

# **The MIT '78 VLSI System Design Course**

A Guidebook for the Instructor of VLSI System Design

**Lynn Conway**

Manager, LSI Systems Area, Xerox PARC, and  
Visiting Assoc. Prof. of Electrical Engineering & Computer Science, M.I.T., 1978-79.

In the fall of 1978, a course entitled "Introduction to VLSI Systems" was offered by M.I.T.'s Department of Electrical Engineering and Computer Science. This document is a compendium of information concerning that course. It is intended to serve as a basic guidebook for instructors preparing similar courses.

## **TABLE OF CONTENTS:**

### **I. The Plan:**

- › Abstract of the course.
- › Goals of the course, relevant student backgrounds and interests, and required prerequisites.
- › A detailed outline of the course.
- › Discussion of the sort of class projects required.
- › Design tools and procedures required for class project implementation.

### **II. The Result:**

- › Letter to the participants, indicating the status of the projects.
- › List of the student LSI projects.
- › List of the participating organizations and schedule for chip set implementation.
- › Map of the M.I.T. '78 multiproject chip set.
- › Photomicrograph of the M.I.T. '78 multiproject chip set.

### **III. Lecture Notes:**

- › A complete set of notes for the seventeen formal lectures and for the various other class meetings of the M.I.T. course (see the detailed Outline in part I). The notes were handwritten out in detail, this being the instructor's method of lecture preparation. Note that a few figures were in color in the original notes; hopefully the reader will be able to deduce the correct colors for these figures.

### **IV. Homework and Project Assignments:**

- › Homework was assigned only during the first half of the course, before the project activities began. The homework emphasized material and experiences useful as background for planning and working on projects. Specific project assignments were given to keep the students on schedule during the short, intensive project phase of the course.

### **V. Other Course Handouts:**

- › Student Background Questionnaire; other questionnaires.
- › CIFTran User's Guides.
- › Examples of the Wire-Bonding Maps returned to students with their packaged chips.

## **I. The Plan**

**Massachusetts Institute of Technology**  
Department of Electrical Engineering and Computer Science

Special Subject in Electrical Engineering and Computer Science:

**6.978: Introduction to VLSI Systems (A) (New)**

Prereq.: Limited Enrollment; permission of the instructor required.

Year: G(1)

3-3-6

An introduction to the design and implementation of very large scale integrated systems. The course provides sufficient basic information about integrated devices, circuits, digital subsystems, and system architecture, to enable the student to span the range of abstractions from the underlying physics to complete VLSI computer systems. The course presents basic procedures for designing and implementing digital integrated systems, including a structured design methodology, use of stick diagramming, use of a scalable set of ratioed design rules, use of a symbolic layout language, techniques for the estimating delay times, and use of a starting frame of layout artifacts for organizing multi-project chip sets and conveying them thru implementation. The course also examines the effects of scaling down the dimensions of devices and systems, as will occur with future improvements in fabrication technology.

After developing this overall context, the course examines the interdependance of integrated system architecture, design methodology, and the procedures used for patterning, fabrication, and testing. Techniques for simplifying, systematizing, and reducing the time involved in integrated system design and implementation will be discussed.

The subject matter will be reinforced by actual experience in LSI design: students will complete, through layout, the design of digital subsystem projects containing on the order of several hundred to perhaps several thousand transistors. NMOS will be used as the technology for course examples and projects. Selected student projects will be organized into a multi-project chip set to be implemented by commercial mask and fab firms.

*L. A. Conway*

Tuesday and Thursday, 1:30 - 3:00

## **6.978: Introduction to VLSI Systems**

### *Goals of the Course, Relevant Student Backgrounds, and Prerequisites Required for this Course:*

This is an interdisciplinary course and is likely to be taken by students from a wide range of backgrounds, including CS students primarily interested in computer architecture and system design methodology, and EE students primarily interested in solid state physics and integrated circuit design. The goals of the course are:

- (i) *for CS students* and those interested in computer architecture: to take the mystery out of integrated circuit and system design, explore some of the many CS issues involved in integrated system architecture and design methodology, and indicate the exciting possibilities ahead for the expansion of computing power by VLSI systems.
- (ii) *for EE students* and those interested in device physics and integrated circuit design: to get a feeling for the nature of the chips that will be designed in the future, take some of the mystery out of the design of complex integrated digital systems, and illustrate the additional opportunities for optimization at the system level as opposed to the device and circuit layout level.
- (iii) *for all participants*: to outline the many opportunities now present for major advancements in the state of the art by collaborative efforts involving applied physicists, electrical engineers, computer architects, and computer scientists.
- (iv) to develop and debug a set of basic tools and procedures for enabling the design and implementation of LSI multi-project chip sets at M.I.T.

Ideally, students will have taken the following prerequisite courses, or have the equivalent background experience: Introductory Network Theory (6.011), Electronic Devices and Circuits (6.012), Structure and Interpretation of Computer Languages (6.031), Computation Structures (6.032), Switching Circuits, Logic and Digital Design (6.082), Digital Systems Project Laboratory (6.112). However, those lacking the required prerequisites (including recommended undergraduates) may take the course with the permission of the instructor.

## **6.978: Course textbook**

Carver Mead and Lynn Conway, *Introduction to VLSI Systems*, July 1978; a pre-publication limited printing of a textbook in preparation; distributed to course participants. This textbook will be published by Addison-Wesley Publishing Co., Reading, Massachusetts, in the fall of 1979.

## 6.978: Introduction to VLSI Systems:

**Course Outline:** [The final outline of the actual course.]

### Week #1:

#### Sept. 12: Lecture #1

Administrative Details.

Course Overview.

MOS as an Abstract technology.

Patterned conducting layers separated by insulating material.

Color coding, Switches, Simple logic structures, Some examples.

#### *Homework #1 ( part 1) (Due 19th):*

Stick diagram several functions.

#### Sept 14: Lecture #2

The MOS transistor.

Dimensions, basic function, transit times, current/voltage characteristics.

The basic inverter.

Structure, function, transfer characteristics, logic threshold voltage, intro. to delays.

NAND and NOR logic circuits.

#### *Homework #1 ( part 2) (Due 19th):*

Logic gate vs switch-logic designs for selector.

#### *Reading Assignment:*

Ch.1: Intro., MOS Transistor, Basic Inverter, Inverter Delay, Basic NAND and NOR Logic Circuits.

Ch.2: Intro.

Ch.3. Intro., Notation, Combinational Logic.

### Week #2:

#### Sept.19: Lecture #3

More on delays.

Transit time and inverter pair delay.

Notation.

Two phase clocks.

The shift register.

Shift register arrays.

Relating Different Levels of Abstraction .

Implementing Dynamic Registers.

Implementing a Stack: the basic cell.

#### *Homework #2 (Due 26th):*

More stick diagram-level designs (two-dimensional stack, several PLA examples).

#### Sept.21: Lecture #4

Stack subsystem design, control timing, generation of control signals.

Register to Register Transfer.

Combinational Logic.

The Programmable Logic Array.

Basic concept, Circuit design, and stick diagram of example.

Finite State Machines.

Basic concept, use of PLA form of, symbolic and encoded transition tables, traffic light controller example.

#### *Reading Assignment:*

Ch.3. Two-Phase Clocks, The Shift Register, Relating Different Levels of Abstraction, Implementing Dynamic Registers, Designing a Subsystem, The Programmable Logic Array, Finite State Machines.

## **Week #3:**

### **Sept.26: Lecture #5**

Patterning.

The Silicon Gate n-Channel Process.

Design Rules, basis for and detailed description of.

### *Homework #3 (Due 3rd):*

Finite state machine example, from problem statement to encoded transition table;  
layout of compact shift register cell; layout of stack cell.

### **Sept. 28: Lecture #6**

Review the process, discuss some advantages of Silicon-Gate process over earlier ones.

Explanation of sheet resistivity; estimating ratios and resistances from layout geometries.

Layout ideas.

Current limitation in metal lines.

Go thru some layout examples: selector; shift register.

### *Reading Assignment:*

Ch.2. Patterning, Silicon Gate n-Channel Process, Design Rules, Electrical parameters, Current Limitations in Conductors.

Ch.4. Introduction.

## **Week #4:**

### **Oct.3: Lecture #7**

Driving Large Capacitive Loads.

Space vs Time.

Super Buffers.

Body effect.

Depletion mode vs enhancement mode pullups.

### *Homework #4 (Due 12th):*

Stick diagram and layout of serial bit string comparator.

### **Oct.5 Lecture #8**

Layout of the PLA.

Detailed description of an OM2 subsystem: the Barrel Shifter.

Description of function and structure of the serial bit string comparator (to get people started on HW #4).

### *Reading Assignment:*

Ch.1. Driving large capacitive loads, super buffers.

Ch.4. Patterning and fabrication.

## **Week #5: [Oct.10 is vacation day]**

### **Oct.12: Lecture #9**

An overview of implementation.

Hand Layout and Digitization using a Symbolic Layout Language.

The Caltech Intermediate Form, detailed description of, including tutorial on transformations.

Description of the sorter subsystem (to get people started on HW #5).

### *Homework #5 (Due 17th):*

Construct CIF code for the OM register cell pair; the Sorter Problem .

### *Reading Assignment:*

Ch.4. Hand Layout and Digitization Using a Symbolic Layout Language, The Caltech Intermediate Form.

### *Project Assignment #1 (Due 26th) :*

[The Lab opens this week (CIFtran now ready)].

Practice editing of text files, listing of text files, plotting of CIF-code layout files on the IIP color plotters;  
Keyin and plot CIF code for HW #5; Select course project and write a short project proposal.

## **Week #6:**

### **Oct.17: Lecture #10**

Discuss plan for remainder of course: what will be in the final project assignment schedule to be handed out next week, planned seminar series, the midterm, some rules of the game for projects.

Description of lab facilities, CIF software for editing/plotting of layouts.

Detailed discussion of the sorter subsystem problem.

### *Homework #6 (Due 24th):*

Construct stick diagram for the Sorter Problem.

### **Oct.19: Lecture #11**

More on course schedule/project schedule, rules of the game, area estimates.

Yield statistics.

Delays in Another Form of Logic Circuitry.

Transit Times and Clock Periods.

### *Reading Assignment:*

Ch.1. Delays in another form of Logic Circuitry, Transit Times and Clock Periods.

Ch.2. Yield Statistics.

## **Week #7:**

### **Oct.24: Lecture #12**

Handout the *Guide to LSI Implementation*, discuss contents, suggest readings.

Discuss how class chip set will be implemented.

Describe cell library, particularly the PLA cells and Input/Output pads.

Detailed description of the multiproject chip concept, starting frame, past examples.

### *Handout Project Assignments #2, #3, and #4:*

#2 (Due Nov.9): Detailed project description.

#3 (Due Nov.21): Preliminary layout (a key item for determining chip set space commitments).

#4 (Due Dec.12): Final project reports.

### **Oct.26: Lecture #13**

More on implementation.

Interacting with mask and fab firms.

What happens when the wafers come back: electrical characterization, packaging, functional testing.

The Stored Program Computer, a tutorial on the basics:

Alternative control structures; The stored program computer; Fetch-exccute sequence; Microprogrammed control.

### *Suggested reading:*

During the next several weeks, read the first half of Chapter 5, and Chapter 6.

## **Week #8:**

### **Oct.31: Seminar**

*Interactive Graphic Aids for Integrated System Design.*

Douglas Fairbairn, PARC

### **Nov.2: Midterm Exam**

A two hour examination on the basics covered so far.

## **Week #9:**

### **Nov.7: Lecture #14**

Discuss midterm examination.

Effects of Scaling Down the Dimensions of MOS Circuits and Systems.

Discuss the Scaling of Transit time, Electrical parameters, Current, Current density, Power density.

### **Nov.9: Lecture #15**

Summarize Scaling so far, introduce idea of switching energy and its scaling, scaling of delays to outside world.

Subthreshold conductance phenomenon, and scaling of.

Discussion of limiting factors.

*Project Assignment #2 Due Today.*

## **Week #10:**

### **Nov.14: Lecture #16**

Patterning and Fabrication in the Future.

Scaling of Patterning and Processing Technology, the Runout phenomenon, E-beam and x-ray lithography

The opportunity for remote-entry, fast-turnaround implementation of designs.

### **Nov.16: Lecture #17**

Memory Cells and Subsystems.

Discussion of the relative area, power, delay, and overhead circuitry for a variety of on-chip memory subsystems, including the shift register, the OM2 register cell, the 6-transistor static RAM, the 3-transistor dynamic RAM, and the 1-transistor dynamic RAM.

## **Week #11:**

### **Nov.21: Seminar**

*VLSI Implementation of Speech Processing Functions*

Richard Lyon, PARC

*Project Assignment #3 Due Today.*

Discussion of project status so far. Beginning to allocate space on the chip set. Update on the rules of the game.

Selected preliminary project design files will be transmitted to PARC this week, for individual project plotting and checking at PARC, and to test the overall data transmission scheme.

[Nov.23 is vacation day]

## **Week #12:**

### **Nov.28: Seminar**

*Electrical test patterns for electrical characterization and process parameter extraction.*

Rick Davies, PARC.

*Project update:*

More information about status of space, map of the chip set, library cells, Q and A session to pin down final details.

### **Nov.30: Description of Selected Projects by Students in the class:**

Four projects (from among those already selected for inclusion in the chip set) will be selected for presentation to the class by their designers.

*Project update:*

Discussion of the final status of projects, distribution of plots arriving from PARC, discussion of various known bugs in the various software packages and how to get around them, contingency plans, getting ready for the upcoming design cutoff date, how to sign-off on your design file.

*Suggested readings:*

Selected portions of Chapter 8.

### **Week #13:**

**Dec.5: Seminar**

*Highly Concurrent Systems.*

Carver Mead, Caltech

*Projects: Today is the design cutoff date; all project design files must be at PARC for merging into the MPC tomorrow.*

**Dec.7: Seminar**

*Recursive Machines: A non-von Neumann VLSI Architecture.*

Wayne Wilner, PARC

### **Week #14:**

**Dec.12: Final Meeting of Class**

*Project Assignment #4 (Final Project Report) Due Today.*

Complete Course feedback questionnaire.

Status of the projects: list of those that got on the chip set.

Multi-project chip set progress report: how the implementation is going.

Planned activities during IAP, after the wafers are returned.

Discussion of further reading references.

## **6.978: Introduction to VLSI Systems**

### *Student Projects:*

Much of this course's material, while basic, is new and is not yet part of the academic EE/CS culture. Thus, a strictly lecture oriented course faces tutorial obstacles, as the terminology, artifacts, and abstractions involved will be completely new to most students. A solution is to orient the course around hands-on LSI digital design projects. Actual design experience will enable students to quickly visualize the overall context of the course.

Early in the course, students will define and begin work on LSI projects on the order of useful digital subsystems based on a single cell design, such as a stack, FIFO, etc. Alternatively, the student might design a simple finite state machine controlling a very simple data structure; the traffic light controller in the text is the sort of thing I have in mind here, but with a bit more than just the PLA itself implemented in LSI. By doing these designs top-down, under time pressure, learning the material as they proceed, with homework each week oriented around the projects to keep everyone on schedule, the students will master the material. Students having the listed prerequisites should be able to complete their projects in the available time. Those who have completed what appear to be workable projects by an early December multi-project chip design cutoff date will have the added satisfaction of seeing their projects implemented. All students must complete their projects thru checkplot of full layout by December 12.

In addition to these projects, students having strong software backgrounds might volunteer to also participate in group projects to provide software support for the editing, plotting, and implementation of the LSI design projects. Similarly, students having strong device physics and integrated circuit backgrounds might volunteer to participate in a group project to provide circuit simulation support for the LSI design projects. Other volunteers from the class will be sought for participation in the merging and management of the class project chip set.

## **6.978: Introduction to VLSI Systems**

### *Required Tools and Procedures for Student Project Implementation:*

The completion of digital subsystem projects, through at least layout, will be an essential part of the course experience for most of the students. Work on these projects will proceed at a rapid pace from the beginning of the course thru mid-November, at which time selected completed projects will be merged for implementation. The remaining project layouts may undergo iteration till the end of the course.

For all this to really be feasible requires certain software facilities and some sort of graphics plotter. There must be software, and computing facilities accessible to all students, for the input and editing of symbolic layout descriptions and their subsequent interpretation to generate CIF files. Ideally the layout descriptions would be given in a language such as ICL. Alternatively, the descriptions could be given directly in CIF. There must be a plotter, software for conversion of CIF files to plotter format, and software to interface/support the plotter. A rather simple subset of CIF2.0, supporting boxes at right angles and symbol definitions and calls, would be adequate for the above.

I will provide a complete and documented starting frame targeted for several alternative mask and fab firms. The starting frame will be specified in the simple subset of CIF2.0, and will contain several alternative alignment marks, mask level codes, fiducial marks if required, scribe lines of appropriate profiles, line width testers, test transistors, and, possibly, electrical test patterns for sheet resistivity and line widths. A documented set of input pads with lightning arrestors and output pads with drivers, also in the CIF2.0 subset, will be provided for student use. This starting frame, along with the above facilities, will enable the forming of a multi-project chip set file for transmission to Xerox PARC via the ARPANET. [ The ARPANET will thus be used by M.I.T. and Xerox PARC, both of which are ARPA contractors, to demonstrate the feasibility of remote submission of student LSI projects to a fast turnaround implementation facility ]. This will be scheduled for early December. Final plots of the chip set will then be done at PARC from the CIF file, to verify successful composition and transmission. The CIF to PG conversion will be done at PARC, and software blowbacks generated for final checking of the PG files. The chip set will be implemented by silicon valley mask and fab firms. Wafers will be fabricated and shipped back to M.I.T. by about mid-January. During the latter part of the independent activities program, the wafers will be electrically tested and then diced into chips, and the chips then mounted, wire bonded, and functionally tested.

## **II. The Result:**

**Xerox Palo Alto Research Center**

*3333 Coyote Hill Road*

*Palo Alto, CA 94304*

*March 31, 1979*

To the students, staff, and faculty  
participants in the fall semester '78  
VLSI system design course at M.I.T.

Dear Friends,

The color photos of the 1978 M.I.T. multi-project chip set have arrived! I'm sending one out to each of you. Enclosed with each photograph is a detailed list of the class projects, and a map to help you locate specific projects in the photograph.

Quite a few people have had a chance to test their projects. So far, the results are really very good, especially when considering the limited tools we used and the time pressure all of us worked under: I have reports that projects 3, 4, and 5 work completely correctly; project 6 has been partially tested, and works correctly so far; the important subsections of project 7 work correctly; project 12 works completely correctly except for a couple of inverted outputs; project 17 has some wiring bugs, but may be partially testable, and I hear rumors that a bigger and better version is planned! I'd appreciate hearing about any additional functional test results as they become available.

