEW OF POSSIBLE FUTURE ROUTES FROM DESIGN FILES TO FINISHED CHIPS WITH MICRON TO SUB-MICRON FEATURES IS GIVEN IN FIG 27 (SLIDE)

- IN IMMED FUTURE: MANUF OVER NEXT 3-4 YRS:

  PROJ. EXP. WILL WORK DOWN TO ≈ 1-2 μm feature size AND PROJ. EXP + STEP & ALIGN ON WAFER WILL GET AROUND ANY RUNOUT PROBLEMS.

  . THIS ALSO ELIMINATES STEP & REP. IN MASKMAKING. COULD HAVE REASONABLE TURNAROUND IF OPTIMIZED FOR THAT (TREND TO MAKING OPT MASKS BY E-BEAM: SIMPLER, QUICKER, THAN P.G.)

- RIGHTMOST PATH: DIRECT WRITING WITH E-BEAMS:

  WHILE SLOW/WAFER PATTERNED, NEVERTHELESS PROMISES THE ULTIMATE IN SHORT TURNAROUND. A LIGHTLY LOADED HIGHLY AUTOMATED PROTOTYPING FACILITY WOULD BE ULTIMATE FOR QUICK DESIGN DEVELOPMENT. WORKABLE DOWN TO $\lambda \approx 0.3\,\mu m$ Features 0.5 to 0.6 μm

- CENTER PATH: POSSIBLY THE ULTIMATE MANUFACTURING

  PATH FOR DENSEST SYSTEMS. CLEARLY WORKABLE FROM PATTERN FEATURE SIZE & ALIGNMENT STANDPOINT DOWN TO $\lambda \approx 0.1$ μm. (Feature 0.2 μm) i.e. to limiting dimensions.

[NOTE: Process Tech.; Device Circuit & System Design Methodology must evolve with all this.]

- Could imagine: Prototyping Design in Q.T. environment at $\lambda = 0.5\,\mu m$, then manufacturing full systems at $\lambda = 0.2\,\mu m$.

6.978

LECTURE #17. NOVEMBER 16.

TODAY: MEMORY CELLS & SUBSYSTEMS

PROJECT LAB: A NUMBER OF ST. WANT TO USE LAB THIS WEEKEND. I'LL BE IN BOTH SATURDAY & SUNDAY FROM ~10+ TO ~4+.

- ALSO, OFFICE HOURS TMRW FROM ~11 TO ~4.
- HOW MANY MIGHT BE INT. IN USING LAB OVER THANSGIVING? IF A FEW WHO CAN COOP. ILL GIVE KEY TO ONE. MAIN THING: CAN'T LEAVE OPEN & UNATTENDED DURING OFF-HOURS.

[ALSO: THE LAB ASSISTANTS COULD STAY LATER (SAY ~9) IF SOMEONE COULD REMAIN IN LAB ~6 TO 7 SO THEY CAN GET DINNER.]

MEMORY CELLS & SUBSYSTEMS

- These presently get lots of attn because of the huge market and money being made manufacturing memory chips.

- Presently, general purpose computing done by Von-Neumen type machines, where memory is made distinct from processing in the hardware [CPU] ⟷ [MEM].

  So much effort is made to increase speed (reduce T_p) of CPUs, and to increase size and reduce T_m (cycle time for mem) of memory. There is an insatiable demand for denser, faster, cheaper memory.

- Some Things To Keep in Mind During Today's lecture:

  (a) Present extreme emphasis on memory chip design is result of present computational environment. This may change in the next decade.

  (b) As the interior device sizes within CPU & MEM are decreased, a higher & higher proportion of the energy & time cost per computation is due to the energy & time to transport info from MEM ⟷ PROC.

(c) Because of the #'s of best LSI designers preoccupied with Memory cell design, it is a CLASSICAL ENGINEERING ART.

All these people are working on variations of just a few cell types, and the alternatives have been explored thoroughly.

Since whole chip is memory, the process, circuit, and subsystem designs have been simultaneously optimized. Even without much computer aids, this is possible because the cells are simple, and subsystems very regular.

(d) To optimize designs for min delay & pwr dissipation will require the use of electrical simulation.

(e) The ultimate 1-T memory cell which I'll show requires very clever (experienced) circuit design in its interface and support circuitry.

- So this is a diff. design environment than that discussed in this course. Rather than doing lots of designs per unit time, designing hierarchically to build big systems, and using design methodology constraints to keep out of trouble, MEMORY DESIGN involves highly optimized lower level process and circuit design.

- Note also: Changes in way we might do computing will be discussed during week of Dec 4-8:
  DEC 5 Carver Mead (H.Com.Proc.) Dec 7 Wayne Wilner (Rec. machines)
  DEC 8 Carlo Sequin, U.C.B., X-TREE [How many can come?]

- To Prepare for that week: Read CH8 p1-9, p31-32, p57; Skim p33-56. (notice res-ltc)

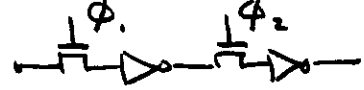
  Also: Make a copy of, and at least skim through the recent highly important paper by John Backus of IBM - "Can Programming Be Liberated from the von Neuman Style? A Functional Style and Its Algebra of Programs"
  CACM AUG '78 V21 #8

- WE'VE USED SEVERAL WAYS SO FAR TO STORE INFO:

  EX: • The Shift Register, to make serial memory
  • The DM Register cell, to make randomly accessed (by word) memory array.

- LET'S EXAMINE THE DENSITY, PWR OF THESE and see how they compare to available memory chips.

- THE SHIFT REGISTER: Recall Fig 8b, CH.4 for layout:

  one bit of storage requires two SR cells: 

  Area $\approx (21\lambda \cdot 2) \cdot 19\lambda =$ 126 μm x 57 μm

  '78 λ = 3 μm use all 3 μm lect. th.

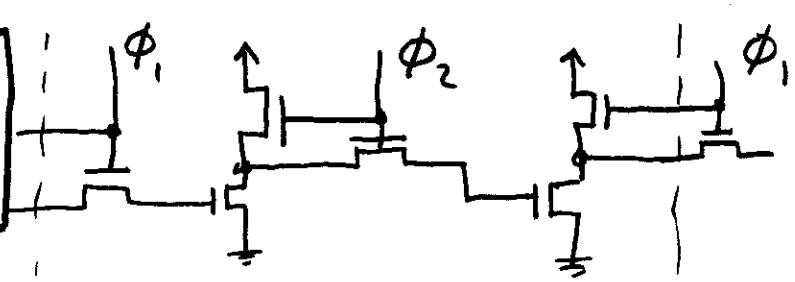
- How many Bits could we place in a big chip, say 4 mm x 4 mm? also use for core + Reg

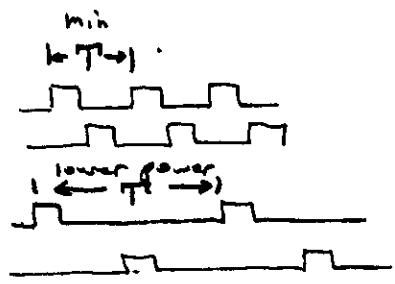
  # BITS $\approx \frac{4000^2}{126 \cdot 57} \cong$ 2K bits

- Remember we previously calculated a power dissipation of ~ 8 to 12 w/cm². This was rather conservative (high) and also could be reduced by using longer pullups, narrower pulldowns to perhaps 3 to 6 w/cm². Still a bit hot.

- There's really no way to reduce cell area much. But there is a way to reduce pwr: trading pwr red. against increased delay: USE ENH MODE PULLUPS (may req longer ratio than 8:1), and CLOCK the PULLUPS:

6 Transistors per BIT of memory



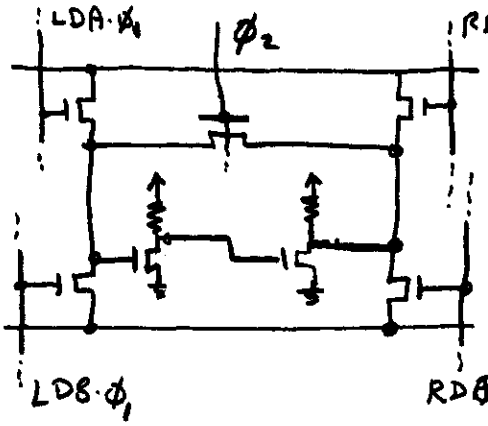


NO STATIC PWR DISS WHEN CLOCKS OFF! IF LOW DUTY CYCLE, THEN LOW POWER (BUT LONG T compared to MIN T)

(SEE THE TI BOOK FOR MORE SR TRICKS!)