If in the future any of you undertake research or development activities related to integrated systems, I'd be very interested in staying informed of your work. Also, if you ever travel through the San Francisco Bay Area, and have some time, plan to stop by and visit at PARC. Good luck to you all.

Sincerely,

Lynn Conway

## M.I.T. 1978 Multi-project Chip Set:

*List of projects (see attached map and color copy of photo):*

Filed on <Conway>/maplist.memo

1. *Sandra Azoury, N. Lynn Bowen, Jorge Rubenstein*: Charge flow transistors (moisture sensors) integrated into digital subsystem for testing.
2. *Andy Boughton, J. Dean Brock, Randy Bryant, Clement Leung*: Serial data manipulator subsystem for searching and sorting data base operations.
3. *Jim Cherry*: Graphics memory subsystem for mirroring/rotating image data.
4. *Mike Coln*: Switched capacitor, serial quantizing D/A converter.
5. *Steve Frank*: Writeable PLA project, based on the 3-transistor ram cell.
6. *Jim Frankel*: Data path portion of a bit-slice microprocessor.
7. *Nelson Goldikener, Scott Westbrook*: Electrical test patterns for chip set.
8. *Tak Hiratsuka*: Subsystem for data base operations.
9. *Siu Ho Lam*: Autocorrelator subsystem.
10. *Dave Levitt*: Synchronously timed FIFO.
11. *Craig Olson*: Bus interface for 7-segment display data.
12. *Dave Otten*: Bus interfaceable real time clock/calendar.
13. *Ernesto Perea*: 4-Bit slice microprogram sequencer.
14. *Gerald Roylance*: LRU virtual memory paging subsystem.
15. *Dave Shaver*: Multi-function smart memory.
16. *Alan Snyder*: Associative memory.
17. *Guy Steele*: LISP microprocessor (LISP expression evaluator and associated memory manager; operates directly on LISP expressions stored in memory).
18. *Richard Stern*: Finite impulse response digital filter.
19. *Runchan Yang*: Armstrong type bubble sorting memory.

*The following projects were completed, but not quite in time for inclusion in the project set:*

20. *Sandra Azoury, N. Lynn Bowen, Jorge Rubenstein*: In addition to project 1 above, this team completed a CRT controller project.
21. *Martin Fraeman*: Programmable interval clock.
22. *Bob Baldwin*: LCS net nametable project.
23. *Moshe Bain*: Programmable word generator.
24. *Rae McLellan*: Chaos net address matcher.
25. *Robert Reynolds*: Digital Subsystem to be used with project 4.

## **M.I.T. 1978 Multi-project Chip Set:**

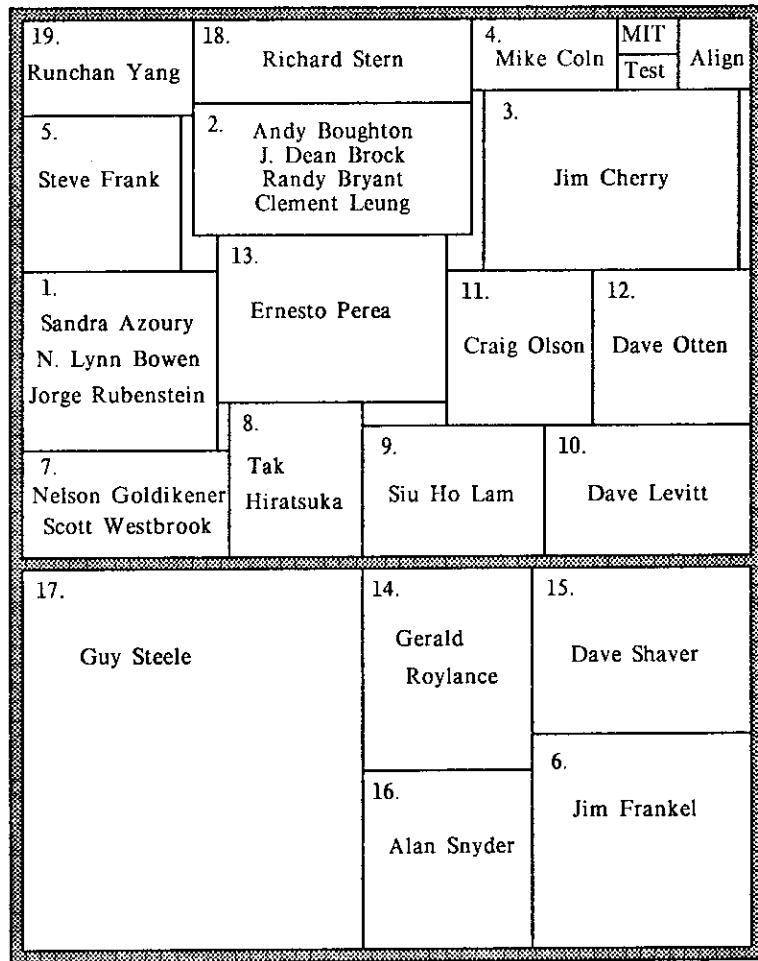
*Chip Set Implementation Sequence:*

Filed on <Conway>impsched.memo

*Information Management:* Xerox PARC;

*Masks:* Micro Mask; *Wafer Fabrication:* HP Deer Creek; *Packaging:* M.I.T.

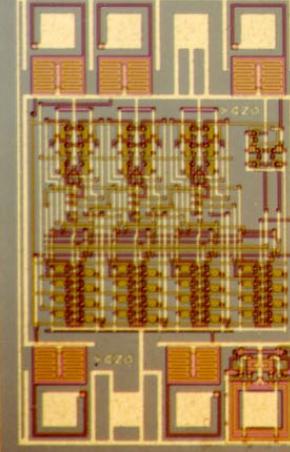
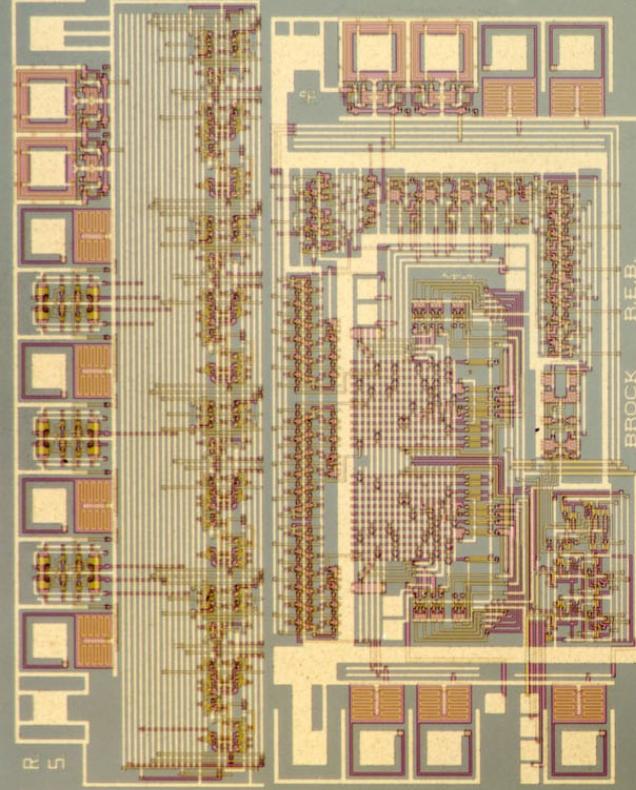
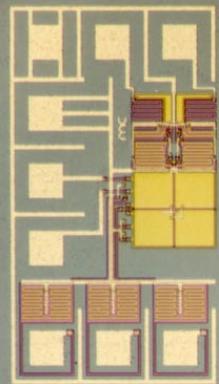
- A. Individual project design files, encoded in CIF2.0, were transmitted to Xerox PARC via the ARPANET during 5 and 6 December 1978.
- B. Files were converted from CIF2.0 to ICARUS format, and merged at PARC into a starting frame using ICARUS, based on a tentative space allocation pasted up at M.I.T. The overall design file was then converted to Mann PG format.
- C. Maskmaking began at Micro Mask, Inc. ~12 December, using their electron beam maskmaking system.
- D. Maskmaking at Micro Mask was pipelined during the next several weeks with wafer fabrication at Hewlett-Packard's Deer Creek Research Lab.
- E. Wafers (total of 41) were fabricated by 8 January 1979.
- F. Initial electrical testing and packaging were done in M.I.T.'s Materials Science Lab. Packaged chips, custom wire-bonded to individual projects, were available to students for functional testing by 18 January 1979.



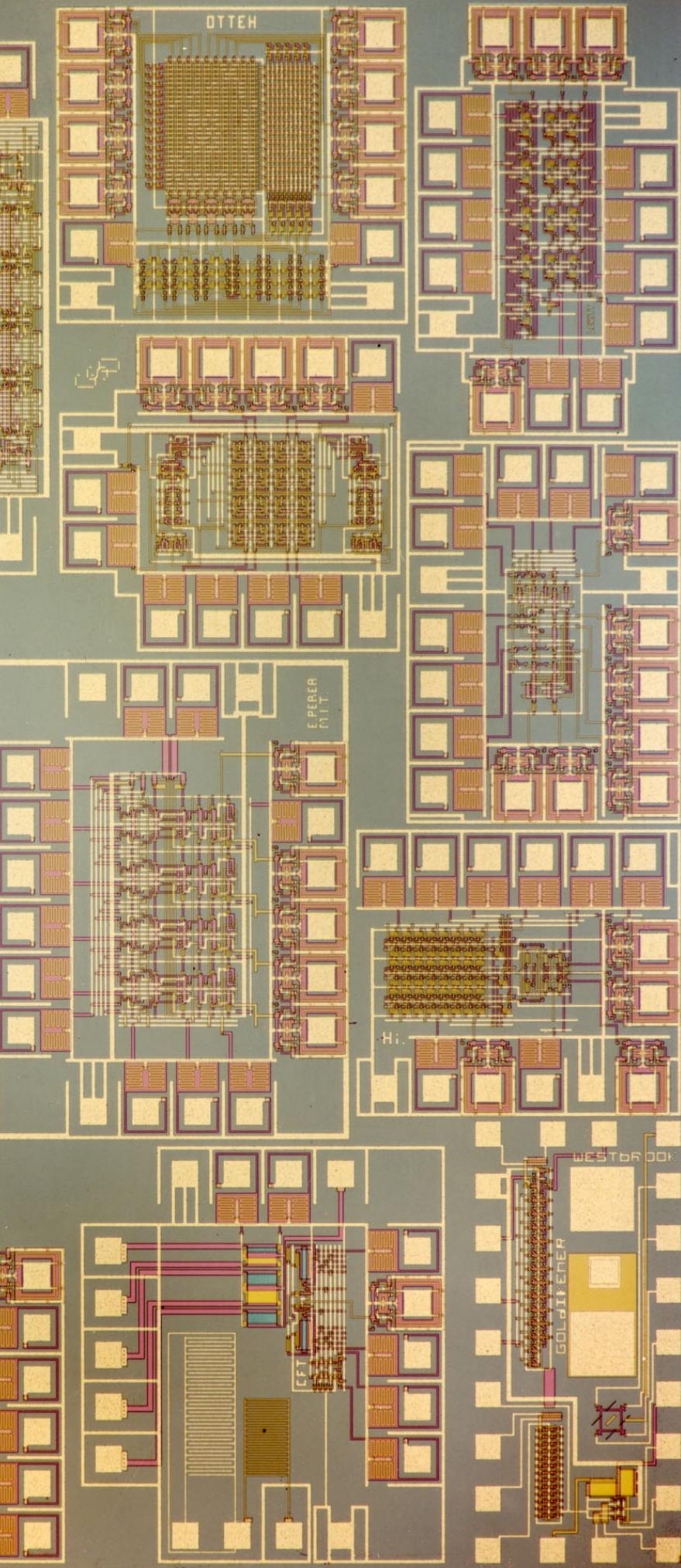
**Map of the M.I.T. '78 Multi-project Chip Set**

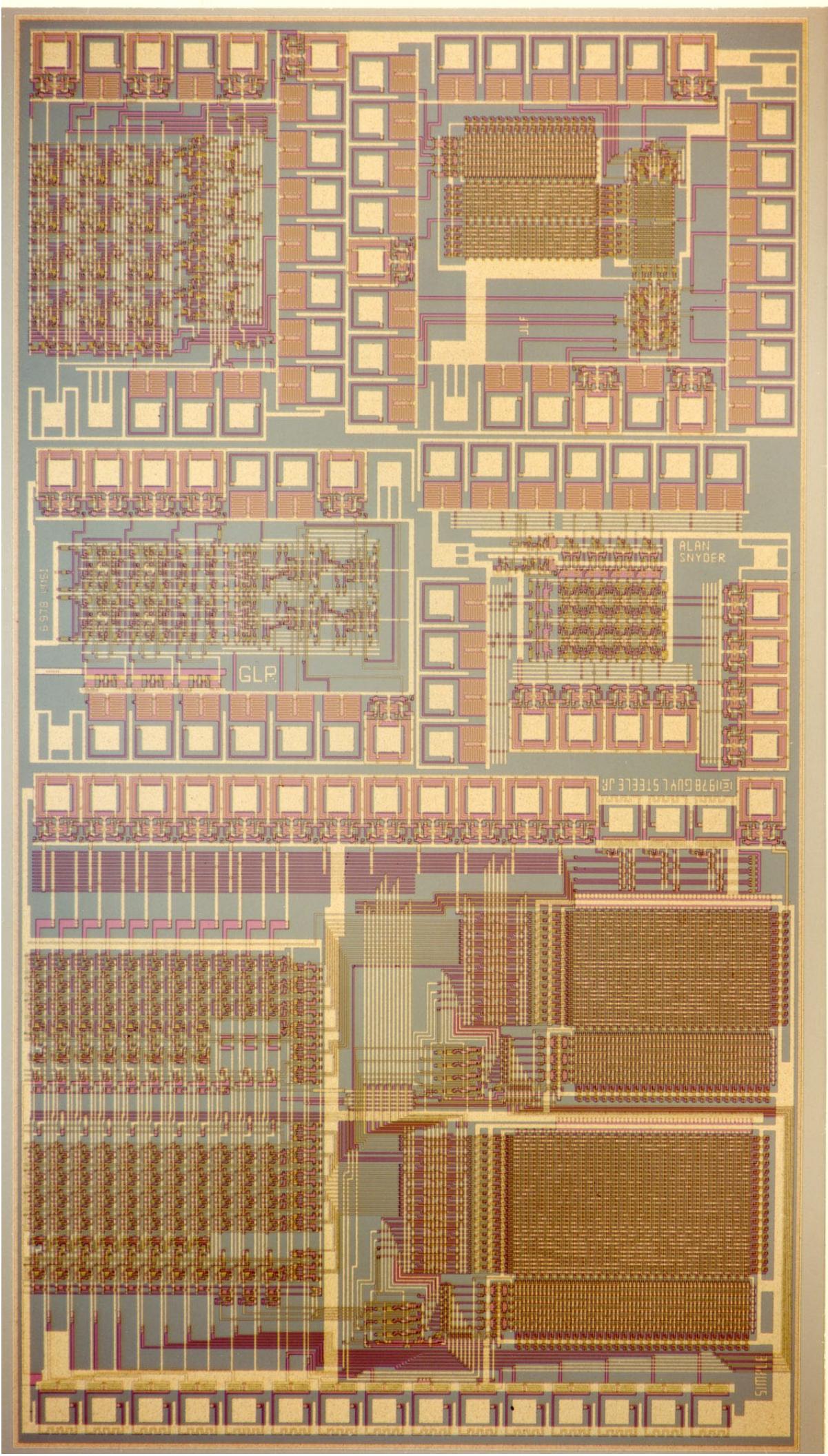
DIF +  
POL +  
CUT +  
MET +

M.I.T.  
'78



B.R.C.





### **III. Lecture Notes**

## 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.

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

Cover Today:

### > 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.
- > 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. Dont 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 MATH & OCCAS. SPOT QUIZZES
- > ASSIGNED READINGS: Periodically I'll indicate sections in the text you should read & study. These 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 THURSDAY, 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 <sup>MUCH</sup> 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 breakdowns):

$$\text{SPOT QUIZES+HW (10x10)} + \text{MT} + \text{FINAL} + \text{PRELIM PROJ} + \text{PROJ FINAL REPORT.} \\ (50) + 100 + 100 + 100 + 100 = 500(+)$$

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

> VLSI : Why "Very---"? Improvements in implementation technologies have resulted in suff. circuit density so that  $\sim$  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  $\times 100$  or more can still be and will be made.

**BLACKBOARD**

Individual paths  $\sim$  4 to 6  $\mu$  wide, separated by 4 to 6  $\mu$ . A micron is  $1/1000$  mm. Chgs are  $\sim$  2 to 6 mm on a side. In future, lines can be reduced to less than 0.5  $\mu$ .

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

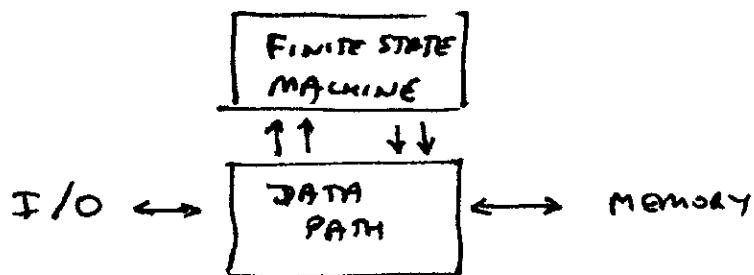
The Design of structures to be implemented on a chip is now an architect's game. This trend will increase in the future.

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

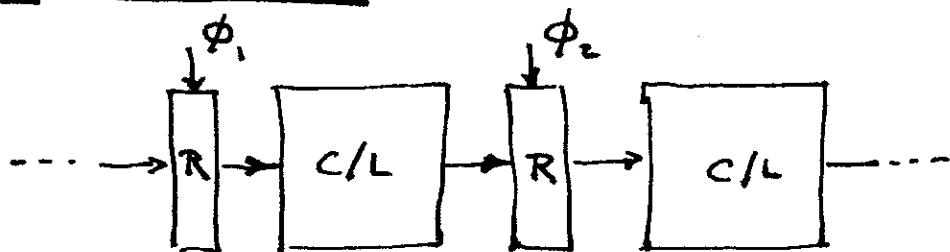
All such are composed of FINITE STATE MACHINES  
CONTROLLING REG-REG DATA TRANSFER PATHS

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

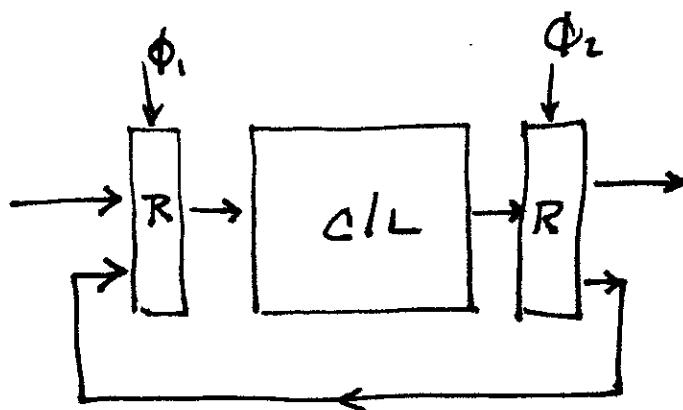
Sketch } EXAMPLE: STORED PROGRAM COMPUTER:  
 on Board }



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 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 Sub-system to logic to circuit to stick (topology) design 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

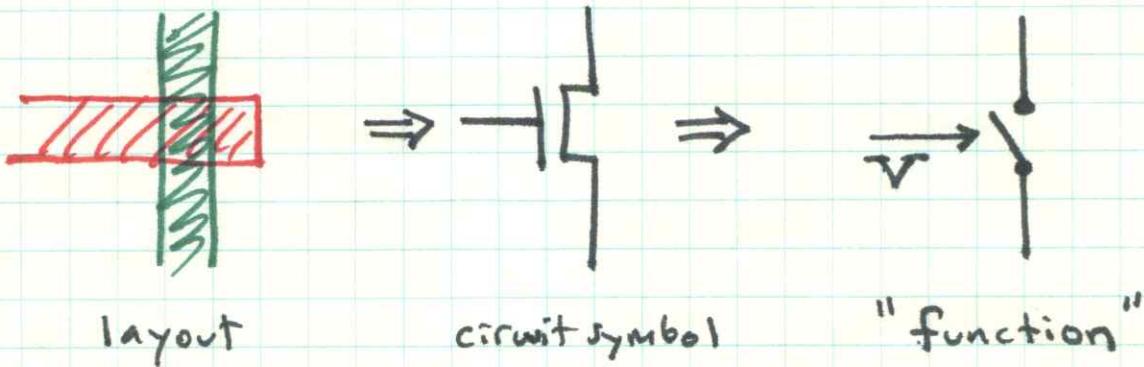
- > I'd like to develop the basic ideas of how systems are integrated in nMOS technology.

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**

3 levels  
of  
conducting  
material  
on  
insulating  
material  
board

By photolithography we can pattern the conducting material to 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:



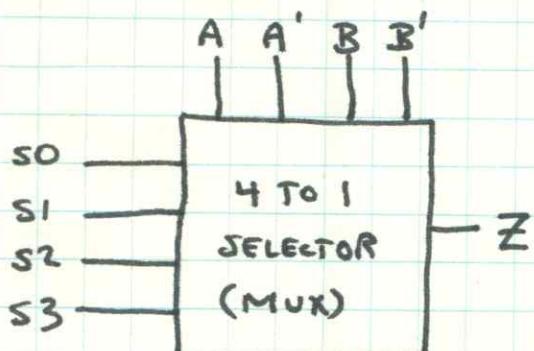
- > WHAT COULD WE DO WITH THESE? Let's look at some simple examples of digital functions we can very easily implement using simple groups of switches, wired together in particular ways.

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

## EX NMOS C/L FUNCTIONS: TOPOLOGY

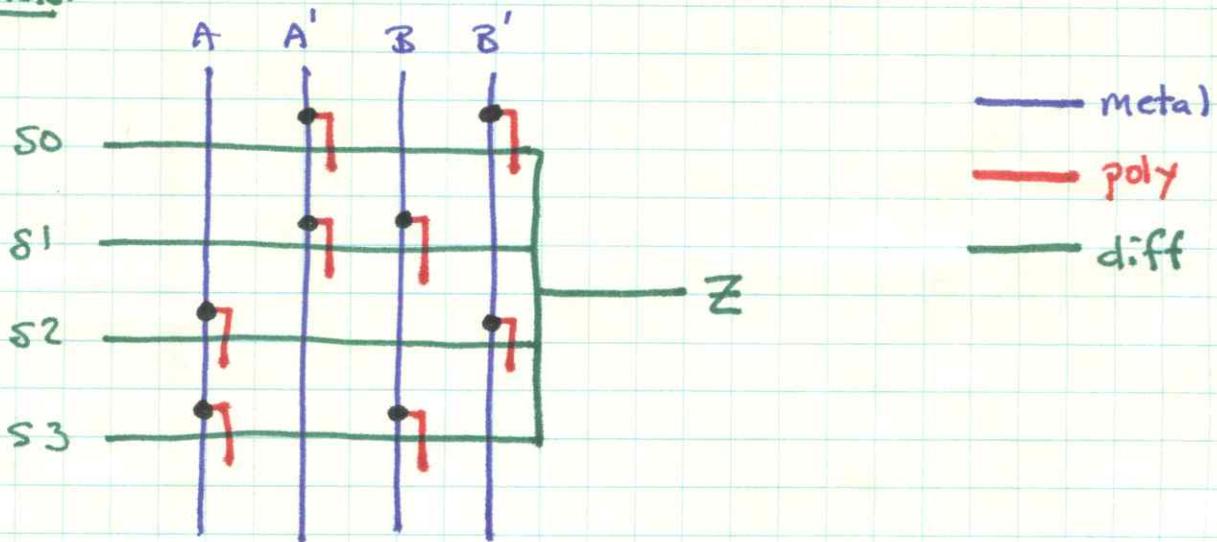
NMOS 2

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

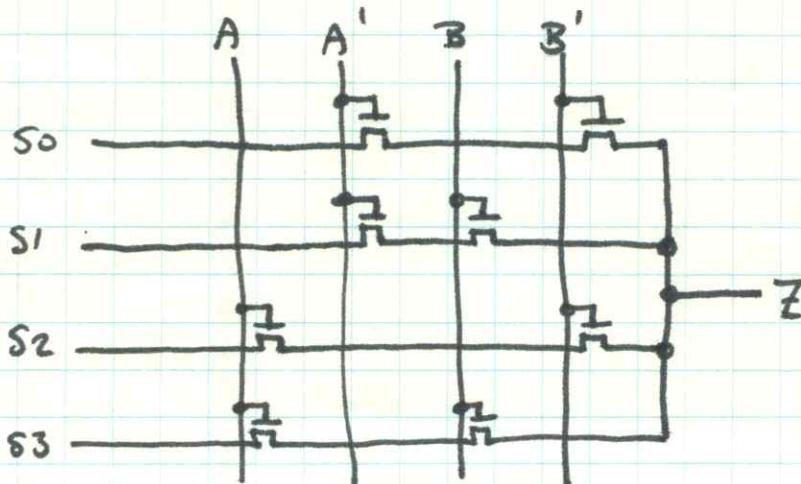


A	B	Z
0	0	S0
0	1	S1
1	0	S2
1	1	S3

SOLUTION:



CORRECT. TRANSISTOR DIAGRAM:  
CIRCUIT

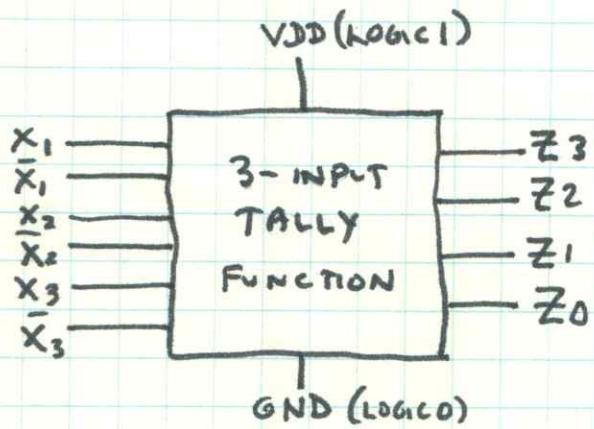


> Interestingly, we will often find it easier & more optimal to design in this way (reminiscent of early relay switching logic) rather than use the formal methods of design synthesis using logic gates directly switching

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

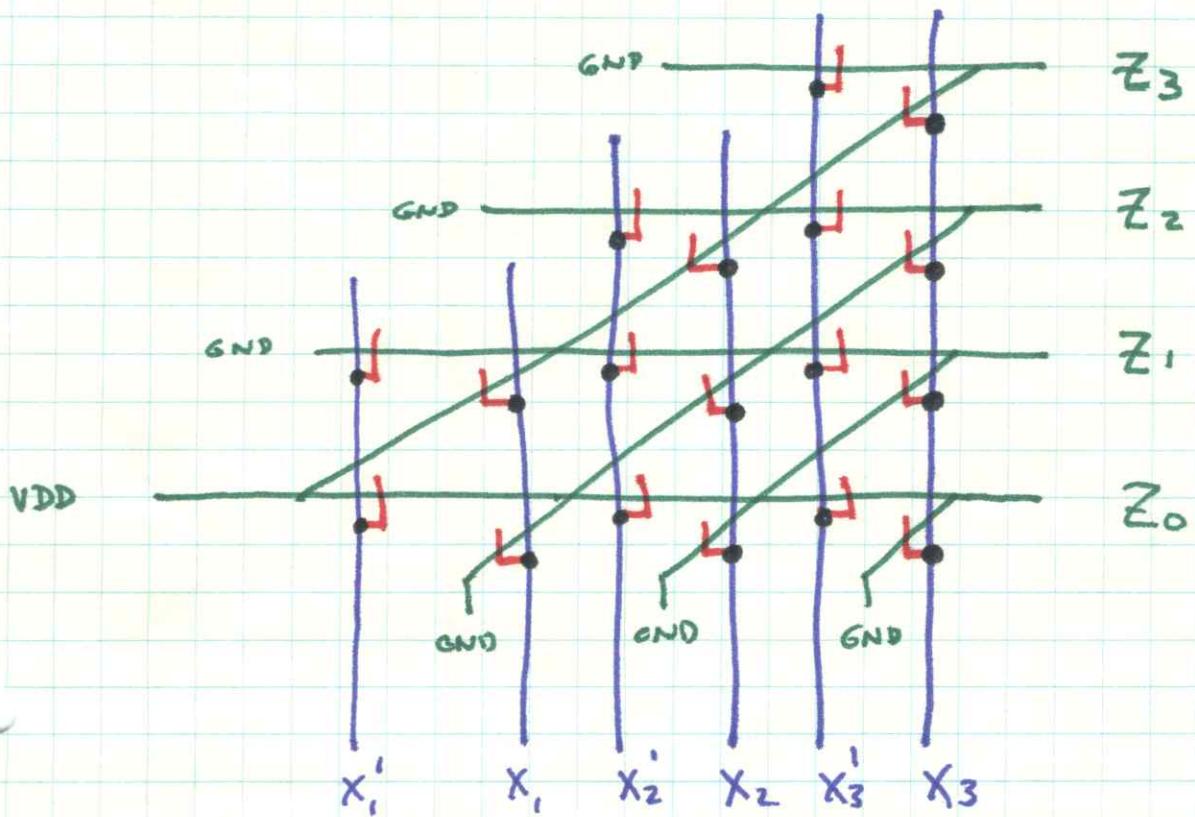
N INPUT VARIABLES, N+1 OUTPUTS.

IF M INPUT ARE EQ 1, OUTPUT # M = 1



X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1
1	0	0	0	1	0	0
1	0	1	0	0	0	1
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 propagation 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 geometries of our layouts, i.e. on how narrow line-widths can be and 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" COMPUTER 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 CARLEN MCAD 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 MAKE 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 CHI-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", AT 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).\*

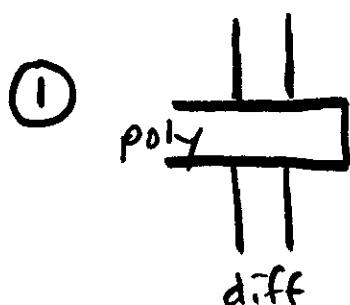
> Mention again: Some problems \* will require design, even invention. In such cases only a small fraction of students may get an answer ---

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

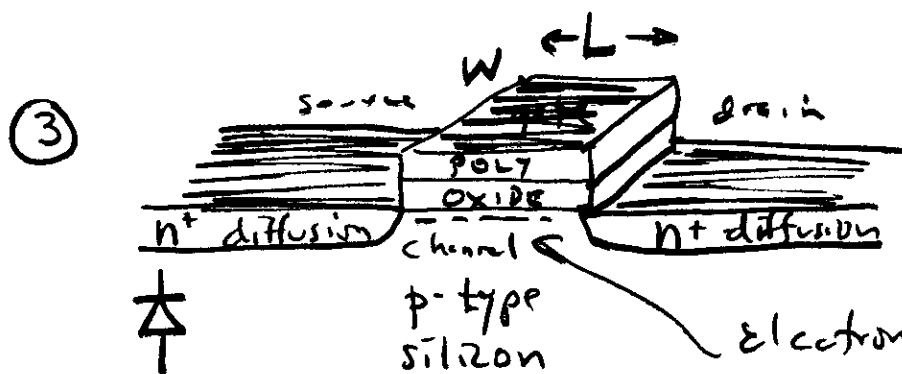
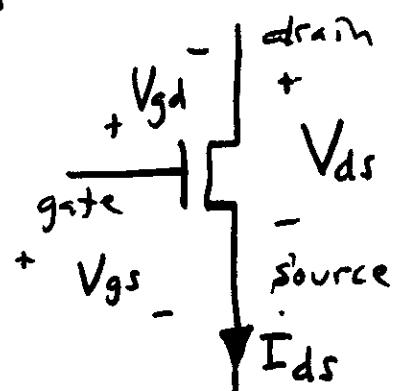
FOUR TOPICS: AND THESE ARE ALSO ASSIGNED READING IN TEXT.

- > The MOS Transistor
- > The Basic Inverter
- > Inverter Delay
- > Basic NAND/NOR Log.2 Gates

### MOS TRANSISTOR: Put up View Graph



② NAMES/SYMBOLS



Electrons attracted if  $V_{gs} = V_{th}$

Enhancement Mode if  $V_m > 0$

- $I_{ds} = Q / \tau$
- $\tau = L / \bar{v}_l = L / \mu E$

- for small  $V_{ds}$ ,  $E = \frac{V_{ds}}{L}$

$$\boxed{\tau = \frac{L^2}{\mu V_{ds}}}$$

- Transit time is the fundamental time unit of our integrated system. All times scaled to  $\tau$  of smallest FET's in system.

- $Neg Q \text{ in transit} = -C_g(V_{gs} - V_m) = -\frac{\epsilon W L}{D}(V_{gs} - V_m)$

- $I_{ds} = -I_{sd} = \frac{\mu \epsilon W}{LD}(V_{gs} - V_m)V_{ds}$

- For given  $(V_{gs} - V_m)$ ,  $I_{ds} \propto V_{ds} \Rightarrow -\frac{R}{V_{ds}}$

i.e.  $\rightarrow \boxed{I_d = \frac{V_{gs}}{R} + V_{ds}}$   $\left. \begin{array}{l} \text{FOR small } V_{ds} \\ \& V_{gs} - V_m > 0 \end{array} \right\} \begin{array}{l} \text{Resistive} \\ \text{Region} \end{array}$

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

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

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

$$\rightarrow I_d = V_{gs} \frac{1}{R} + V_{ds}$$

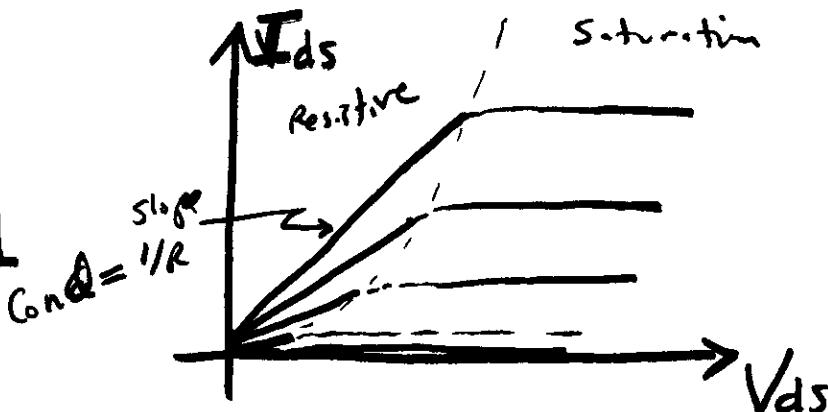
$I_{ds} \approx \frac{\mu \epsilon W}{2LD} (V_{gs} - V_m)^2$

In Saturation,  
FET behaves  
like Current  
Source controlled  
by  $V_{gs} - V_m$

{ Trans. time is

$T = \frac{L^2}{\mu (V_{gs} - V_m)}$

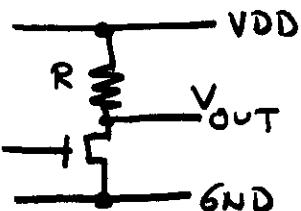
Draw & Discuss  
Characteristics  
Summarizing  
These Regions



## THE BASIC INVERTER: PUT UP SLIDE 2

- FCN: OUTPUT TO BE COMPLEMENT OF INPUT
- WHEN DESC LOG FCN: Logic-1 = Voltages  $\geq$  Defined logic threshold  
Logic-0 = " "  $<$  This logic threshold

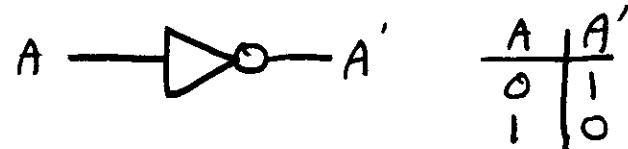
- If could make resistors, could build inv.:  
explain function  
But, R's especially at large value, long wire.