- So the SR isn't really very dense. It is, however, extremely easy to interface as a subsystem for storing small amounts of info.
- How do we get more density, especially without incr pwr/area substantially? We use RAM:
- LETS BEGIN WITH THE OM REG. CELL, Then progress thru series of denser RAM cell designs
- OM REGISTER CELL:

IF 2-BUSSES:	9-T's /BIT
IF 1-BUS	7-T's /BIT



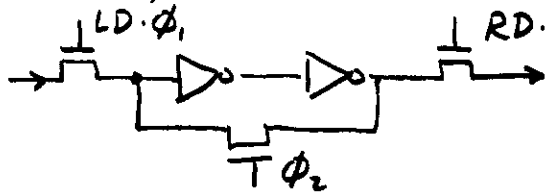
Busses Precharged. Reading just has to pull down. So fast.

NOT STATIC

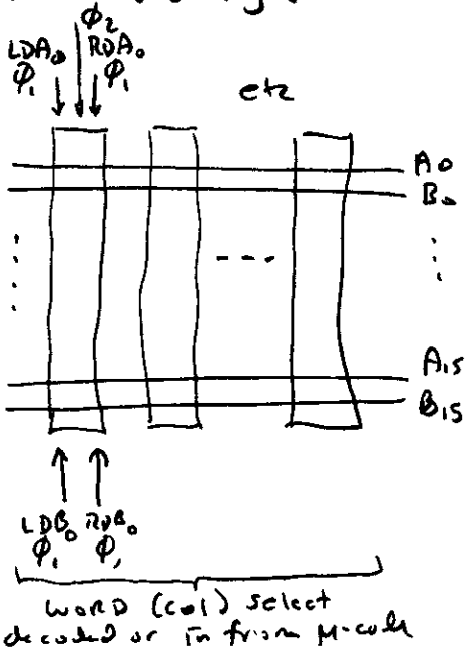
Basically very simple. Really like SR which feeds back on itself:

Loading into C_G only

Reading from Inv output only



So Very easy to make a memory subsystem which is directly interfaceable with, compatible with our 2-Φ clock scheme: No clocking tricks needed: For example the 16-16bit 2 port REGISTER SUBSYSTEM IN OM-2



DENSITY: CELL IS ~ 54λ x 48λ

∴ In 4x4mm: $\frac{(4000)^2}{162 \cdot 144} \approx$ 700 BITS

EVEN LESS THAN SR

Even if used one Bus (7-T's/cell) could only get ~ 1K in 4x4mm.

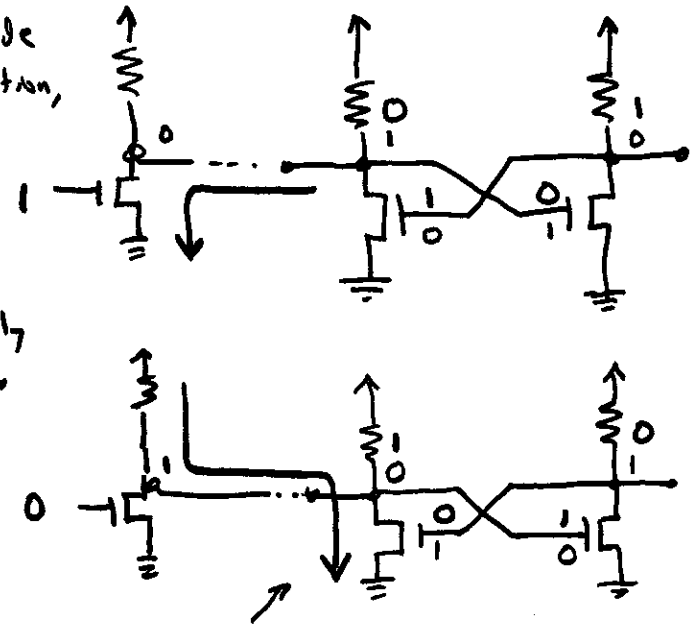
NO Problem with PWR.
BUT MUST BE CLOCKED

HOW TO MAKE A STATIC RAM

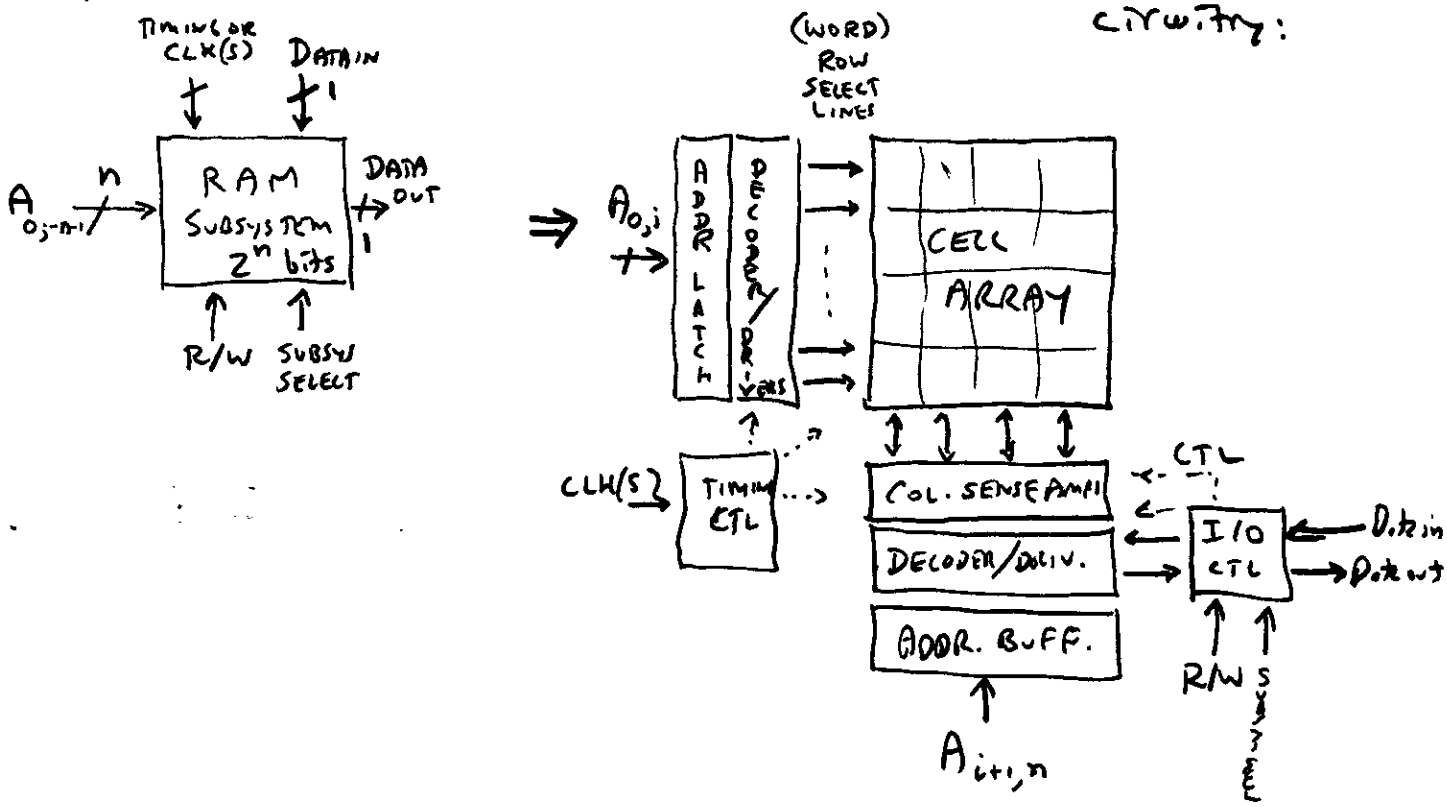
THE SIMPLEST STATIC RAM CELL IS BASED ON CROSS-COUPLED INVERTERS:

Same as our Reg cell, but no clocked pass-T in feedback path:

- If either input/output node is pulled down, by ext. action, other side turns off, latching the pulled down side ON.
- Will hold state indefinitely (STATIC) unless power goes off.
- Note: Ratios have to be right: external driver has to have width pull-up in order to source enough current to raise input node above threshold. Usually use double rail input