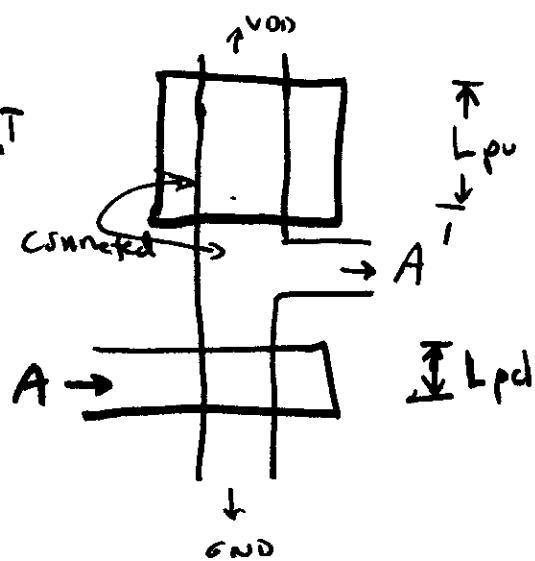
- So, we use a depletion mode MOSFET in place of a resistor

- The depletion mode MOSFET has a threshold voltage  $V_{dep}$  that is less than zero. During Fab ---

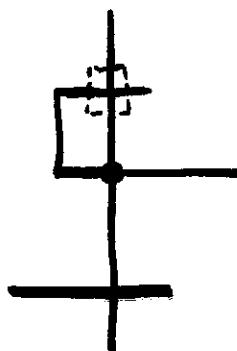
- 



- LAYOUT



- STICK



$\rightarrow / \leftarrow$   
W

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

INPUT VOLTAGES  $\geq$  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 typical pair of MOS Transistors used to impl. inverters

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

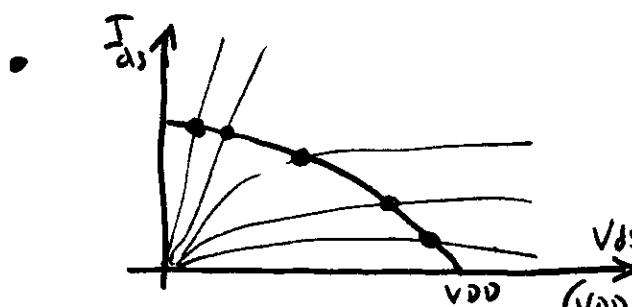
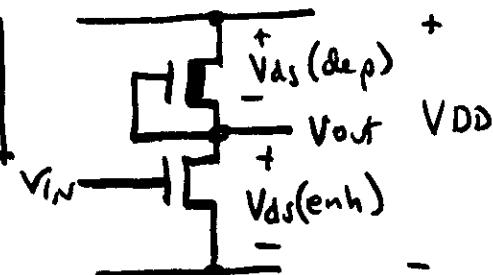
### CALCULATE INV. TRANSF. CURVE: i.e. $V_{out}$ vs $V_{in}$

- 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

- $V_{ds(\text{enh})} = V_{DD} - V_{ds(\text{dep})}$

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

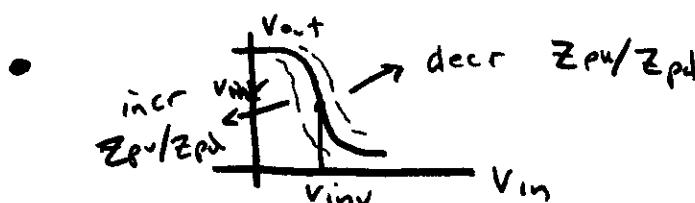


{ only one curve for D-MOS FET is relevant:  
that for  $V_{gs} = 0$

Superimpose  
 $I_{ds(\text{enh})}$  vs  $V_{ds(\text{enh})}$  (VS)  
 $I_{ds(\text{dep})}$  vs  $(V_{DD} - V_{ds(\text{dep})})$

Since currents equal, intersection of curves yields

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



|slope| = Gain

## INVERTER LOGIC THRESHOLD VOLTAGE

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

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

ITS VALUE DEPENDS ON RATIO OF

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

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

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

$$\boxed{\text{Typically } V_{th} \cong 0.2VDD}$$

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

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

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

$$\boxed{V_{INV} = VDD/2}$$

- .2 & -.8 choices

$$\boxed{V_{INV} \approx \frac{VDD}{\sqrt{\frac{L_{pu}/W_{pu}}{L_{pd}/W_{pd}}}}}$$

- ~~(Assume  $L_{pu} = L_{pd}$ )~~ This leads to a pull-up/pulldown ratio of

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

## 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  $\sim \tau'$ , pulldown of 1st remove  $Q \approx VDD C_g$  from gate of second inverter.
- Pullup of second must supply this charge to  $C_g$  of third, but it can supply only about  $1/k$  times the current of pulldown of 1st.
- SO speck of inverter pair delay (one lowgoing + highgoing transition)

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

- If one inv drive more than one succeeding inverter, for example  $f$  of them,

(identical)

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

- These simplified notions of delay are based on a "switching" 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

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

so  $Z_{pu}$  must be  
bigger  
( $L_{pu}$  longer)

- 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's 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:

This Week We Move Up a Level :- Discuss Inverter Delays.

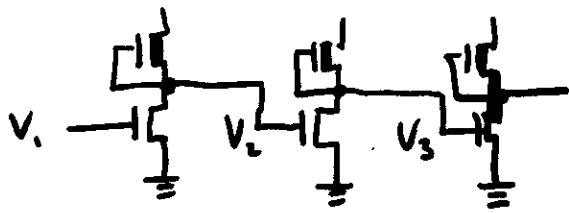
- We'll learn how to make Registers.
  - We'll study an example subsystem : A Stack
  - We'll learn how to impl. irreg C/L in a regular way using PLA's
  - We'll learn how to implement Finite State Machines using registers & PLA's.

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]

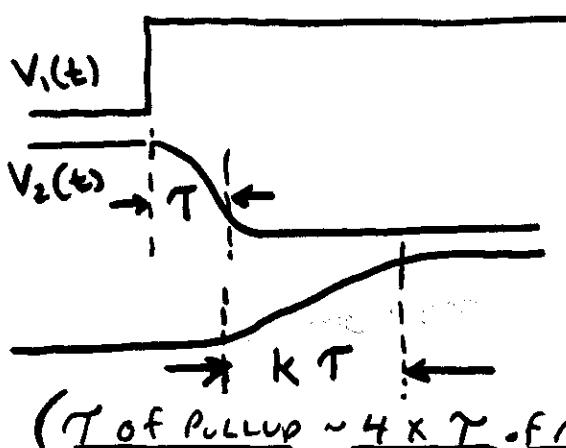
- Resistive Region:  $T = L^2 / \mu V_{ds}$
- Saturation Region:  $T = L^2 / \mu(V_{gs} - V_{th})$  {larger  $V_{ds}$  doesn't reduce  $T$ }
- Examine case where an inverter drives successive similar inverters:



Suppose:  $V_1 = 0$ . Then at  $t=0$ ,  
Drive  $V_2 \rightarrow V_{DD}$ .

What happens?

### Graph of Effect:



- It takes  $\sim T$  (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 (turn pull-up of 2nd) to  $V_{DD}$ .  
 $(T_{of\ pullup} \sim 4 \times T_{of\ pulldown})$
- But pullup has less current capacity than pulldown, so this charging up takes longer, by  $\sim$  ratio  $k = \frac{I_{PULLUP}}{I_{PULLDOWN}}$ .
- So normally speak of inverter p.r. delay =  $(1+k)T$
- If Fanout  $f$  (i.e. 1st feed of 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  $6 \times 6 \mu\text{m}$

- Resistances

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

Transistors:  $10^4 \Omega/\square$

NOTE: Res FET's >> Res wires

- Capacitances: (to substrate)

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

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

Note:  $C_g$  only  $\times 10$  that of wires. But wires typically  $\times 10$  area of gate they feed. So typically must multiply  $\times 2$  the gate capacitance to estimate delays. (Call this parasitic capacitance)

- Calculate  $T$  TWO WAYS: (Ballpark, to get order of magnitude)

$$> T \approx L^2 / \mu \left( \frac{V_{DD}}{2} \right) \quad \text{Now good, } \mu = 800 \text{ cm}^2/\text{volt}\cdot\text{sec}$$

$$T \approx (6 \times 10^{-4})^2 / 800 \left( \frac{5}{2} \right) \approx .38 \times 10^{-9} = 0.38 \text{ ns.}$$

So actual inverter  $T$  (incl.  $\times 2$  for parasitics) =  $0.36 \text{ ns}$

$$> T \approx R_{FET} C_g \times 2_{\text{parasitics}}$$

$$T \approx 10^4 \times 4 \times 10^{-4} \times 10^{-12} \times 2 \times 36_{(\mu\text{m}^2)}$$

$$T \approx 288 \times 10^{-12} \approx \boxed{0.29 \text{ ns}}$$

- Above are actually what we would measure for  $T$  in RING oscillators for the best current  $6 \mu\text{m}$  processes. Over many processes, typically  $0.3 < T < 1.0 \text{ ns}$

NOTATION: You've read about Notation. In particular, 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.

»» sometimes easier to visualize for w.r.t. a particular type of mixed notation

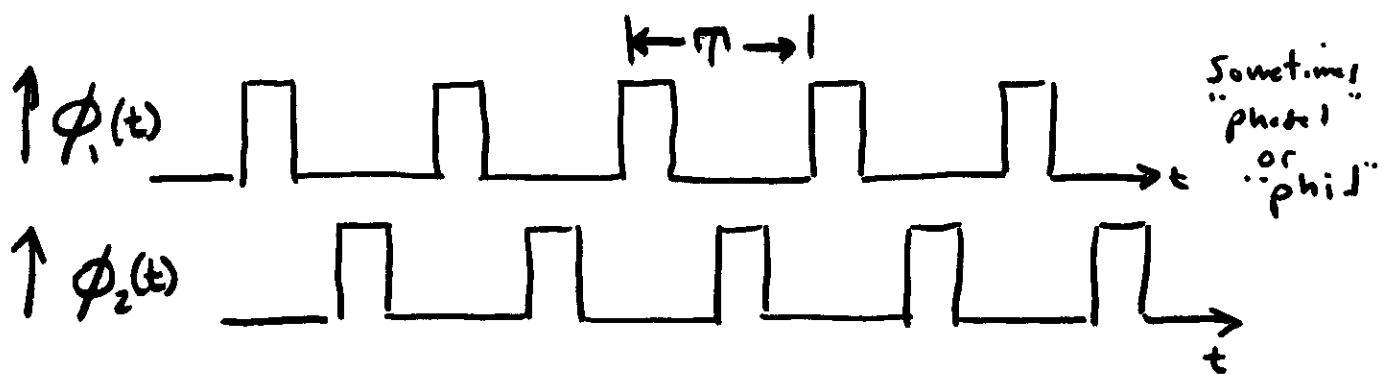
### TWO PHASE CLOCKS:

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  $f(t)$ :



> The signals switch between  $\sim 0$  volts and  $\sim VDD$ .

> Both have the same period  $T$ .

> The high times of both are shorter than their low times

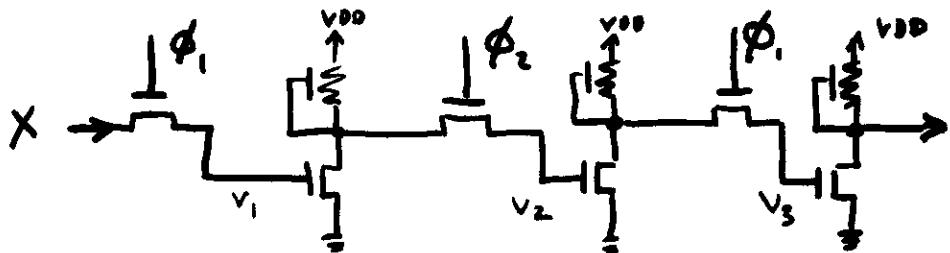
> They are never both high at the same time, i.e.

- [why? The lock master must never need to be open] they never overlap.

- TIMING / SYNCHRONIZATION are in the general case subtle, complex. We'll come back to this later. For the sort of system --- sync'd. the Z-Q is perfectly O.K.

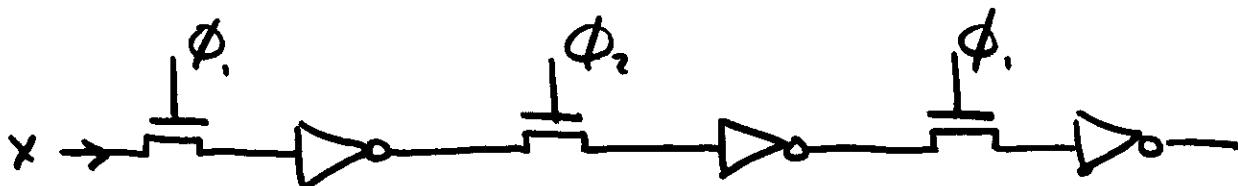
## THE SHIFT REGISTER:

- 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:

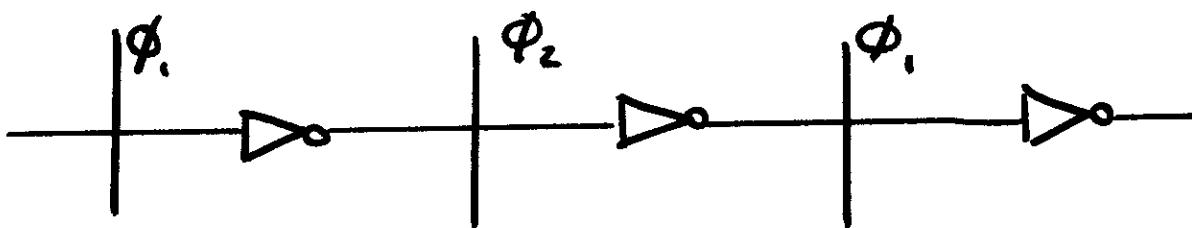


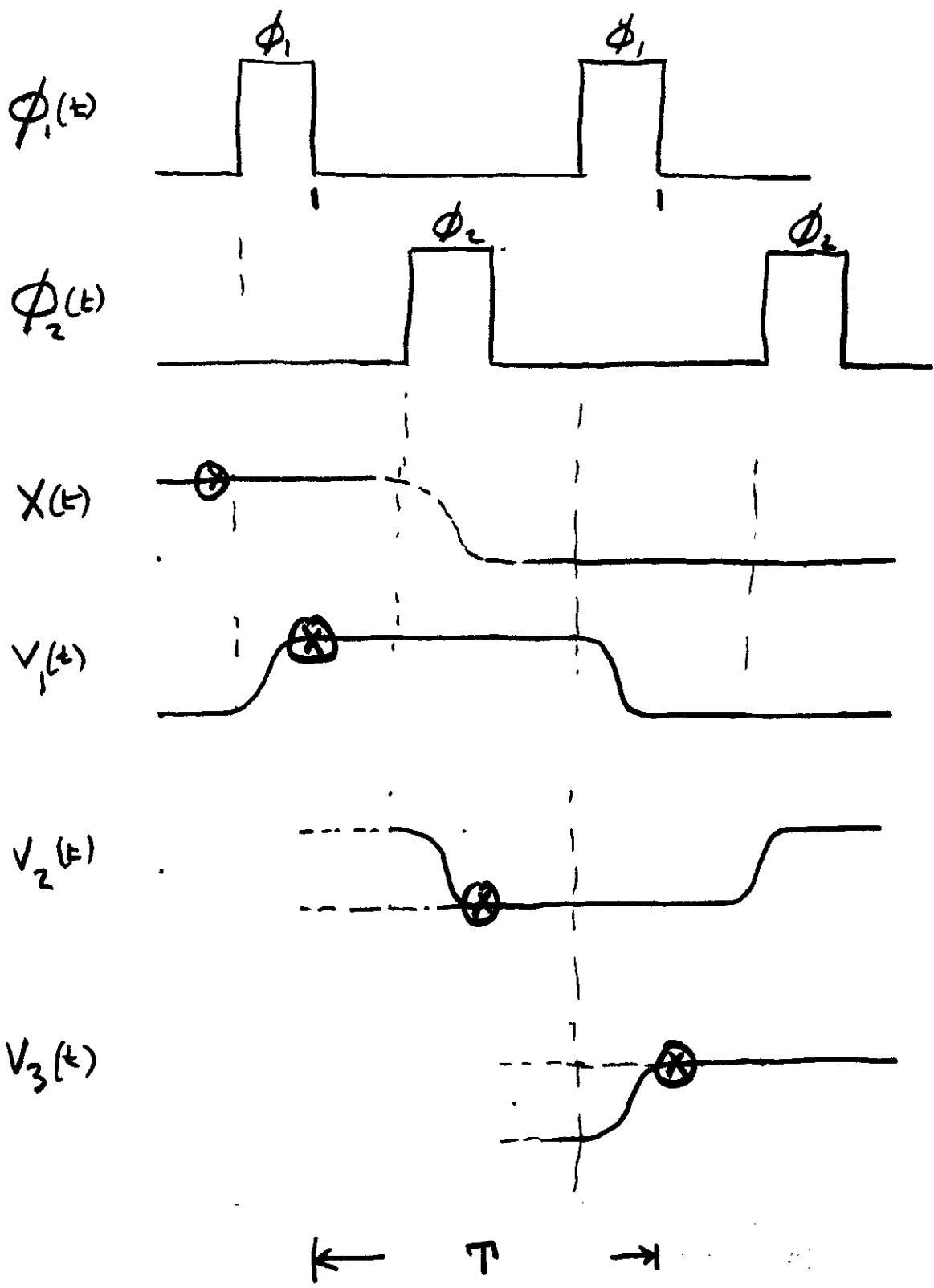
maybe graph  
 $V_1, V_2, V_3$   
 vs  $\Phi_1, \Phi_2$

- Describe movement of data during  $\Phi_1$ , followed by  $\Phi_2$   
 (mention term "pass transistor" or "transmission gate".)  
 as those transistor "switches" not part of pull-up/pull-down 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  $S_g$ , the gate of a FET.
- Now start to show Part of stick diagram:



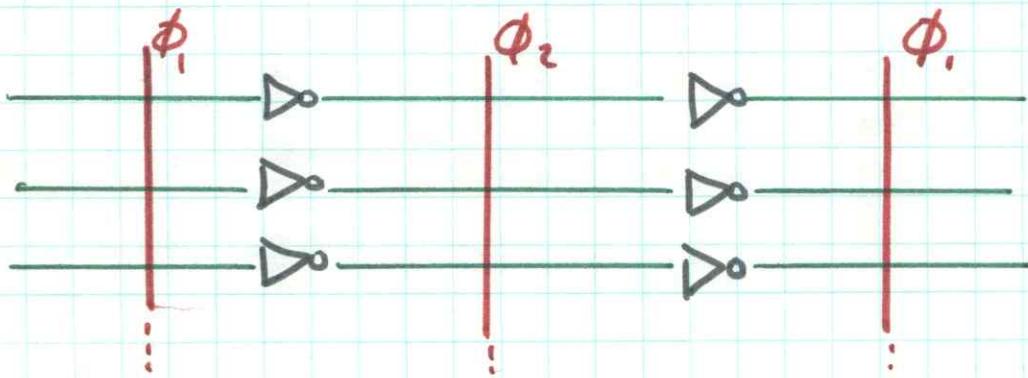


$$V_1 \rightarrow V_3 \quad \text{in} \quad \tau = " \phi_1 " + " \phi_2 "$$

## Continue to Build on Shift Register Idea:

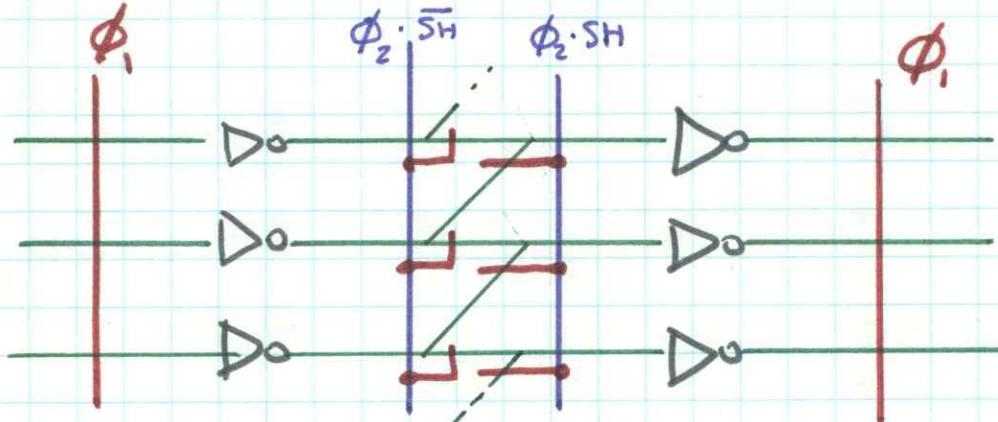
How could we move a sequence of words from register stage to register stage - rather than just a sequence of bits?

- 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

---

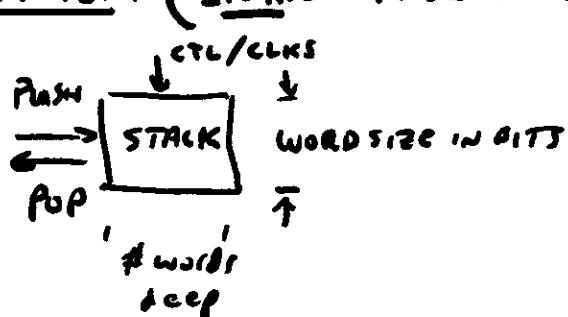
- HOW TO IMPLE SIMPLE REGISTERS:

SHOW & DISCUSS FIG 7-9 SLIDE

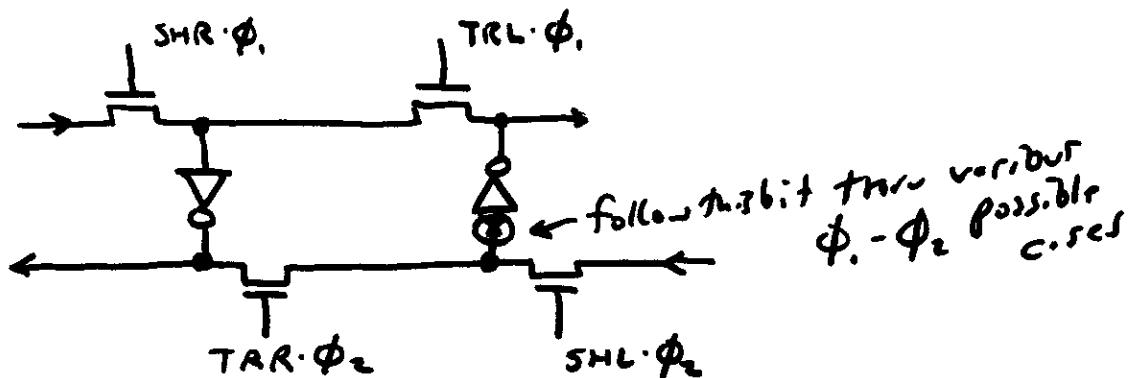
---

- DESIGN OF A STACK SUBSYSTEM (INTRO: TO BE CONT. NEXT TIME)

- 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.

(WE'LL 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 state diagram

## Lecture #4: Thurs 21 Sept:

- Before we begin today's meeting: 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 ~ midafternoons 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.

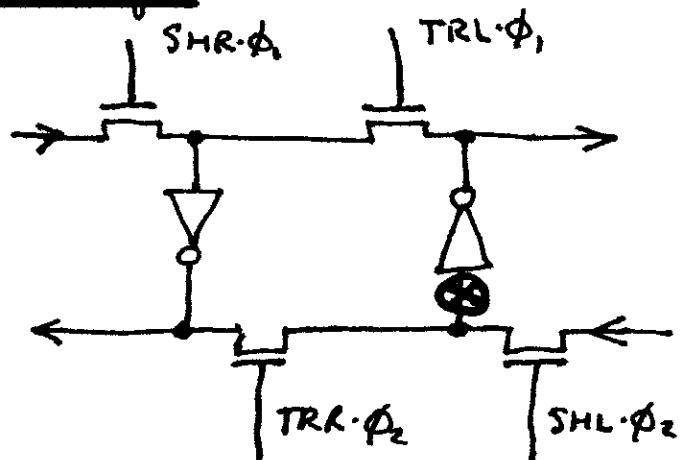
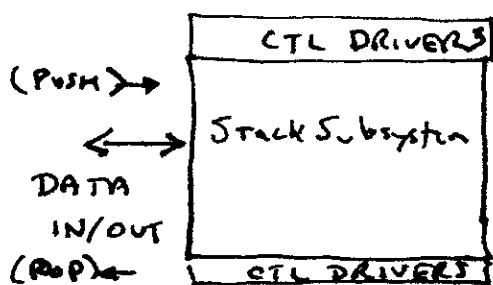
> SEMINAR SERIES: In addition to Carver Mead lecturing, there are a number of individuals currently actively research in the area of integrated systems. Some of these people will become very well known as time passes.

Mention: Carlo Sequin, Bob Sprout, Chuck Seitz, Doug Fairbairn, Dick Lyon, Wayne Wilbur, Rick Davies, H.T. Kung, Bob Horn, etc.

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

If  
have  
time

## Continue with Stack Example:



- Note data moves HOP, control lines run vertically.
- Walk thru again: If on  $\phi_1$ ,  $\notin$  on  $\phi_2$ , then Fcn:

TRL	,	TRL	$\rightarrow$	NOP
SHR	,	TRR	$\rightarrow$	PUSH
TRL	,	SHL	$\rightarrow$	POP

### SLIDE: TIMING DIAGRAM:

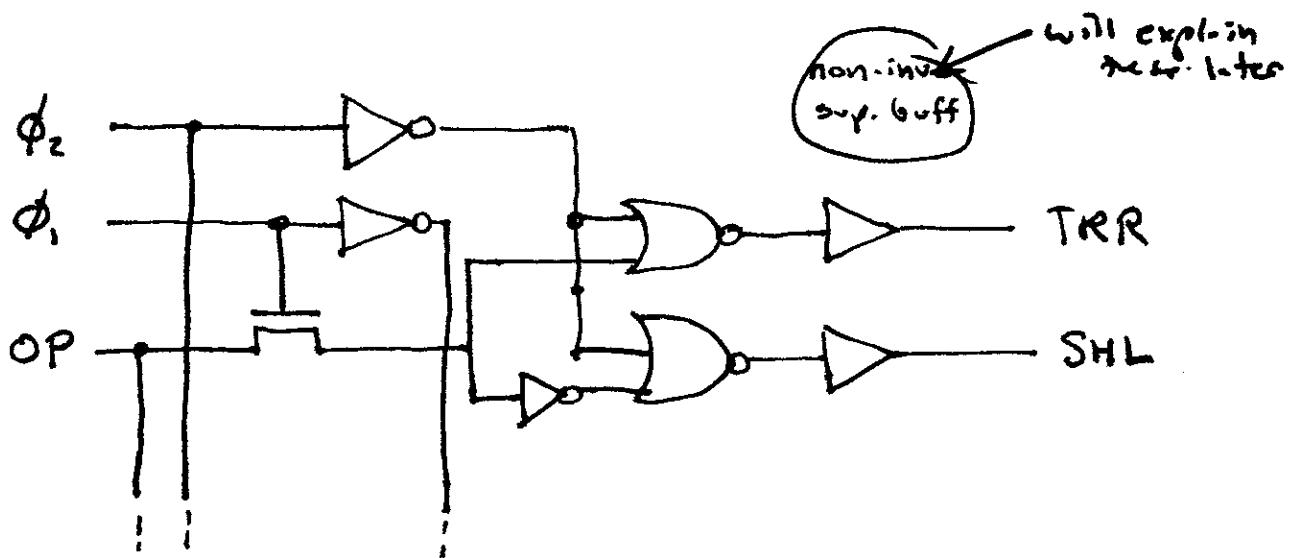
- >  $\phi_1, \phi_2$  always running.
- > if NOP most of time, then see that TRL, TRR, occur most of time.
- > We need only one control signal (call it "OP") to cause (if on at  $\phi_2$ ) the TRR to be followed by TRL and SHR, TRR [PUSH]
- > OR to cause (if on at  $\phi_1$ ) the TRL to be followed by TRR and SHL [POP]

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

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

From OP,  $\phi_1$ ,  $\phi_2$  to input to drivers which operate the control lines running across the stack:

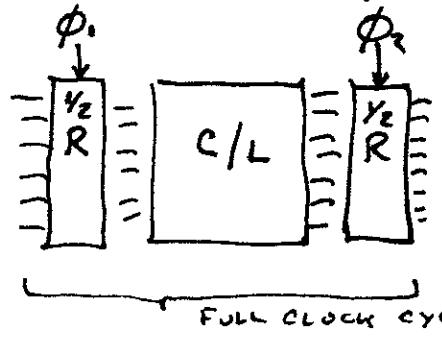
Here's a possible design:



- When  $\phi_1$  is High, if OP high then drive SHL.  
if OP low then drive TRR.
- When  $\phi_2$  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  $\phi_1$  goes off, whichever NOR gate output is high stays high till  $\phi_2$  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 of with same one line. This can overlap  $\phi_1$ ,  $\phi_2$  ops. Tricky idea to reduce # macro-code lines.

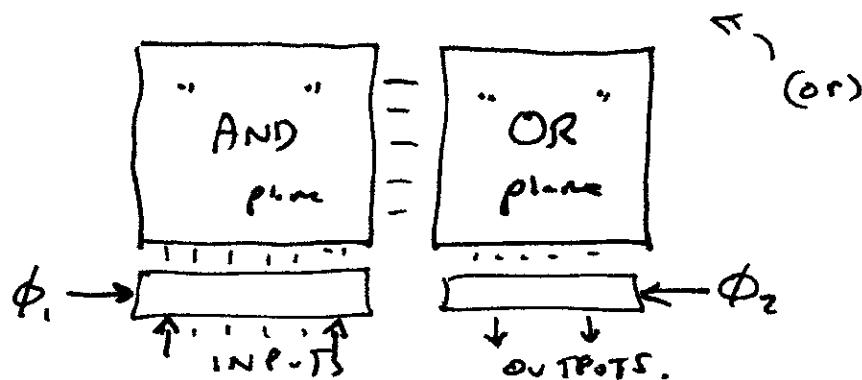
Now, Can make REGs, Some forms of C/L. If could only implement arbitrarily irregular C/L in some regular way, we'd have all we need:

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



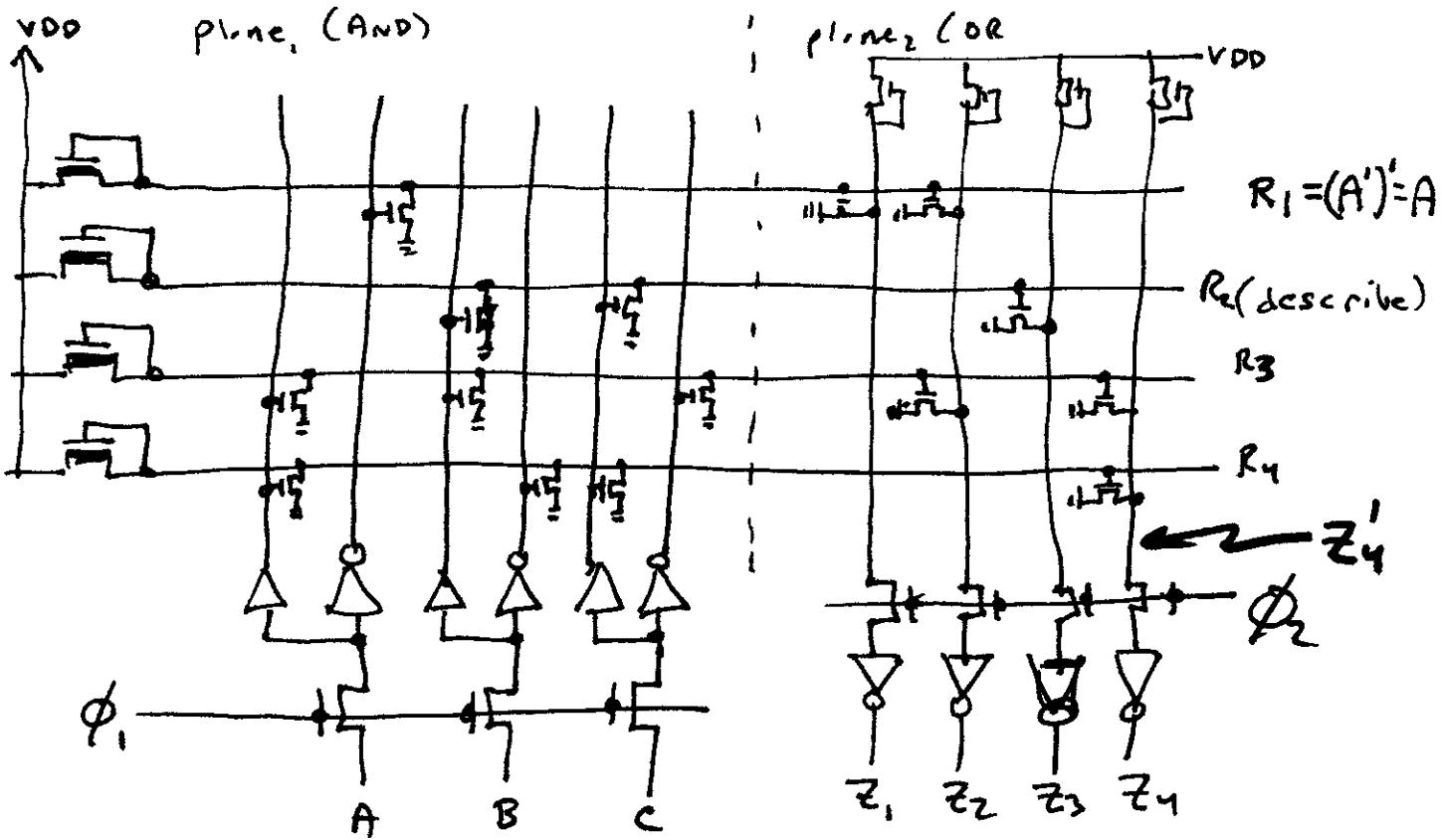
COULD USE A MEMORY,  
BUT This would REQ  
ALL  $2^n$  poss comb. of  
inputs  $\times$  # bits in output.  
Often wasteful. (mention)  
PROMS

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 = 1  
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  $\phi_1$ ,  $\phi_2$  registers to indicate how it is imbedded in system. Also to anticipate the Finite State Machine.



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

$$\begin{aligned} \text{i.e. } (R_{000}) & R_1 = (A')' = A \\ & R_2 = (B+C)' = B'C' \\ & R_3 = (A+B+C)' = A'B'C' \\ & R_4 = (A+B'+C)' = A'B'C' \end{aligned}$$

OR Play: If Row B high it pulls Vert in low, so output high  
(if Transistor)

Notice how the OR plane now forms the "OR" of input rows

$$\underline{\text{Ex:}} \quad Z_4' = NOR(R_3 R_4) = (A'B'C + A'B'C')'$$

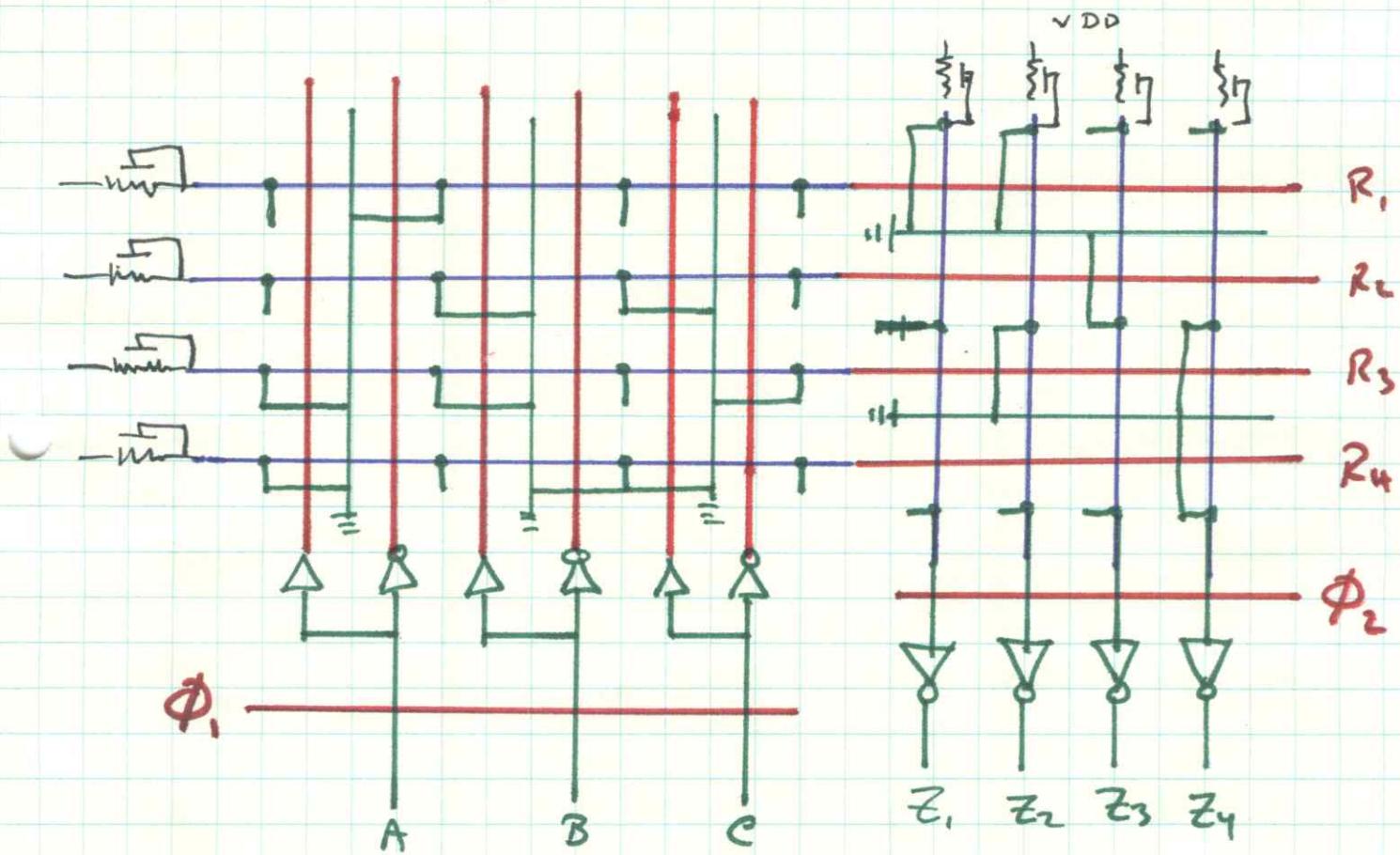
$$\text{Thus: } Z_4 = A'B'C + A'BC'$$

[FLIP OVER SHEET/ USE WHITEBOARD]

6

## STICK DIAGRAM THE PLA EXAMPLE

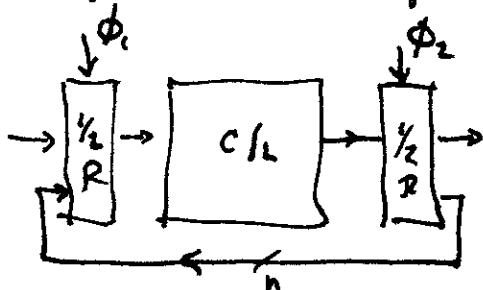
- Run control lines in POLY
- Pullup to output lines in Metal
- Run Ground paths in Diff between alternate poly lines
- Both planes the same, just tilt over an AND plane to get an OR plane.
- Put in input regis/drivers. Pullups.
- Program with transistors at appropriate places.