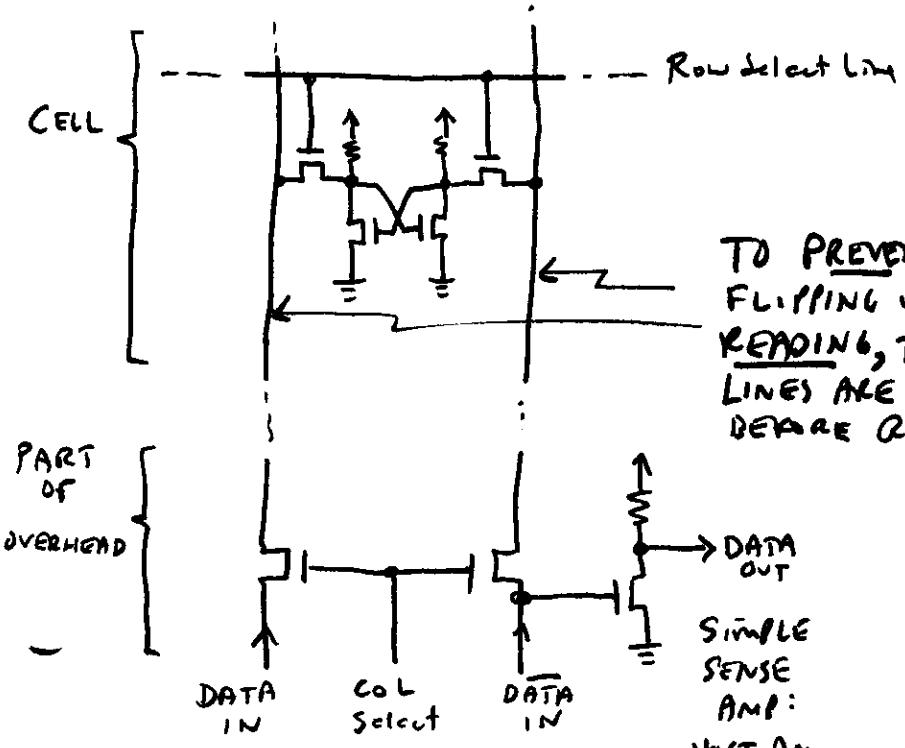
IN GENERAL: MAKING RAM SUBSYSTEMS: There is overhead circuitry:



VERY IMPORTANT POINT ABOUT OVERHEAD CIRCUITRY:

- There is some fixed overhead for given type of memory cell
- For Address size n, cell array area goes as n^2 but overhead usually goes as n ... to n ln n.
- So, "bigger" memories require less fractional area in overhead.
- HOWEVER: The trickier, dynamic memories which have increasing density, smaller cell sizes, require greater fixed overhead.
  - > no such thing as a free lunch. i.e: can't get small memory subsystems with ones/bit on order of the 16K RAMs, because of fixed overhead.
  - > So, for small RAMs use REG CELLS, OR AT LEAST STATIC RAMS - Easier to interface.

THE 6-T STATIC RAM:



Area: Maybe 125 μm x 125 μm

So: $\frac{(4000)^2}{125 \cdot 125} = (32)^2 = 1K$

per 4mm x 4mm

Not counting OVERHEAD

TO PREVENT FLIPPING WHILE READING, THESE LINES ARE PRECHARGED BEFORE READING

i.e. ~ same as single bus REG CELL.

If on-chip subsystem, could READ/WRITE WHOLE WORD AT ONCE

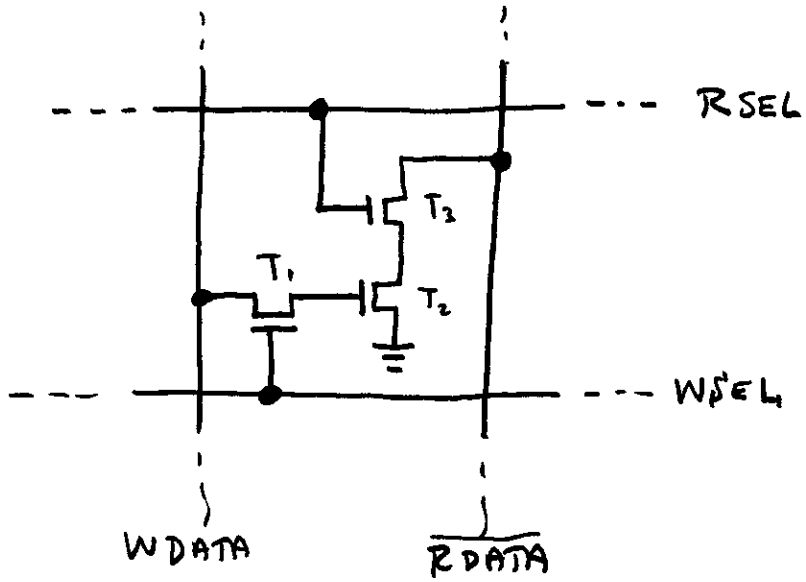
NO PWR/AREA PROBLEMS SINCE NOT DENSE ENOUGH! (IN BITS/AREA)

- **HOW DO WE MAKE DENSE RAMS?**

WE USE FORMS OF DYNAMIC RAM WITH FEWER T's AND WIRES PER CELL (NOT REALLY #T's That count but # WIRES, ALTHOUGH # WIRES ROUGHLY ∝ #T's)

- **The 3-T Dynamic RAM Cell:**

We've used charge stored on gates in our SR's, and in our clocked inverter REG cells: Let's make a RAM cell that uses this effect more directly:

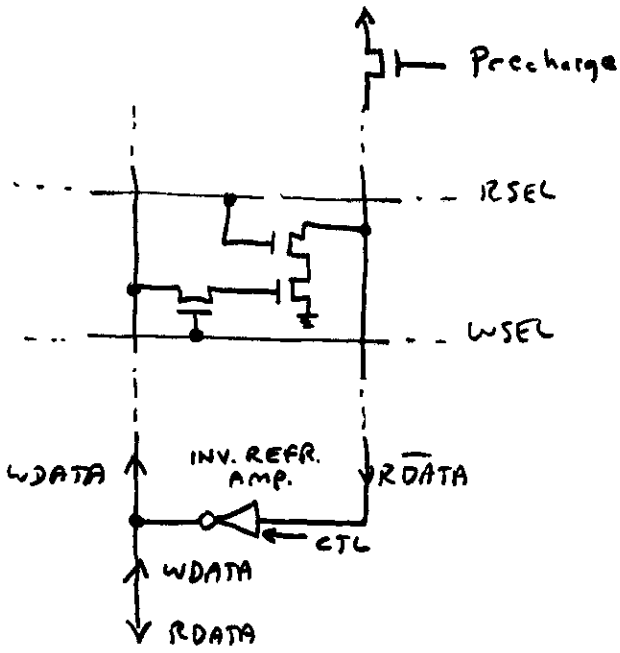


- 2-Select Lines, 2-Data lines per cell.
- Data stored on gate of T_2
- TO WRITE: Put data in onto WDATA. Raise WSEL for a while, then lower.

- **TO READ:** > RDATA is Precharged high.
  - > RSEL is then turned on. The RDATA line is then discharged only if input to T_2 is high.
  - > Note: Output is complement of input
  - > Note: The Read is non-destructive
  - > Note: The cell must be periodically refreshed.

3-T RAM (cont.)

- Lets Add some of the cell's overhead circuitry (see TI p. 125)



A Possibility:

- Read Out data for entire ROW Selecting appropriate column to route to output.
- REQUIRES ONE REFRESH AMP PER COLUMN.