- 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 ( $\lambda$ )
- Since use NOR form, delays are not too bad.

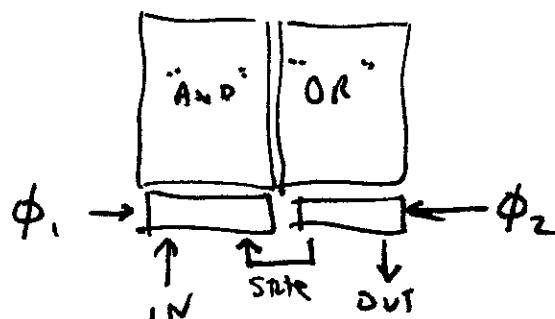
## FINITE STATE MACHINES

- In many cases in the processing of data, it is necessary to know the outcome of the current proc. step, before proceeding 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 stage having  $n$  segments:



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 ( $n$  encoded) by feedback paths ( $\# \text{states} = 2^n$ ). 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

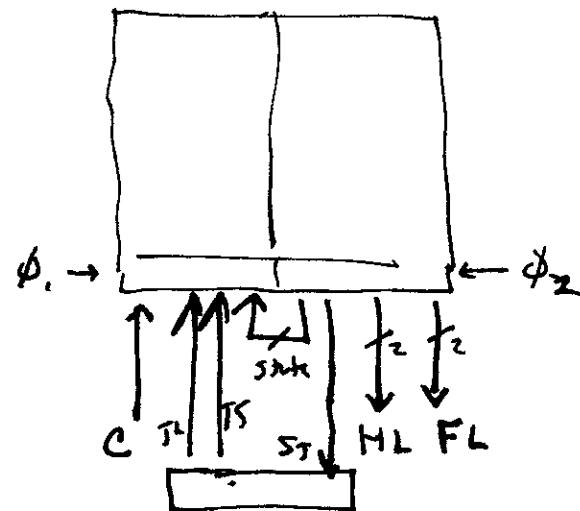
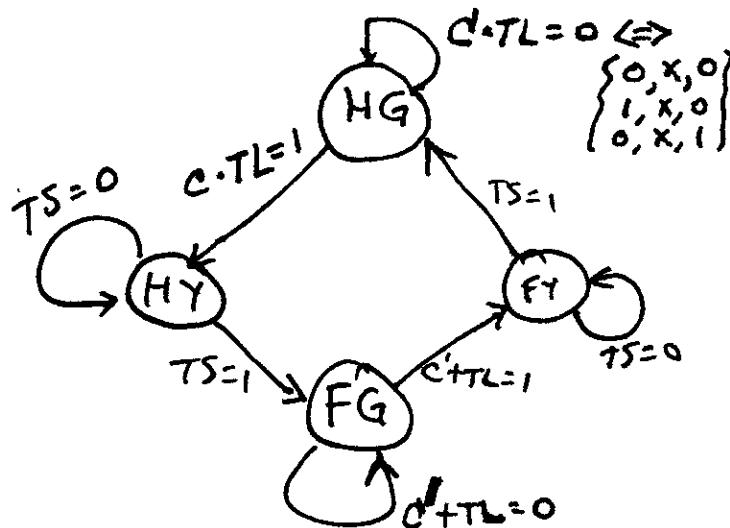


Show  
Part 2 Design a Traffic light controller:

Let's go thru a complete example: **SLIDE**,

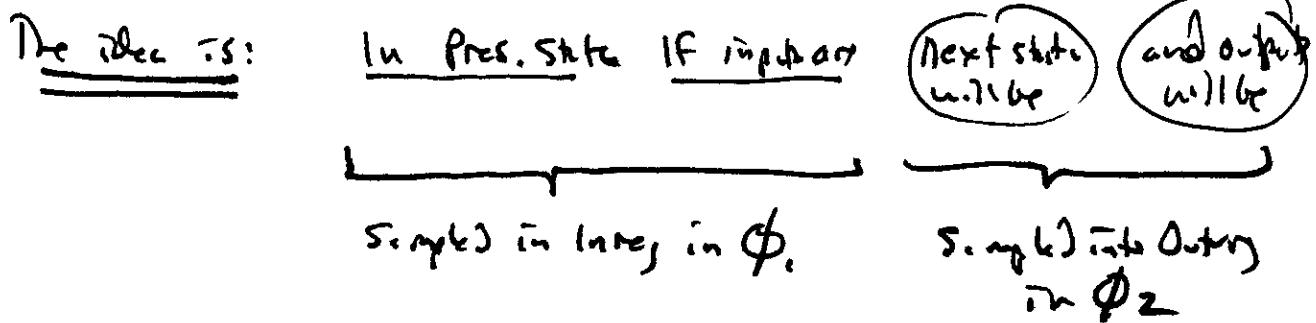
- Busy highway intersected by a seldom used farmroad.
- Detectors installed which cause a signal to go high when cars are at either position C.
- If no cars are at positions C, we wish to control the traffic light so that it remains green.
- If cars are present, want highway lights to cycle thru caution 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 then not interruptible 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 possible 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.



- 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 and convince yourself that the stick diagram impl. the required function.

# Do Before Traffic Light Controller

10

- 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  $\sim \frac{1}{125}$  of a chip. It can run at a clock rate  $\sim 10^7$  times as fast as the real-time problem, By late 80's, it might occupy  $\sim \frac{1}{25000}$  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 STK diagrams.

LECT: SEP 26

LECTURE #5 (TUES)

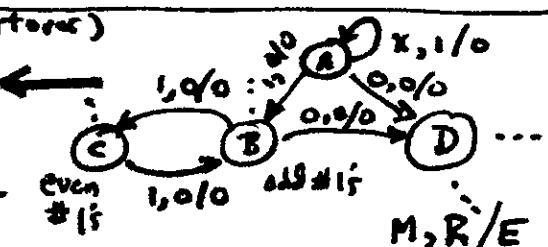
A (startover)

Admin

• Handout Homework #3 ←

• Handout Questionnaire

• Collect Homework #2 even #15



• Announce: New Room, St. next time: 39-400

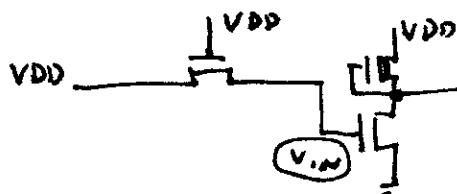
• Explain Questionnaire: Lab Sched / Pass. Seminars.

### Discuss HW#1:

- Most did very well. If have questions, see me.
- Show best solution to #2(a) - J. D. Brooks, several others had similar solutions. **(ON BOARD)**

Interesting thing: Area goes linearly with # inputs.  
Solution is shown before,  $\leftrightarrow$  those that most got,  
went as  $N^2$ .

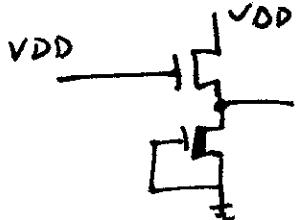
- Anticipate  
next week's  
lectures*
- Show 2 versions of Selector. Indicate what's to come in delays in pass transistors:  $\propto n^2$ .  
So, not too many of these: ~~many many many~~ but add delay.  
They simplify/compact layout,
  - So, there are real limits to how far we can carry the “arrays of pass transistors” idea. Well get more into this next week.
  - Note: Already hinted that if couple inv. logic with pass transistors, then must use 8:1 pullup/pulldown std. why?



*We'll calculate this more next week*

VDD is highest voltage. If VDD on gate of pass-FET, then V<sub>in</sub> can at most be VDD - V<sub>th</sub>. So, for V<sub>out</sub> to go close to VDD, V<sub>in</sub> = VDD - V<sub>th</sub>, as worst normal inverter with V<sub>in</sub> = VDD, must keep higher voltage

- bring this up now because: Some people use dep. mole pull-downs & enh. mole pull-ups.  
This was o.k. given info you had. But be careful:



Suppose want a non-inverting level restoring stage: If  $\text{in}_+ = \text{VDD}$ ,  $\text{o}_\text{out}$  at most  $= \text{VDD} - \text{V}_\text{th}$ .

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 stuck dig. running.

3(C): More switches. Some might be very nice., some went on to find PCB construction

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 Slides  
Archit. - Des - Symbols - Layout - Des layout then  
MAKE THE MASTER PATTERNS FOR EACH LAYER (MASKMAKING)  
then transfer these into each layer at approp. process step (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 geometric rules for generating layouts before we can have any to describe.

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

- PATTERNING: There will be 5, or 6, (or 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 classic case is the patterning of  $\text{SiO}_2$ . This is done several times during transistors process.

## Talk Thru 2 Slides. May be use Board.

- > Oxide grown (expose bare wafer to  $\text{O}_2$  in furnace) on silicon wafer.
- > Coated with a photoresist material: a partially organic compound.
- > 2 kinds of resist: Positive (<sup>uv</sup> 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 through gaps in mask, it enters the resist,  $\text{SiO}_2$ , Si. No effect on  $\text{SiO}_2$ , Si. But, breaks down resist.
- > Develop it in organic solvent which rapidly dissolves broken down resist (but only very slowly. Fat all attacks other).
- > Now we use a selective etchant in this case HF, which dissolves  $\text{SiO}_2$  but not resist and not Si.
- > Use stronger organic solvent to remove resist.
- > Have Transferred the Opaque Mask pattern into  $\text{SiO}_2$  Positive

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

- Now: Let us 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 original pattern is transferred into the resist.]
  - > [Note missing ore mask. We'll come back to that later when we look at a sequence covering a more complex structure which contains Digi-Metal pull-ups]
  - > Emphasize that this is going on everywhere across wafer at same time. The process steps are Pattern Independent. It's like Developing Film.

- Now: Let's 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 d-T diffusion (Fig 12)
- > Contact cuts only go just so deep.
- > Metal over cut to poly next to d-TT: Buttin Contact more later
- > Mention Sixth mask: Oxide less & run out to Pads
- Mention that most books/articles concerning the process show these profiles. We usually don't need to see these. Bh: and there aren't 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.

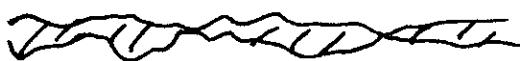
- What are major problems in the process? FOR GIVEN PROCESS:

>> There is some standard deviation in line widths

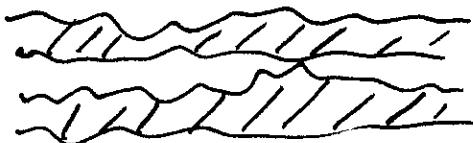
$$\begin{array}{c} \text{Wavy lines} \\ = \\ \text{Average width} \end{array} \quad \sigma = \sqrt{(x - \bar{x})^2}$$

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

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



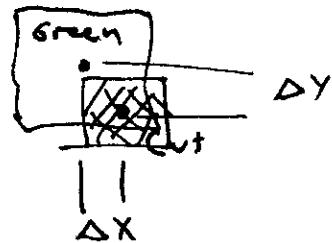
> If put them too close together, they'll sometimes touch!



>> 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 empirical value: if you try to make lines smaller/closer, you'll get into trouble.
- We define the half-width of the minimum lines as the length unit  $\lambda$ .  
for the particular process
- Think of  $\lambda$  as some moderate # multiples of the standard deviations  $\sigma_w$  and  $\sigma_r$ .
- Think of  $\lambda$  as the "resolution of the process"

$$\frac{\text{minimum width}}{\text{lines}} \stackrel{+}{=} \lambda$$

[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  $\lambda$ ]

- 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 larger objects.
- However: 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.
- For a product, 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     $\frac{\text{min diff line}}{\text{" poly line}} = 2\lambda$   
Fig 23     $\frac{\text{" poly line}}{\text{min diff line}} = 2\lambda$

> Min Sep is same: min poly sep (Fig 18) =  $2\lambda$

> But We usually use min diff sep (Fig 16) =  $3\lambda$   
 since if b/w voltage of N+ve, if too close,  
 then depletion regions may overlap and  
 current flow between them.

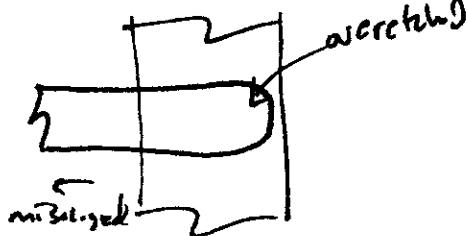
Note: Some processes use a special step to eliminate  
 this problem.

- > Now: what about two level separations:

Fig 19: Must keep at least  $\lambda$  bet. poly & diff (uncrossed)  
 or will get an unwanted large capacitance  
 when red overdrives green

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

Must overlap by a least  $2\lambda$ . Reason, if ever  
 there is not an overlap, will get a short circuit



An EXAMPLE OF SPECIAL RULE:  $\begin{cases} \text{If Poly width} > n\lambda, \\ \text{then overlap can be} \geq 1.5\lambda \end{cases}$

## Continue with Slide 1

- Fig 21 shows examples: overlap Poly 2λ, Poly-diff 2λ, diff must be 2λ wide (triangular)
- To Form Depl. mask FETs: Yellow region must extend 1.5λ beyond gate; i must be  $1.5\lambda^+$  from any enhancement mode gates. Fig 22
- Contacts: See Figures 23, 24, 28.
  - > Cuts must be min. linewidth size: min =  $2\lambda \times 2\lambda$  square
  - > Must be surrounded by at least one on all sides, to insure contact, and no contact with unwanted layers
  - > Cuts to diffusion must be  $2\lambda$  from nearest gate to positively insure no shortout of FET
  - > usually use many small cuts to contact 1-2μm of diff, which decreases contact resistance.
- METAL: Due to Steep slopes, rough terrain, Metal Should be  $3\lambda$  wide  $3\lambda$  long (except contacts  $0.4\cdot 1\lambda$ )
- BUTTING CONTACT: If want to contact Poly to Green, use Butting contact (described)
- 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 (fixed film line) can make direct poly-diff contact. Rates are very fabrication dependent.  
Advantage: much smaller contact area. Can run metal over top. Makes some layouts much easier.
- In principle can contact to Poly over Gate: But, can't do this to minimum size gate especially if less than  $2\lambda$  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: <sup>(MORE RULES)</sup> 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 ~2½ 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 position), > thin oxide, > then metal. [Gate not self-aligned]

- Mobility of Holes is only  $\approx 1/2$  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 constraint design methodology, you can see why LSI at first appeared uninteresting to system architects - the attitude was that such circuit work should be left to how chips are designed.  
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 unstructuredly designed logic when there are huge variable capacitances laying 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 I port, when in fact you're The "Steel" Analogy: working a report card into .001... <sup>copying algorithm</sup> fully directly in silicon structures.)

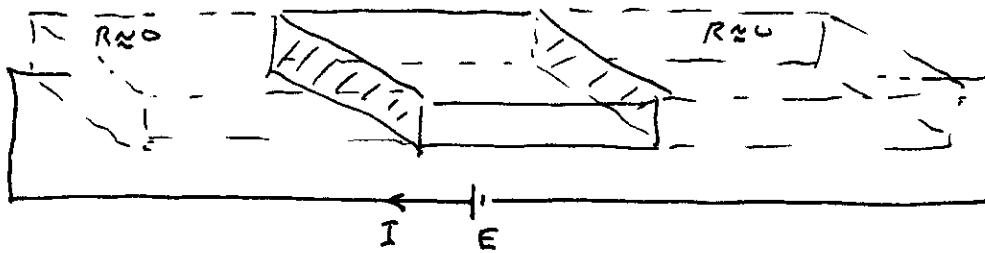
### LAYOUT:

#### > 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  $R \sim 100\Omega/\text{D}$ .

Poly ok for clock lines, and we'll see other long run uses later. but not OK 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 D between really good conductors. Measure  $R = \frac{E}{I}$

So have:

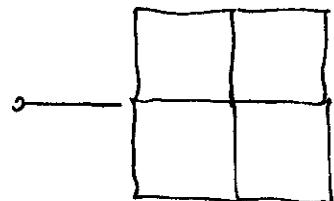


Now if d=D this:



$$R' = 2R$$

{ If d=0 this :



$$R'' = R$$

So, for given thickness, Resistance is same for  $\square$  of any size.

Thus we quote the "sheet resistivity in  $\Omega/\square$  (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.