- REFRESH BY RSEL, WSEL ON GIVEN ROW, TURNING ON ALL REFRESH AMPS AT ONCE. MUST DO THIS FOR EACH ROW EVERY FEW MILLISECONDS, REQ ADDIT. EXT. CTL.
- TO WRITE: SIMPLY SELECT COLUMN, PUT DATA-IN ON WDATA, AND SELECT ROW'S WSEL.
- THERE ARE MANY VARIANTS ON THE 3-T DYN RAM. SEE TI BOOK. FOR EACH CIRCUIT VARIANT THERE ARE MANY STICK'S, FINALLY MANY LAYOUT ALTERNATIVES. (NOTE ADJ COL. OR ROWS CAN SHARE A GND RETURN)
- DENSITY:

Conservatively: 60 μm × 60 μm per cell. So, # in 4mm × 4mm (not counting overhead)

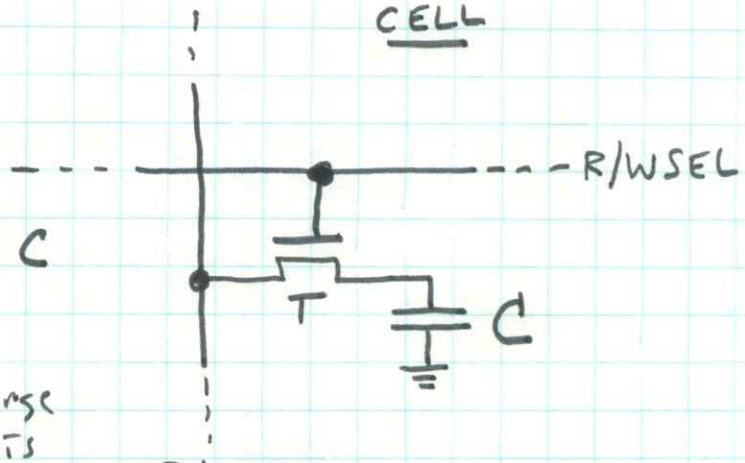
is: $\frac{(4000)^2}{(60)^2} \approx$ **4K bits**

- PWR: No problem: Switching power, and short dissipation thru T_2 -- T_3 of energy stored on RDATA.

4 Times as Dense as STATIC RAM. ≈ SAME PWR/AREA.

SUPPOSE THIS IS STILL NOT DENSE ENOUGH!

THE ULTIMATE (AT LEAST NOW) IS THE SO-CALLED
1-T DYNAMIC RAM



- Really two Poly over Diff Regions: one to form a switch, and a larger one to form a Capacitor C
- Works by simply storing charge (or lack of) on C when T is on. Later, sense charge on C by closing T again and seeing what happens to Data line.
- **Major Problem:** If R/W DATA line (and sense amp inputs) have capacitance C_L, then when close T to read, the voltage on C divides:

  Ex: If C_L low, C holds VDD, then a 1 on $C_L \approx VDD\left(\frac{C}{C+C_L}\right)$

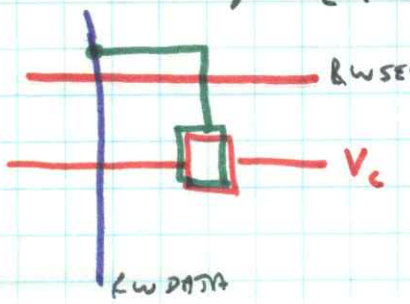
- In Real RAMs $C_L >> C$. Maybe by x20, x50, x100! So this gets really hairy!

> Need special form of sense amplifier circuitry.

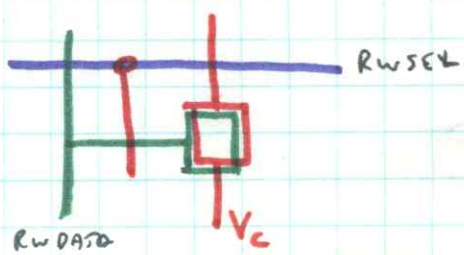
> Also, Readout is DESTRUCTIVE, & MUST Rewrite after every read, not just to refresh.

- **LAYOUTS**: Tricky: I think that to eliminate a Poly to Red contact, the Capacitor input is on Green, & other side goes out on Red to some voltage $V_c \neq GND$. 2 Possibilities:

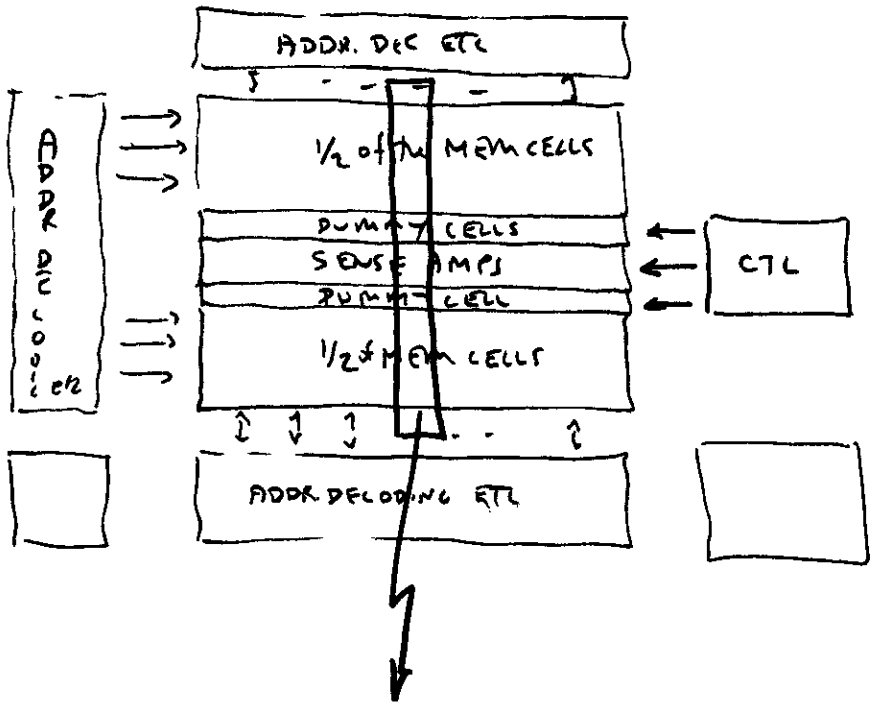
(a)



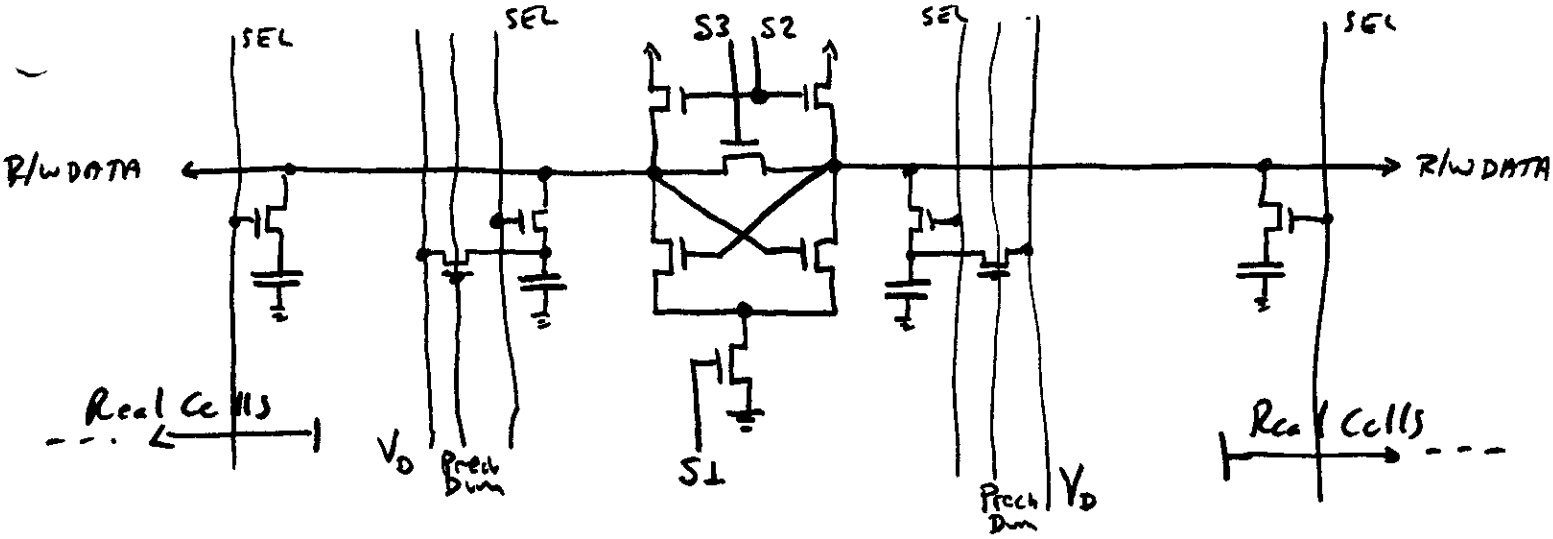
(b)