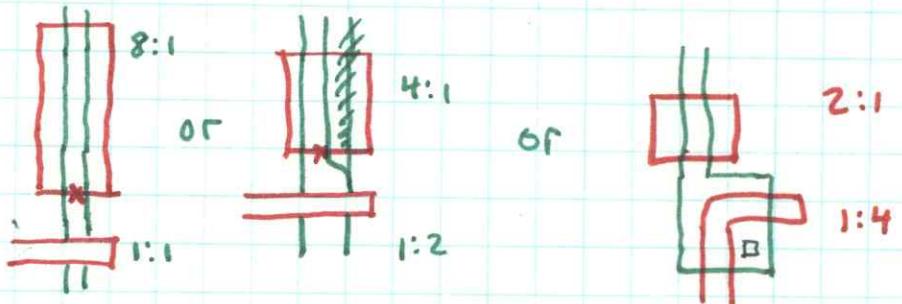


IN DIFF, at  $10\Omega/\square$ ,  
This would have  $R \approx 100\Omega$

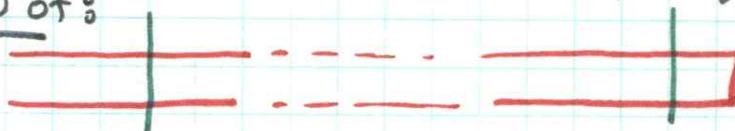
$R_M$	$\approx 0.1\Omega/\square$
$R_p$	$\approx 15 - 100\Omega/\square$
$R_d$	$\approx 10\Omega/\square$
$R_g$	$\approx 10^4\Omega/\square$

AYOUT IDEAS: • Variety of ways to obtain any given Pull-up / Pull-down Ratio:

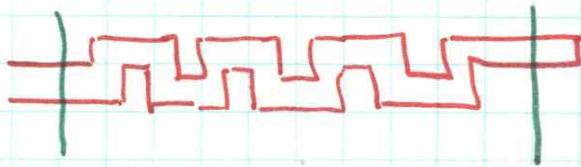
Consider: 8:1



- How would you make a really big pull-down? Instead of:

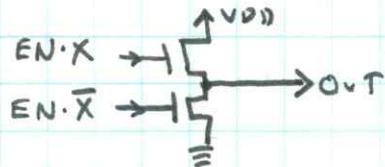


You could use a caterpillar:



(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 my dictate 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 calculation of  $T_{drive}$ )
- EXAMPLE: IN OUTPUT DRIVER, MAY WANT REALLY BIG T'S TO DRIVE OFF-CHIP CAP. LOAD: CONSIDER "TRI-STATE DRIVER:



**SHOW SLIDE**

(will use something like this)  
(only single-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  $10^5 \text{ A/cm}^2$   
i.e. [a few milliamperes /  $\mu\text{m}^2$ ]

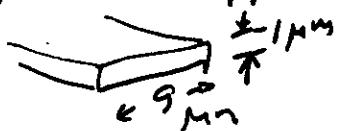
Let's use a limit in this course of  $1 \text{ mA}/\mu\text{m}^2$ . Note the scaling we will predict shows Vertical dimension; all voltage scaling directly with the horizontal scaling. But power density will remain same. Thus current densities will increase.

If you want your syst. Design to last a while, you might use an even more conservative value, say [  $0.5 \text{ mA}/\mu\text{m}^2$  ]

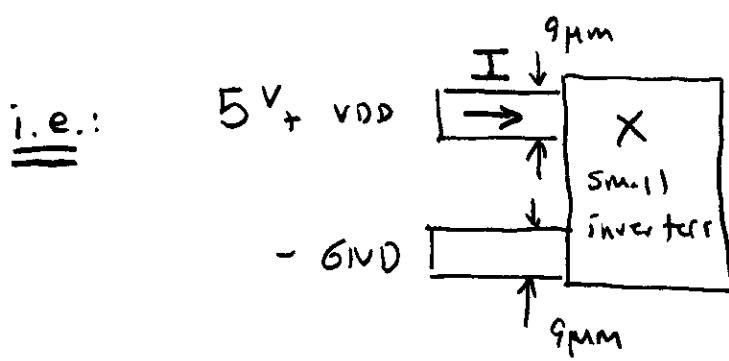
Unless someone finds a good way to fab high aspect ratio wires [  $9 \mu\text{m}$  ]

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

Metal Wires are  $3\lambda$  wide =  $9 \mu\text{m}$ , and  $\approx 1 \mu\text{m}$  thick



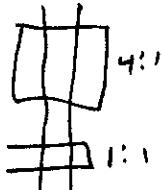
Consider the Question: [ How many minimum size 4:1 inverters could power lines of minimum size support? ]



what is X  
when  $I/\text{area} = 0.5 \mu\text{A}$   
?

for 4:1 inverters:

{min size}  
(pull down)



$$\text{ON Resistance} = (4+1) 10^4 \Omega = 5 \times 10^4 \Omega$$

$$\therefore I = \frac{5}{5 \times 10^4} = 10^{-4} \text{a} = 0.1 \text{mA}$$

- So a wire 9μm wide ~~can't~~ has cross sect.  $\text{dn} = 9 \text{mm}^2$ ,

and can supply  $9 \times 0.5 \text{mA} = 4.5 \text{mA}$  So  $X \sim 50$

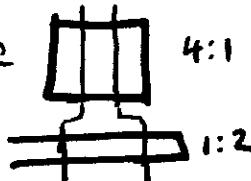
- But usually half are on, half are off

So  $X \sim 100$

- And if they are 8:1 rather than 4:1 :

$X \sim 200$

- But if make 8:1's like then  $X \sim 100$



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

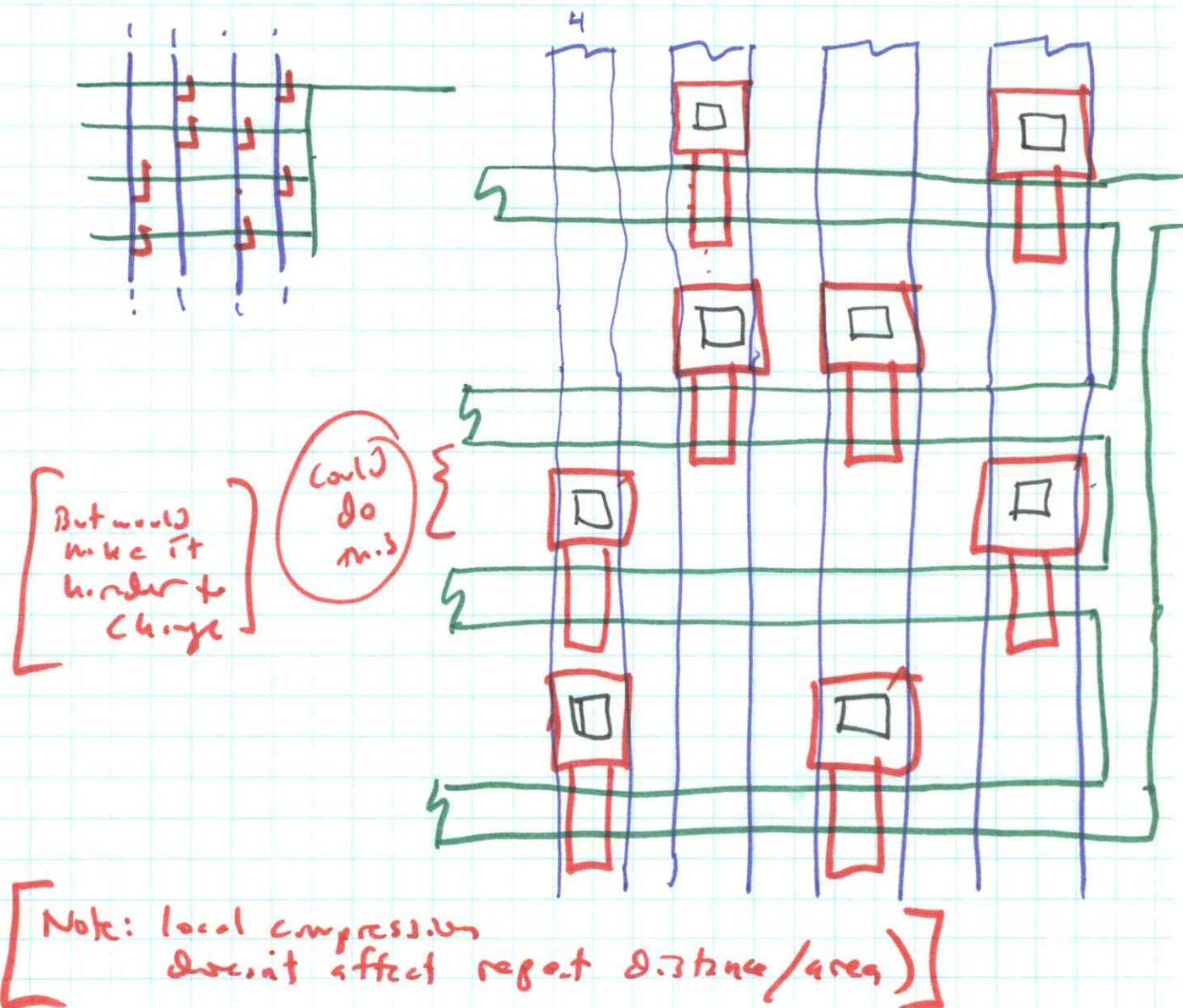
As we move up to larger subsystems, and groups of them, we at some point must start calculating the current requirements and widening the power mains appropriately:

**SLIDE: FRONTISPICE**

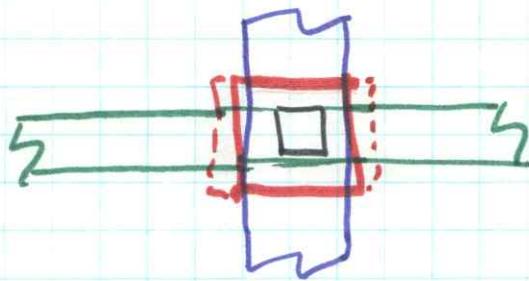
**SLIDE: CHAP 5 FIG 24**

- 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 week)   
 For now: look for pull-ups / sh<sub>tr</sub><sup>10j,2</sup>
- Don't use poly or diff at these max densities. If must make crossover/crossbar use Diffusion. Within it a bit. Keep it short. Else if >10 $\Omega$ 's of diff --- it will start dropping voltage significantly.

Layout EXAMPLE: on white board:



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

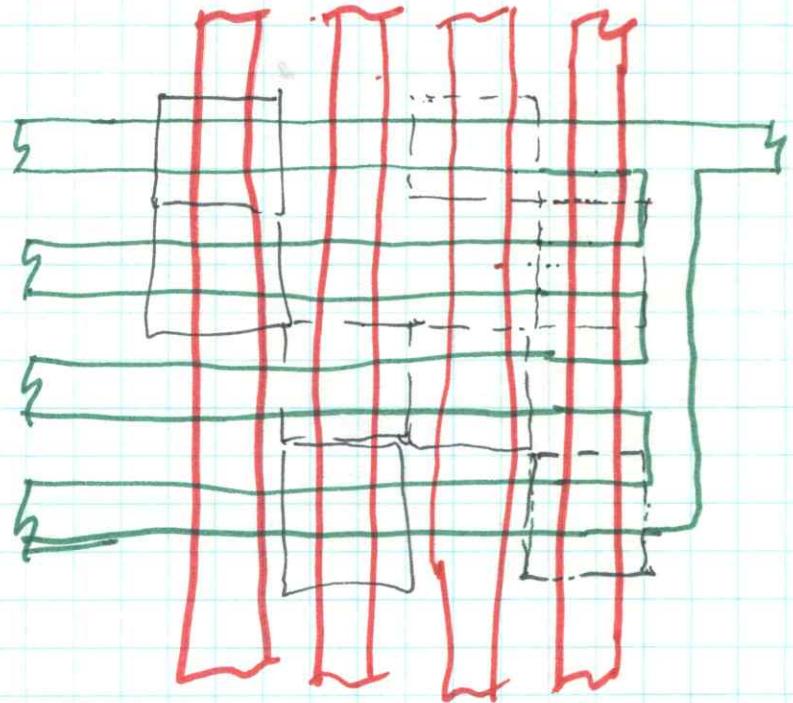
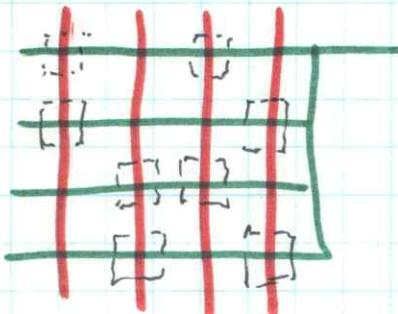


[BUT recommend if do this, enlarge to 1.5 to 2 lambda at edges over green]  
(Don't want to short out)

[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 ~~as~~s, AND MORE & MORE FINE STRUCTURE) CONFLICTS NOT ONLY WITH DESIGN TIME BUT ALSO LAYOUT DESCRIPTION (AMOUNT OF CODE).

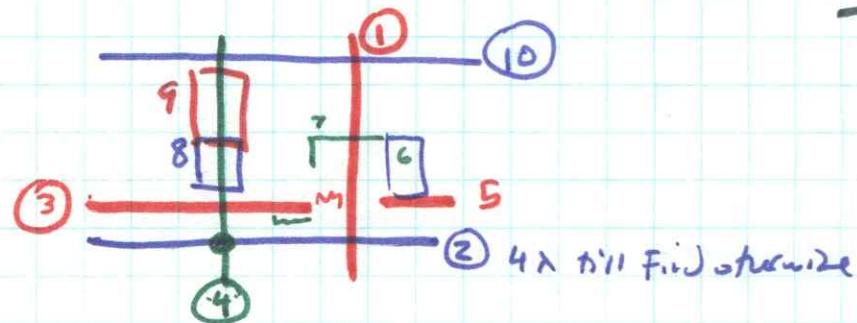
BIOLOGY ANALOGY: WANT SUBSYSTEMS THAT PERFORM FUNCTIONS USING ARRAY OF SINGLE CELL TYPE, PERHAPS SURROUNDED BY INTERFACE CELLS, ALL OF WHICH IN ADDITION TO SPEED LOW POWERED ET2 IS DERIVABLE IN MIN. AMOUNT OF CODE (GENB!)

### 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  $21\lambda$  wide? why isn't it  $20\lambda$  wide?  
[ $3\lambda$  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. Room Diff]

> this version draws a lot of power. If used  $16\lambda : 2\lambda$  pu and  $2\lambda : 2\lambda$  pull-down, would use  $\approx 1/3$  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 better - might be able to angle the pullup around a b.t.

> 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 8b chap 4]  
SHADE IN COLORS LIGHTLY

If you want to apply another constraint, do so let state it!

(SHOW SLIDE)  
OF TOPOLOGY OF STACK

SUGGEST TRACING DIRECT ; MIN. ORDERED CELL  
TO CHECK CELL-CELL DESIGN RULE



## 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 committing to all that work !!!)

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

(examples, in order of small to moderate <sup>first</sup> 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

[• I'LL 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 FOR

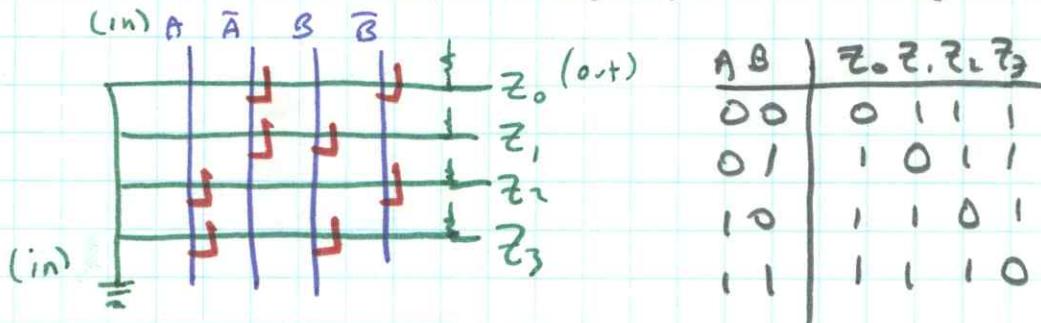
- Start Thinking About projects. During next two weeks you'll finish all the boxes you need to begin. Also we'll see more examples. Perhaps sketch out any ideas of interest. I urge you all to find collaborators; for at least design checking. Talk to others - it may help you get in the - if you have more 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 Ideas?

# 6.978 . LECTURE #7

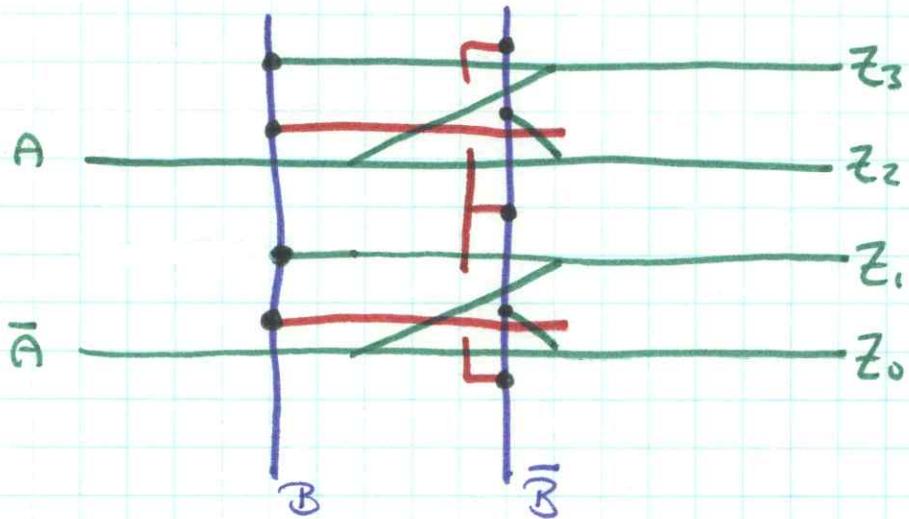
TUES OCT 3

- 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 inputs/output, and hook up VDD & GND:



> 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  $Z_0$ , if  $B=0$  blocks more up and more down, and  $\bar{B}$  passes  $B=0$  to  $Z_3$ .

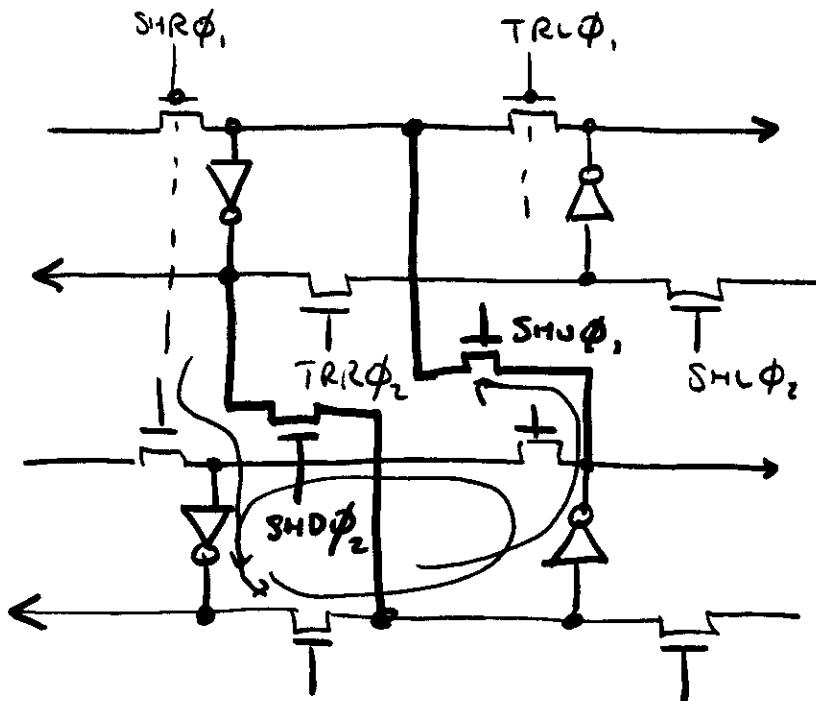
If  $\bar{A}=1$ , then  $B=1$  passes to  $Z_1$ ,  $\bar{B}=0$  blocks more through,  $\bar{B}=1$  passes  $\bar{B}=0$  to  $Z_0$ .