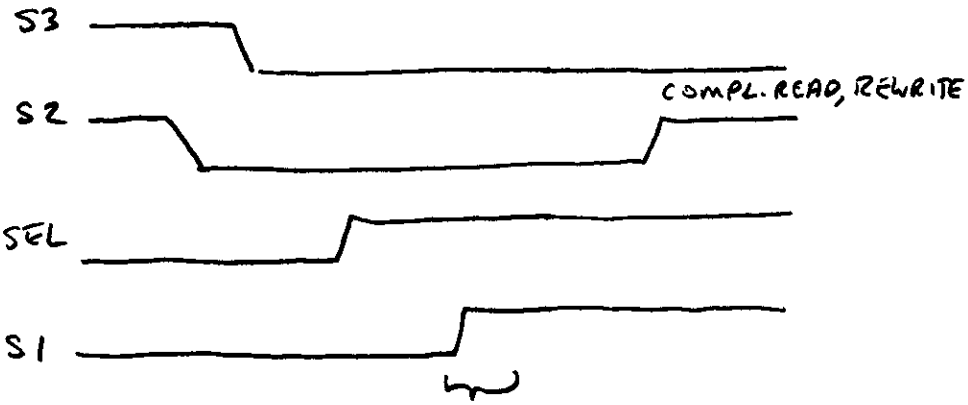
POSSIBLE WAY TO SENCE / REWRITE: ORGANIZE SUBSYSTEM FURTHER AS FOLLOWS:



TILT OVER ONE COLUMN OF THIS ARRAY:



- Dummy Cells Precharged to some V_D then to READ/REWRITE:



- when S2, S3 on, S1 OFF everything prech. high
- whichever s.th higher on SEL, other s.th discharges faster!
- when S2 comes back on, value is locked-in and can be rewritten

- SO, VERY TRICKY DESIGN. TRY TO MAKE C BIG (BUT CONFLICTS WITH SMALL CELL SIZE), MAKE C_L SMALL, MAKE SENSE AMPS THAT WORK WHEN $C_L >> C$.

  VERY HIGH OVERHEAD DESIGN COMPLEXITY. BUT LOOK AT DENSITY: CONSERVATIVELY: '78 $\lambda = 3\,\mu m$

- CELL SIZE $\approx$ 30 μm x 30 μm, SO IN 4mm x 4mm:

  $\frac{(4000)^2}{(30)^2} \approx$ **16 K bits** not counting overhead.

- Actual 16K RAMS MADE WITH λ ON ORDER OF ~ 2.5μm (5 μm wires). The new 64K rams will be similar and use λ ~1.25μm or a little smaller, (i.e. ~ 2.5 μm wires).

IF TIME: INTRO SOME MATL IN CH 8:

- Delays caused by huge relative capacitive loads. Applying the theory we developed in CH 1, we might think of organizing our memories hierarchically, rather than just making "longer wires" (SLIDE 3)

- If do this, we may pay a penalty / bit in that area / bit increases as the Branching Ratio α decreases (i.e. we branch more often). (SLIDE) FIG 5

- But analysis shows that: Area-Time Product has a minimum for some value of α (SLIDE) FIG 6

- AND THAT ENERGY / ACCESS: Also has min at some α. (SLIDE) Fig 7

- WE BELIEVE THAT THIS EMERGING THEORY CAN BE APPLIED TO "SMARTER" MEMORY STRUCTURES, TO MIXED MEM-COMP, AND MAY HELP PROVIDE A BASIS FOR A THEORY OF COSTS OF COMPUTATION IN HIGHLY CONN. SYSTEMS. MORE WHEN CARVER PROBABLY MISSES IN DEC 5

SEMINAR : NOVEMBER 21

ON BOARD: 6.978. TODAY: SEMINAR BY RICHARD LYON, M.R.S. X RPARC
" VLSI IMPLEMENTATION OF SPEECH PROCESSING FUNCTIONS".

- First: Project Info / Status: If you have plots or more details on your project - hand in after Seminar.
- A number of projects are far enough along to be likely candidates for implementation. Please keep me informed of your exact bounding box dimensions as you determine or change them. I'm building a tentative map of the project chip.
- I'll begin allocating space on the chip starting early next week. If you want to put your project on the chip, see me, show me evidence of your progress, and size of project. Late next week we'll send preliminary files to PARC for plotting - getting the plots back here for checking. Final Design files should be ready ~ 5 December.
- ~~Be prepared to get new versions of library cells as discussed or correction print changes - just that we might find slight errors ---. We'll [illegible].~~
- Digitize so that Box centers and edges fall on 1/2 micron grid and no finer. i.e. numbers in centimicrons should end as ---50.
- I'll be in today & tomorrow. I'll be away during Thanksgiving. If group want to use 1.6 - organize and see me tomorrow for key.
- Note: My home # is 494-8188. During next few weeks feel free to call me at home about projects -- questions, info, status, etc.
- WE'VE TESTED THE ARPANET FOR SHIPPING FILES TO PARC -- HAVE PLOTS BACK FOR FILES OF DESIGNS BY GUY STEELE, RICHARD STALLMAN, JOHN [illegible].

TODAY: I'm very pleased to introduce Dick Lyon, who is a Member of the Res. Staff at Xerox PARC.

Dick is Xerox Corporation's Principle Investigator in the areas of Speech and Signal Processing.

His subject Today:

"VLSI Implementation of Speech Processing Functions"

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

Memorandum

To: The IC Group (and others interested)

From: Lynn Conway

Subject: Seminar Announcement

On Tuesday, November 21, from 1:30-3:00 p.m. in room 39-400, Richard Lyon, Member of the Research Staff, Xerox PARC will be giving a seminar titled:

VLSI Implementation of Speech Processing Functions

Abstract:

The cost of implementation of special-purpose signal processing hardware has been rapidly decreasing in recent years under the influence of new technologies. Very Large Scale Integration now enables the design and construction of digital filters and other signal processing structures at costs low enough to spur an applications revolution.

At Xerox PARC, speech recognition has been the motivating application for the development of several new integrated subsystems for speech processing. The predicted low replication costs for VLSI systems has had a major impact on the design of the speech processing algorithms, and on the architecture and design of the integrated hardware subsystems which implement the algorithms.

An overview of the Xerox PARC speech recognition project will be presented. The design of some of the integrated speech processing subsystems will be described, including the design of a single chip digital filterbank and its on-chip memory subsystem.

SEMINAR NOVEMBER 28

[NOTE: CONTACT OTTEN, CHERRY, FOR PRES. OF THEIR PROJECTS NEXT TIME. ALSO: SNYDER: DIMENSIONS? FILE TO PART?]

- TODAY WE HAVE ANOTHER VISITOR, WHO WILL BE GIVING A SEMINAR. BEFORE WE BEGIN, I'D LIKE TO BRING YOU UP TO DATE ON THE PROJECTS & ALSO, NEXT TIME WE WILL BE SPENDING THE ENTIRE LECTURE ON PROJECTS - SOME STUDENTS WILL BE DESCRIBING THEIR PROJECTS - AND I'LL BE COVERING LOTS OF LAST MINUTE DETAILS RELATED TO THE PROJECT CHIP.

- DESCRIBE STATUS LIST (ON BOARD).

- WILL SHIP FILES TO PARC OVER ARPANET TODAY, BET 5 & 6 PM, FOR PLOTTING. THEY'LL BE BACK IN TIME FOR CLASS ON THURSDAY. ALL THE CATEG #1 FILES, & ANY OTHERS IF YOU SEE ME AND LET ME KNOW STATUS / FILENAME.

- HAVE ROOM FOR CATEG 1, 2, + a couple of more. WILL COMMIT AREA AS I SEE PLOTS, GET FINAL DIMENSION

- HAVING A PLOT COME BACK THURSDAY THAT LOOKS OK WILL BE GOOD WAY TO INSURE YOUR PROJECT GETS ON.

- LIBRARY CELLS: MUST MAKE CHANGE IN PAD VDD:

  SEMICOLON MISSING: (12 ITEMS) ;
  LNG16; BOX - - - ;

  optional: one of the boxes is slightly too small i.e. looks weird if plotted on large scale.

  optional: SHOW PAD IN:

  You could simply replace PAD VDD, PAD IN with the new updated versions in <LSI.CELLS>

- ELECTRONICS: YOU MIGHT FIND THE RECENT ISSUE OF - - -

PUT ON BOARD:

PROJECT STATUS AS OF NOON, TUES. NOV 28:

1. COMPLETE / NEARLY COMPLETE. HAVE SEEN PLOT. FINAL DIMENSIONS KNOWN. COMMITTED TO GO ON CHIP:
   - ✓✓ LOGO — MIT.CIF
   - ✓✓ STEELE — (STEELE.CIF)
   - ✓✓ CHERRY — PROJ.CIF
   - ✓✓ OTTEN — CLOCKMASTER
   - ✓✓ LAM — LAM.CIF
   - ✓✓ STERN — PROJ.CIF
   - ✓✓ FRANK — WPLA.CIF
   - ✓✓ YANG — SORTER.CIF
   - ✓✓ COLN — DRA.CIF
   - ✓✓ ROYLANCE — ZZZZZZ.CIF
   - ✓✓ LEVITT — F.F
   - [redacted]
   - ✓✓ PEREA — BYTELSI
   - ✓✓ BROCK, BOUGHTON, BRYANT, LEUNG (DATA MANIP.) — BROCK.CIF
   - ✓✓ HIRATSUKA — ~~COMPAR.CIF~~ HIRATSUKA.CIF

2. NEARLY COMPLETE. APPROX DIMENSIONS KNOWN. LIKELY TO GET ON CHIP, IF FINISH IN TIME.
   - ✓ WOLSON — OLSON.CIF
   - ✓✓ SNYDER — (SNYDER.CIF)
   - ✓✓ SHAVER — MFM.CIF
   - ✓✓ BAIN — PROJECT
   - ✓✓ BALDWIN — NTS.CIF
   - ✓✓✓ WESTBROOK, GOLDIKENER (E-TESTS) — <LSI.GOLDIKENER> ROUGH [redacted]

3. WORK IN PROGRESS. MIGHT GET ON CHIP
   - [redacted]
   - ✓✓✓ FRAEMAN — PROJECT.CIF
   - RUBENSTEIN, AZOURY, BOWEN (CFT PROJ).
   - ✓✓ FRANKEL — (FRANKEL.CIF) —

4. INSUFF. INFO: ✓✓ RUBENSTEIN, AZOURY, BOWEN (CRT CONT.) <LSI.RUBENSTEIN> CRT
   - REYNOLDS
   - McCLELLAN

W = *SENT TO PARC

- TODAY I'M VERY PLEASED TO INTRODUCE

RICH DAVIES, MEMBER OF THE GEN. STF, XEROX PARC,

RICH WILL BE GIVING A SEMINAR ENTITLED:

I THINK YOU WILL FIND THIS MATERIAL QUITE INTERESTING: WE'LL BE GOING BACK AND TAKING A CLOSER LOOK AT THE FET, AND HOW IT WORKS,

AND ALSO AT WAYS OF EXTRACTING SIMULATION PARAMETERS FROM ACTUAL MEASUREMENT ON WAFER, SO AS TO PROVIDE BETTER "CALIBRATED" SIMULATIONS.

LECTURE #18 NOVEMBER 30

- FINAL PROJECT DETAILS / PLANS
- PRESENTATIONS (~10 to 15 min each) of 4 projects:

DAVE OTTEN: BUS INTERFACE CLOCK/CALENDAR
JIM CHERRY: TRANSFORMATIONAL MEMORY ARRAY
STEVE FRANK: WRITEABLE PLA
GUY STEELE: SMALL INTEGRATED MICROPROCESSOR FOR LISP EXPRESSIONS (SIMPLE).

ANNOUNCEMENTS:

- PROJECT REPORTS: O.K. TO SLIP TO TUES. DEC 12, BUT PLEASE, NO LATER THAN THAT.

  TRY TO KEEP COMPACT, BUT MAKE TUTORIAL AS POSSIBLE - SHOW ERNESTO PEREA'S. (MENTION MULT. PARTNER PROJ)
- CRAIG OLSON'S SUGGESTION: MAY EDIT AND GROUP CTR PROJ. RPTS INTO A REPORT ON THE COURSE, ESPECIALLY SINCE SO MANY GUEST. PROJS. HAVE BEEN DONE-

  SO: WRITE FOR PERHAPS A WIDER AUDIENCE. IF IN SPEC FIELD, FOR EX, DIGITAL FILTERING, GIVE REFS RATHER THAN EXT. TUT. ON DIG FILTERING.

  INCL. COMMENT ABOUT WHAT YOU LEARNED FROM ACTUALLY DOING THE PROJECT WOULD BE INTERESTING. ALSO, MENTION/DESCRIBE ANY DESIGN AIDS YOU BUILT YOURSELF.

- REMEMBER: SEMINAR SERIES NEXT WEEK.

  HANDOUTS. ALSO --- I'M GOING TO TRY TO ARRANGE ~~ORGANIZE A~~ GET TOGETHER N 5 OR SO --- MAYBE AT THE FACULTY CLUB -- WOULD REALLY LIKE TO HAVE STUDENTS COME TO THAT --- TALK INF. WITH CARVER.

(SOME SAMPLE MSG TRAFFIC)

DATE: 29 Nov 1978 4:52 PM (WEDNESDAY)
FROM: LYON
SUBJECT: MIT PROJECTS
TO: CONWAY, FAIRBAIRN, ABELL
CC: LYON

HERE IS STATUS OF PRELIMINARY DESIGNS RECEIVED HERE:

SUCCESSFULLY CONVERTED AND PLOTTED:
BAIN (1 MISSING SEMI)
BROCK
CHERRY (STUFF AFTER END)
COLN
FRANK (STUFF AFTER END)
FRANKEL "
GOLDIKENER
LAM (MISSING SEMIS)
LEVITT
OLSON
OTTEN
PEREA (STUFF AFTER END)
RUBINSTEIN (MANY ERRORS)
SHAVER
STERN
YANG (STUFF AFTER END)
(NO COMMENT DOES NOT IMPLY NO ERRORS)

MESSED UP DATA; CONVERTED BUT UNUSABLE:
ROYLANCE (ALSO MISSING A SEMI)
STEELE (SEVERAL ERRORS)

BOMBED FIS TO SWAT:
BALDWIN
FRAEMAN
HIRATSUKA

ALL ITEMS (NO SYMBOLS): CONVERTED, BUT TOO BIG FOR ICARUS
SNYDER

MANY PROJECTS HAD MORE THAN ONE SYMBOL CALL AT
THE TOP LEVEL, WHICH IS VERBOTEN (ESP. SNYDER)!

LYNN, PLEASE SEND BOUNDING BOXES AS SOON AS KNOWN.

DICK

@LOFO\O\FGO

LOGOUT JOB 21, USER CONWAY, ACCT 1, TTY 24, AT 11/29/78 1711
  USED 0:0:13 IN 0:5:34
!@NE@KE@(FTI)^

^LENET
@@3 DL5

TERMINAL=

@ID ;415XR/PARC
PASSWORD =

415 XR1 CONNECTED

PARC DLS #35
>MAXC1...

 PARC-MAXC TENEX 1.34.19, MAXC1 EXEC 1.54.10, 6 JOBS, LOAD = 0.07
@LOG CONWAY 1
 JOB 17 ON TTY4 30-NOV-78 05:20
 PREVIOUS LOGIN: 29-NOV-78 17:05
 [YOU HAVE NEW MAIL]
@READMAIL.SAV;21
*
DATE: 29 NOV 1978 1713-PST
FROM: FAIRBAIRN
SUBJECT: CHIP PROCESSING
TO: ABELL, CONWAY, LYON, TRIM, FAIRBAIRN, ROWSON, BALDWIN,
TO: WILNER, M-NEWELL, STROLLO, SUTHERLAND

ANOTHER FALSE HOPE... HP LOVELAND DIVISION CANNOT PROCESS OUR
CHIP BECAUSE IT DOES NOT REALLY HAVE THE RIGHT PROCESS (THEY SAY). TE DEER
CREEK FACILITY DOES HAVE THE PROCESS AND CAN PROCESS THEM IN 2 TO 3 WEEKS. THE
CHRISTMAS HOLIDAYS REPRESENTS A PROBLEM TO THEM.

I WILL TALK WITH MEC TOMORROW TO GET A FIRMER IDEA FROM THEM ON
TURN AROUND TIME. MY FEELING IS THAT WE CAN WORK MORE CLOSELY
 WITH THE MEC PEOPLE AND THE INTERFACE WILL BE SMOOTHER AND MORE PREDIATABLE.
I WILL INVESTIGATE BOTH ALTERNATIVES MORE THOROUGHLY TOMORROW AND WE
WILL TRY TO REACH A CONCLUSION TOMORROW.

DOUG

Date: 29 NOV 1978 2058-EST
From: GLS at MIT-AI (Guy L. Steele, Jr.)
Subject: Sigh
To: Lyon at PARC-MAXC
CC: GLS at MIT-AI, Conway at PARC-MAXC

LL be ready to ship another file in an hour or two.
Will that be soon enough? I'll try to find the other file also.
-- Guy

Date: 29 Nov 1978 6:04 pm (Wednesday)
From: Lyon
Subject: Baldwin bug
To: Conway
cc: Lyon, Abell, Fairbairn

Lynn,
The problem with Baldwin et al is a comment length limitation in FIS
(signal 5001B, string bound error).
I removed the comments from Baldwin's, and it parsed OK, but aborted
due to circular def references. I have not tried the others.
Dick

Date: 29 NOV 1978 1841-PST
From: ABELL
Subject: problem with GLS and Roylances CIF files
To: GLS at MIT-AI
cc: Lyon, Conway, Fairbairn

The ploblem appears to be that there exist symbol definitions that
are empty. GLS's design has an empty def as the first one on the chip.
I have changed that to have something in it. It appears to do
better but I haven't tried plotting it yet. Roylance also had
an empty symbol. This was due to a few garbage
characters in the file. I did not try fixing his.
Alan

Date: 30 NOV 1978 0446-EST
From: GLS at MIT-AI (Guy L. Steele, Jr.)
Subject: Sigh
To: Lyon at PARC-MAXC
CC: GLS at MIT-AI, Conway at PARC-MAXC, Abell at PARC-MAXC
CC: Fairbairn at PARC-MAXC

<HOLLOWAY>STEELE.CIF @ PARC is the latest version of my file.
I fixed the empty cell problem (but according to the Mead and
Conway book, an empty symbol is valid in CIF).

@J^J
@
@
@
@
@
@

- STATUS: > Net was down in early evening on Monday.

  > Files were shipped later, successfully

  > Problems with FIS (CIF → ICARUS) program prevented some files from plotting successfully. We're fixing that program.

  > 3 Plots arrived today. (OTTEN, CHERRY, FRANK)

  > more will arrive tomorrow ≈ 10-11:00: including, I believe:

  BAIN, BROCH, COLN, FRANKEL, GOLDFEVER, LAM, LEVITT, OLSON, PEREA, RUBENSTEIN, SHAVER, STERN, & YANG

  > FIS problems with

  ROYLANCE, STEELE, BALDWIN, FREEMAN, HIRABUKA.

  > ICARUS problem with SNYDER

- DIAGNOSTICS OR CORRECTIONS: SEE ME FOR DETAILS

  BAIN: MISSING SEMICOLON
  BROCH: IN UPPER RIGHT RAND. LOGIC: Some pullups have channels not long enough
  CHERRY, FRANKEL, FRANK, PEREA, YANG: STUFF AFTER END (;?)

  OTTEN: IMPLANT IN PULLUPS TOO SHORT

  SNYDER: PUT DS -- DF around whole design, Nancell.

  ROYLANCE: MISSING SEMI AFTER COMMENT.
  LAM: MISSING SEMI
  RUBENSTEIN: MANY ERRORS

MISC.

- FIS: Has a comment length limitation

  Don't know limit, But really big comments bomb it.

  May have been problem with:

  Baldwin, Fraeman, Hiratsuka

- FIS: Doesn't like DS; with nothing in it.
  DF;

  This bombed steele's plot.

- NOTE ASYMETRY IN PADOUT IT LOOKS SYM. BUT ISN'T!
  SLIDE

- EXTEND IMPLANT 1.5 λ Beyond DIFF ∧ POLY in all directions: SLIDE

- Previously: ( ) (;) missing } would plot
  LN --; Box --; }

  even though in error.

  Now it won't! You might find such an error when doing final plots.

- Plans, Contingency Plans:

  > Modify Icarus to increase symbol space by deleting lower window: special version for project merging

  > Contingency plan: use TECO to premerge some or many design files by placing PADCELL LIBRARY IN FRONT, appending files, and deleting definitions of PADCELLS within each appended file. Place delete definition statements between files for #'s ≥ 100. Note: This requires delete definition portion of FIZ to work.

  > Contingency Plans: Do two separate mergings of the two individual chip types in the set, and then merge these at the PG level.

  > MUST find and fix bug in FIZ which is causing error in plotting: Bounding boxes of symbols are not being correctly calculated, when symbols are mirrored / rotated. Does not affect ICARUS - PG but can't plot projects with this disease. So: Must fix to see merging effects.

  > Contingency Plans:

FINAL RULES OF THE GAME:

SHOW BOARD OF PROJECT PASTEUPS

- PUT COMMENT AT/NEAR BEGINNING WITH YOUR NAME IN IT:
  (PARC FILE = LNAME.CIF);
- PUT ALL LIBRARY SYMB. DEFS NEXT. (# ≤ 99)
- PUT YOUR SYMB. DEFS NXT. USE #'S ≥ 100
- PUT DS, DF AROUND ALL THIS. USE # ≥ 100
- THEN ONE CALL OF WHOLE PROJECT.

  i.e. CALL 151;

  Preferably with no Transformations applied.
- WHEN YOU ARE REALLY FINISHED:

  Get a message to me: pref. a piece of paper, which gives me

  > directory and filename
  > final bounding box with the one call as described above:

    i.e. minx, maxx, miny, maxy

    just as output by CIFTRN
  > sign off on this.
- LAB WILL BE OPEN THIS WEEKEND, SAT & SUN >1PM.

- MUST HAVE YOUR SIGN OFF BY TUESDAY @ CLASS.
- PROJECTS WILL BE POSITIONED TO FINAL PLACEMENT.

  & THEN SENT ON WEDNESDAY MORNING.
- DURING WED/THURS, MERGING WILL BE DONE AT PARC.
- POSSIBILITY OF LAST ITERATIONS OCCURRING (IF FIND SMALL PROBLEMS) ON THURSDAY.

  WILL BUMP OFF PROJECTS WILL OBVIOUS GROSS ERRORS (such as Pads not hooked up --- VDD/GND Shorts etc.) AND REPLACE WITH OTHERS.

- ~~May need help with project merging / transmission Any volun~~ (no --- announce on Tuesday)

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

To: EE/CS department faculty, staff, and students.

From: Lynn Conway, 36-595, X3079.

As a part of course 6.978, a series of seminars concerning VLSI computer system architecture will be held on Dec. 5, Dec. 7, and Dec. 8, as listed below. You are invited to attend.

"Highly Concurrent Systems",
Carver Mead,
Prof. of Computer Science, Electrical Engineering, & Applied Physics,
California Institute of Technology.
Tuesday, Dec. 5, 1:30-3:00, Rm 39-400.

"Recursive Machines: A non-von Neumann VLSI Architecture",
Wayne Wilner,
Member of the Research Staff,
Xerox Palo Alto Research Center.
Thursday, Dec. 7, 1:30-3:00, Rm 39-400.

"Project X-Tree". (see attached abstract),
Carlo Sequin,
Assoc. Prof. of Electrical Engineering and Computer Science,
University of California, Berkeley.
Friday, Dec. 8, 3:00-4:30, Rm 39-400.

6.978. SEMINAR DEC 7

"Recursive Machines: A non-Von Neumann VLSI Architectures".

- undergraduate work at M.I.T.
- earned the Ph.D. in computer at Stanford working under Don Knuth Dr.
- ~~previously~~ Computer architect particularly, Burroughs, one of the architects of the B1700.
- Now mem. Res Staff at PARC, Doing some very innovative work in Computer Architecture.
- The Subject of this Seminar is:

- Projects all successfully transmitted to PARC, successfully merged into ICARUS design file. Now being conv. to PG format and checked --- then on to maskmaking. I'll keep you informed of progress.

6.978 SEMINAR

This is the third in a series of Seminars this week on the subject of VLSI Computer System Architecture.

It is a real pleasure to introduce today's speaker:

Prof. Carlo Séquin of the
Dept of EE & CS at
University of California, in Berkeley

- Carlo earned his Ph.D in Physics at the University of Basel, in Switzerland.
- He then joined Bell Labs as a Memb. of the Tech. Staff, in the early 70's and was one of the Pioneers in Charge Transfer Devices such as CCD's. Carlo is the senior author of a text on that subject.
- In 1977, he joined the Faculty at Berkeley, and has since been very actively involved in both research and teaching in the area of very large scale integrated systems.
- The subject of Carlo's talk today is Project X-Tree.

Carlo - - -

Corrections:
- Synder's: Address Mod Quotients are treated as merged.
- Corby's: " " as trees deleted.
- Leung/Bryant: Deadlock loops in Fifo's.

PROJECT X-TREE

C.H.Séquin, A.M.Despain and D.A.Patterson

Computer Science Division
Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720

The question how future computing systems can best exploit the computational power of forthcoming VLSI components is studied. "X-tree" is one possible answer based on a modular approach in which the basic building block, "X-node" is a single-chip VLSI computer. An unlimited number of these components are organized into a binary tree, which is enhanced by additional links to provide fault tolerance and a more uniform message traffic distribution. This organization overcomes the communications bottleneck of traditional multiprocessor systems, while remaining within the constraints of the limited pin number of the single-chip components.

X-node itself consists of a dynamically microprogrammable processor, a two-level memory hierarchy and a communications switching network, all ulimately to be integrated on one or two high-density VLSI MOS chips. The project is at an early stage of research. The construction of prototypes of the link between nodes and the communications parts of X-node have been started.

Interconnection topology, addressing scheme, routing algorithm, message format and tentative ideas on switching hardware and the architecture of X-node will be discussed.

6.978. FINAL MEETING 12 DECEMBER

- TODAY: FINAL CLASS IN COURSE 6.978
  - > STATUS OF PROJECTS
  - > ACTIVITIES DURING IAP
  - > ~~SOME OPPORTUNITIES~~ LOOKING AHEAD.
  - > READING REFERENCES
  - > TIME FOR QUESTIONS
  - > COURSE FEEDBACK QUESTIONNAIRE
  - > CLASS PHOTOS !

- PROJECT STATUS:
  - > 19 Projects got onto chip set: let me mention so you know for sure!
    one by Brock, Boughton, Bryant; Leung; Cherry; Coln; Frank; Frankel; Miratsuka; Lam; Levitt; Olson; Otten Perea; Roylance; Shaver; Snyder; Steele; Stern; Yang;
    one by Bowen, Azoury; Rubinstein;
    one by Goldikener; Westbrook.
  - > 6 projects didn't make it. These were ones for which I didn't have enough project report info to evaluate, or which were finished right near the last minute. Some of these were very interesting and you might try to get them onto future M.I.T. chip sets.
  - > Things now look quite good for return of completed wafers by early to middle January. I can't promise this, but it will likely happen.
  - > So, that brings us to activities during IAP.

IAP: Assuming that wafers return during early to mid Jan: There will be two major activities during IAP:

1. Packaging & electrical testing by Faculty/Staff in the materials science area.

2. Functional Testing, organized by Prof. Don Allen.

Dimitri Antoniadis will discuss plans/procedures to be used in packaging

Don Allen will discuss plans for electrical testing.

HANDOUT QUESTIONNAIRE FOR INFO

* LETS GET A LIST OF STUDENTS WHO HAVE PROJECTS ON THE CHIP SET WHICH THEY WISH TO TEST:

NAME, WHERE CAN BE REACHED DURING IAP (INCL PHONE), DO YOU WISH TO JOIN JON ALLEN'S EFFORT OR DO YOU PLAN TO TEST INDEPENDANTLY?

WE WILL PACKAGE & BOND UP SEVERAL CHIPS FOR EACH STUDENT IN THIS GROUP

ALSO, NOTE: I WILL FAVOR TESTED PROJECTS FOR WHICH I'VE RECEIVED RESULTS, WHEN I RUN OFF MORE ARTIFACTS (VERSATEC PLOTS, SOLID COLOR PLOTS, etc.). I.E.: LET ME KNOW IF IT WORKED, OR IF NOT, WHAT YOU FOUND OUT WENT WRONG.

Include Arpanet Mailbox if any.

ARTIFACTS: INCLUDE ADDR. WHERE I CAN MAIL ARTIFACTS. I'D LIKE EVERYONE TO HAVE A FEW UNMOUNTED CHIPS - BE SURE TO LOOK AT THEM UNDER VARIOUS MICROSCOPES - AND MORE PLOTS OF YOUR PROJECT --- ESP COLOR PLOTS. IF YOU AREN'T GOING TO BE AROUND IN THE SPRING - SO INDICATE - AND SEND ADDRESS LATER.

MY ADDRESS AFTER FEB 1st WILL BE --- PARC ---

ANY QUESTIONS ON PROJECTS ? IAP ? ETC.

LOOKING AHEAD : OPPORTUNITIES :

Grad. work ; Teaching ; Research ; Development ; Entrepreneurial

[; right now: helping the arena grow : participating in exploring the academic / industrial collaboration.]

[; in helping do this right here at M.I.T. For example, participation in next fall's course.]

READING REFERENCES:

I brought along several of the recommended reading references, and I'd like to comment a bit about them: [You might want to look at these after Class to see if you think they're worth pursuing]

Now that you are really on top of this material, you might want to add a few of these to your library - for expanding your background further into adjacent fields, or for future reference: ⊙ buy rec.

⊙ Sci. Amer: Special Issue on Microelectronics Sept '77. IS available in reprinted hard-cover form.

Very good general background reading, and to show others what this arena is all about. Lots of pictures.

⊙ A.S. Grove, "Physical and Technology of Semiconductor Devices". a bit dense, but still the classic on process technology and device physics. Excellent Reference.

⊙ P. Richman "MOS Field Effect Transistors & Integ. Circuits" Very readable, excellent reference text & tutorial text on MOS-FET's. Excellent supplement to CH. 1.

- Penney & Lau (Eds) "MOS Integrated Circuits". An early general text on MOS-LSI containing useful info. A lot of info on inverter characteristics, etc.
- TI "Semicond. Memory Design & Application". If you're interested in memory subsystems, consult this reference.
- Zvi Kohavi: "Switching & Finite Automata Theory" Good textbook
⊙ Bell & Newell "Computer Structures: Readings and Examples". A classic encyclopedic work on computer architecture. Lots of history and examples. A new edition is about to come out - a recommended buy. Not much if anything on int. circ. / int. syst. but still very interesting.

⊙ AND, OF COURSE: NEXT SUMMER, BE SURE & BUY MEAD & CONWAY

- LET'S TAKE 10 MINUTES -- FILL OUT COURSE FEEDBACK QUESTIONNAIRE.

NOTE:
- CROSS OUT REC. INST. → REPLACE LECTURER: LYNN CONWAY
- PUT N.A. IN T.A.
- PUT N.A. IN ITEMS 11, 12.

- TIME FOR ANY REMAINING QUESTIONS / COMMENTS ABOUT COURSE, FIELD, ETC.

- BEFORE YOU ALL TAKE OFF - I WANT YOU TO KNOW WHAT A GREAT EXPERIENCE THIS HAS BEEN FOR ME - IT'S BEEN A REAL PRIVELIDGE TO TEACH SUCH A FINE GROUP OF STUDENTS.

INDIVIDUALLY AND AS A GROUP YOU'VE ACCOMPLISHED FAR MORE, & IN MUCH LESS TIME, THAN ~~STUDENTS~~ AT ANY OF THE OTHER SCHOOLS. I'M VERY PROUD OF YOUR ACHIEVEMENTS.

I HAVE A FEELING THAT SOME GREAT THINGS WILL BE DONE [redacted] BY STUDENTS IN THIS CLASS - I'D ENJOY HEARING ABOUT YOUR ADVENTURES IN THE FUTURE.

I'D REALLY LIKE TO REMEMBER YOU ALL - AND I'D APPRECIATE IT IF YOU'D LET ME TAKE A COUPLE OF PICTURES OF THE WHOLE CLASS - - -

---

- AFTER CLASS: I'VE GOT REFERENCE BOOKS HERE IF YOU WANT TO LOOK AT THEM, AND ALSO SOME PREVIOUS PROJECTS FOR THOSE WHO HAVEN'T HAD A CHANCE TO LOOK AT ANYTHING.