, etc.

## HW #2 (cont.)

> THE 2-D STACK:

(Prob 6)



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

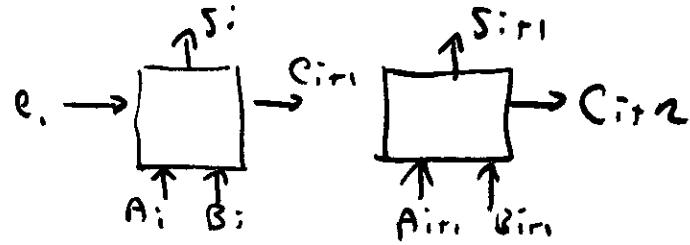
[Note that inverter output goes thru clocked line into node having no other inputs active, and outputs blocked by unclocked lines. For example, this doesn't happen

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

> MANY<sup>(stack)</sup> layouts possible. Don't know what's best layout.

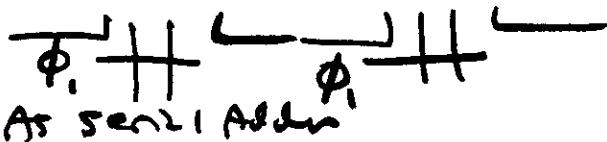
> The Adder: Most people got ~right answers for details of PLA's for problems 7 & 8.

However:



Then 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, i 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 w/ the next. Example: the "clocking of the adder."

FROM LAST TIME: Pushed 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):

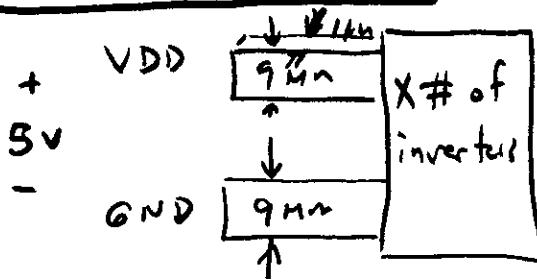
Question: How many minimum sized inverters will a minimum size VDD --- GND wire supply?

ON SLIDES

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



Have a bunch of inverters:



For 4:1 inverter,

$$R \approx (4+1)10^4 = 5 \times 10^4 \Omega$$

$$I = \frac{5}{5 \times 10^4} = 0.1 \text{ mA}$$

> LIMIT  $\approx 0.5 \text{ mA}/\mu m^2$ ,  $\therefore 9\mu m$  wire carries  $\approx 5 \text{ mA}$

$$\therefore X \approx 50$$

> But Half are usually off

$$\therefore X \approx 100$$

> And if 8:1's, then

$$\text{then } X \approx 200$$

> But if make 8:1's like

$$\text{then } X \approx 100$$

## 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 FN 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 DO! --- WE GET TO THAT NEXT WEEK).
- > SO LET'S GO BACK AND MAKE SURE WE UNDERSTAND DELAYS IN DRIVING CAPACITIVE LOADS AT IT 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  $C_L/C_g = Y$ .

Intr.ately, 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  $\frac{T}{f}$

(first known  
we'll see why  
later)

③ If inverter delay is  $T$  (or prop. to  $T$  log<sub>e</sub>(k) for  $k$  constant)  
 then  $N$  such stages have a delay of  $= NfT$ . [because in 5 inv. & after]

④ But  $f^N = Y = \frac{C_L}{C_S}$

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

If we use small  $f$ , need more stages, but each will have shorter delay.  
 Support  $Y=16$ , calculate  $f=2, N=4$   
 or  $f=4, N=2$ .

What  $N$  minimizes Overall delay for given  $Y$ ?

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

$$\text{Delay at one stage} = fT$$

$$\text{Total delay} = NfT = \ln(Y) \left[ \frac{f}{\ln(f)} \right] T$$

∴ Delay is proportional to  $\ln Y$

Figure 5 plots  $\frac{f}{\ln(f)}$  as function of  $f$ ,

normalized to its minimum value of  $e$

The minimum total delay =  $T$  times  $e$  times  
 natural  $\ln C_L/C_S$

$$\boxed{\text{Min Tot. Del.} \sim T e \ln \left[ \frac{C_L}{C_S} \right]} \quad \text{when } f=e$$

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  $\approx f = 6$ , Almost as fast, but less area.
- > Show DM DRIVER SLIDE

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

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

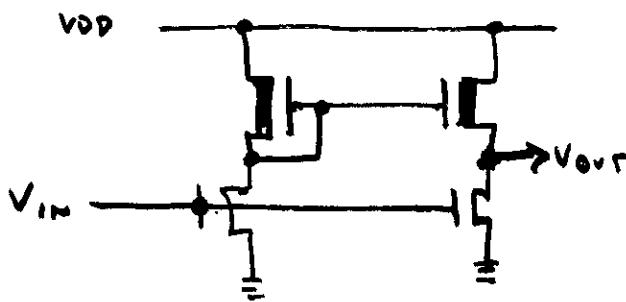
There will be further important uses of this simple set of ideas in the development of a Theory of the space, time, and energy

costs of computation in hierarchically organized systems, In Chapter 8

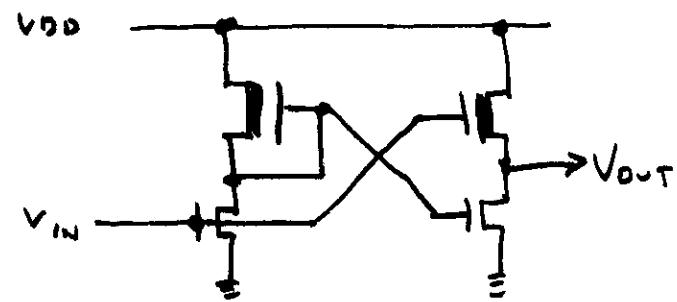
pp 49-57

## SUPER BUFFERS

- Ratio logic as we've seen has an asymmetry: It can discharge a  $C_L$  thru its pull-down much faster than it can charge one thru its pull-up.
- 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

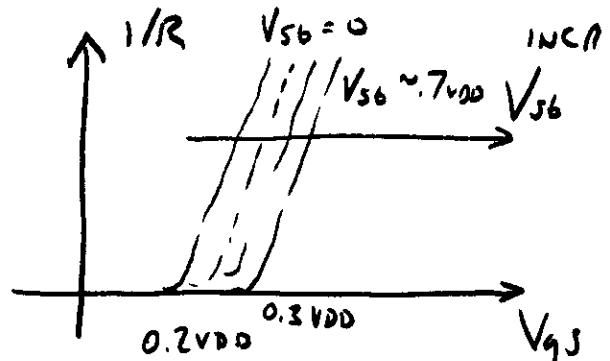
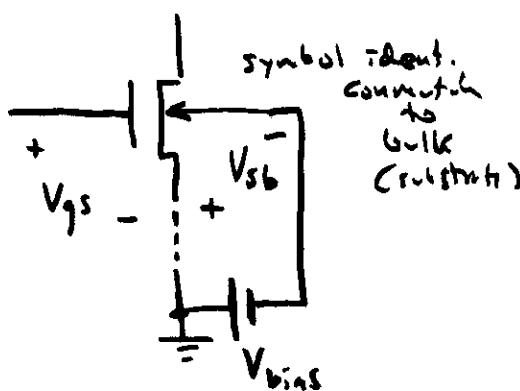


NON-INVERTING SUPER BUFFER

- IN EACH CASE, WHEN PULLDOWN INPUT TO 2ND STAGE IS  $\sim V_{DD}$ , THEN THE PULLUP GATE  $\sim 0V$  AS IN USUAL INVERTER.  
SO PULLDOWN THE SAME
- BUT, WHEN PULLDOWN INPUT TO 2ND STAGE GOES TO ZERO,  
THE PULLUP GATE GOES RAPIDLY TO  $V_{DD}$ , SINCE IT IS  
ONLY LOAD ON PREVIOUS STAGE.
  - > The pullup is TURNED on with  $\sim 2 \times$  the voltage it would normally have with gate tied to source.
  - > Since in saturation,  $I \propto V_{GS}^2$ , the SUPER BUFFER PULLS UP about  $4 \times$  as fast as regular inverter.
  - > i.e., pullup/pulldown are  $\approx$  symmetrical.  $\beta = \gamma$   
[that's why we got away with  $\gamma$  in the last section]
  - some exp. says it very slow effect.

The Body Effect: We're considering  $V_{Th}$  to be a constant, independent of the source to substrate voltage (i.e., only  $\propto$  for  $V_{GS}$ ).

- It isn'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?

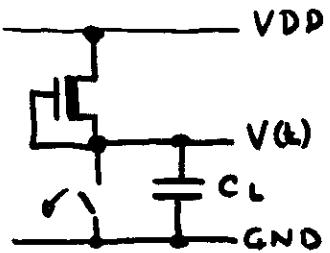
- Because it increases 2 effects we've already introduced informally:
  - ① The difference in time and final output voltage between an enh. & depl. mode FET driving a capacitor load.
  - ② The need to increase the Pull-up / Pull-down Z ratio, when coupling logic stages by pass transistors.

## DEPLETION MODE vs ENHANCEMENT MODE Pull-ups:

### 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 exponent:  $t$ :

$$V(t) = V_{DD} [1 - e^{-t/RC_L}]$$



i.e.,  $V(t)$  goes rapidly to  $V_{DD}$ , with time constant  $RC_L$ .

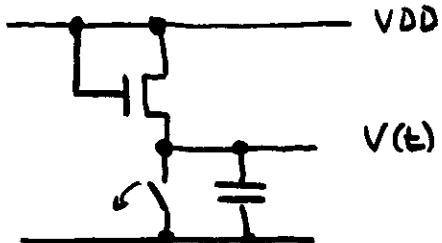
- For inverter ratio  $k$ , pull-down trantime  $T$  and gate C  $C_g$ , the time constant is

$$RC_L \approx kT C_L / C_g$$

### Enhancement Mode

(used in early MOS)

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



$$V(t) \approx V_{DD} - V_{th} - \frac{C_L L D}{\mu \epsilon W t}$$

SHOW SLIDE AND COMMENT

[worsened by body effect]

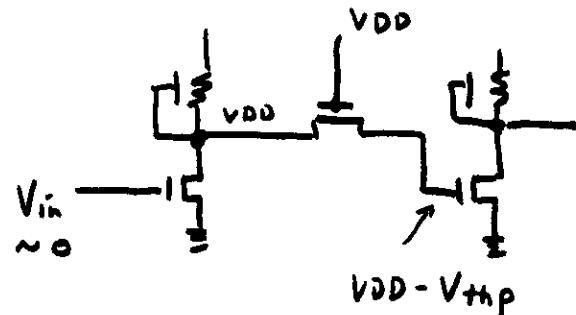
Depl. mode crit. goes to  $V_{DD}$ .  
Enh. slowly (over) until it reaches  $V_{DD} - V_{th}$ . (body effect)

## (IF TIME)

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

- We found earlier that  $4:1 = Z_{p+}/Z_{pd} = \frac{L_{pu}/W_{pa}}{L_{pd}/W_{pd}}$  yields equal inverter margins and also provides output sufficiently less than  $V_m$  for  $\text{input} = VDD$ .

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

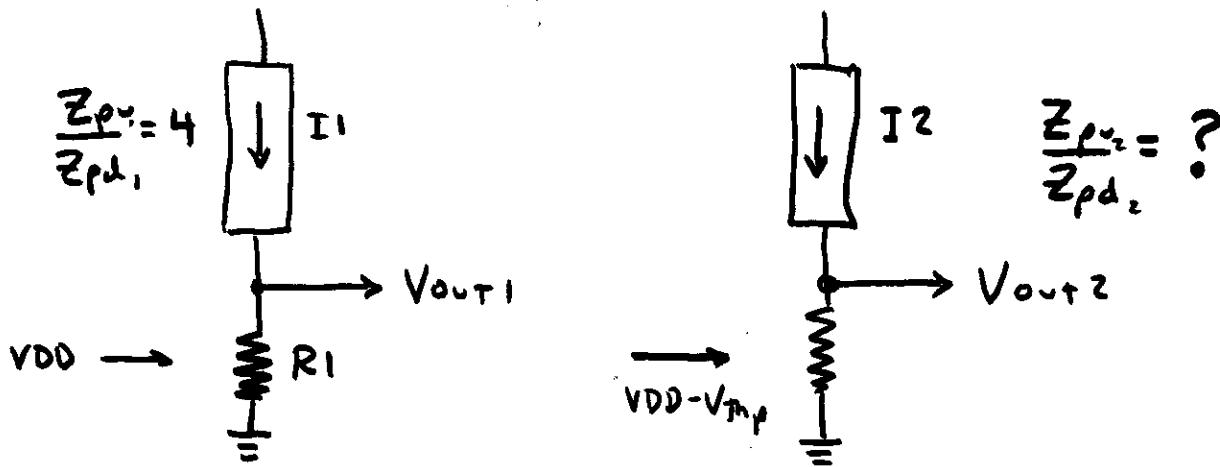


- If input to first is zero, and thus output  $\approx VDD$ . If Pass T input is  $VDD$ , then at most the input to the second stage is  $VDD - V_{Typ}$ .
- Since Pass T source is near  $VDD$ ,  $V_{Typ}$  is near its maximum of  $\approx 0.3VDD$ .

Question: What must be the  $Z_{p+}/Z_{pd}$  of second stage, if it is to have its output go as low with  $\text{input} = VDD - V_{Typ}$ , 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:  $I_{ds} = \frac{\mu \epsilon W}{2L D} (V_{gs} - V_{th})^2$  (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:

$$\frac{Z_{pu1}}{Z_{pd1}} [V_{DD} - V_{th}] = \frac{Z_{pu2}}{Z_{pd2}} [V_{DD} - V_{th} - V_{Thp}]$$

- Now:  $V_{th}$  of pull-downs is  $\approx 0.2 V_{DD}$ , and take safe.  $V_{Thp}$  of pass T's is  $\approx 0.3 V_{DD}$  to 0.35 V\_{DD}. BODY EFFECT

$$\therefore Z_{pu2}/Z_{pd2} \approx 2 Z_{pu1}/Z_{pd1}$$

$$\frac{Z_{pu2}}{Z_{pd2}} \approx 8:1$$

(WE USUALLY EITHER SPECIFY OR MEASURE)  
FOR INTEGRATED MFG., NOT FOR

~~WHAT HANDS OUT HONORABLE MENTION~~

~~IF TIME, THINK ABOUT PROJECTS A BIT.~~

REMEMBER, THINK  
[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 others others might use?]



## 6.978. LECTURE #8:

OCTOBER 5

- BE SURE YOU'RE REGISTERED ... WILL SET UP ACCOUNTS BASED ON WHOSE REGISTERED (OR MAIL) NETWORK
  - TODAY: EXAMPLES:
    - LAYOUT OF THE PLA
    - DESIGN OF THE BARREL SHIFTER
    - DESIGN IDEAS FOR A SERIAL BIT-STRING COMPARATOR
  - WERE ~1/3 THRU COURSE. SO FAR STARTUP TRANSIENT -- NEW STUFF -- HW
  - NEXT 1/3 BEGIN TO LEARN BY DOING -- STUDY EXAMPLES, DO INFORMAL SNFF IN LAB -- FIND OUT WHAT DO WE NEED TO KNOW --
  - FINAL '13 PRESUMPTIVE TO FINISH A PROJECT. SPECULATE ABOUT FUTURE
- 

- LETS USE WHAT LEARNED SO FAR, STUDY EXAMPLES:  
TO CLARIFY MATERIAL. 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 PLAT IN IT.
- LETS REVIEW THE STRUCT/FUN OF PLA, ; THEN DEVELOP A LAYOUT
- WE CAN ALL USE THIS LAYOUT IF WE LIST IN OUR DESIGNS
- THIS LAYOUT ISN'T "HARD" OR "TRICKY" IN TERMS OF DESIGN RULE CHECKING.  
BUT IT IS A NEAT<sup>1</sup>, COMPACT LAYOUT. THINGS JUST SORT OF FALL INTO PLACE
- THIS EXERCISE 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 let's review PLA design:

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

### 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 to PLA).
- Looks like the INPUT & OUTPUT parts are different, but each INPUT, each OUTPUT are just reflects.
- SO FAR, MAYBE JUST FOUR KINDS OF CELLS ???  
WE'LL SEE
- Now, REVIEW STICK DIAGRAM:

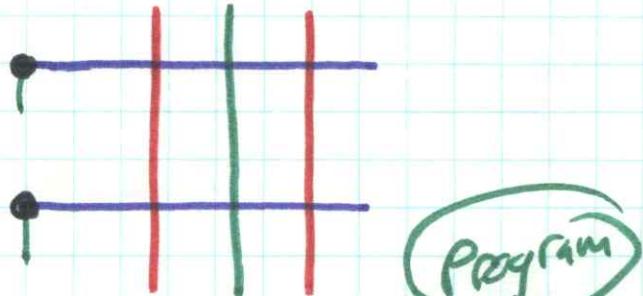
### SLIDE OF PLA STICK DIAG (TALK THRU STRUCTURE) & FUNCTION A BIT

- 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!

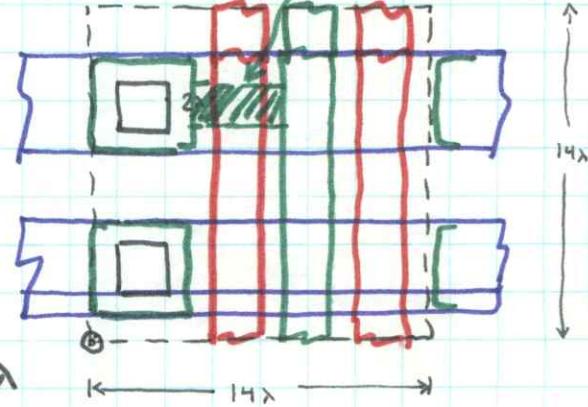
- We note that the most important region to do REGULARLY & COMPACTLY is probably the "PLANES":

[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")

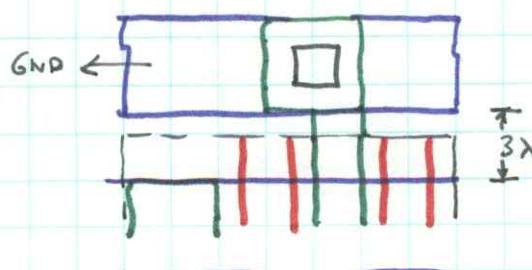


- This just reflects.
- How to lay out. Well, first put vertical reds and green as close as poss:
- Then put metal across this, as close as poss:  
But contacts get vs, so use  $4\lambda$
- PLACE DIFF FLASHES, DETERMINES PITCH OF CELL.
- So we've laid out a "PAIR of PLA Cells" For the AND/OR Planes **ITS SQUARE!**



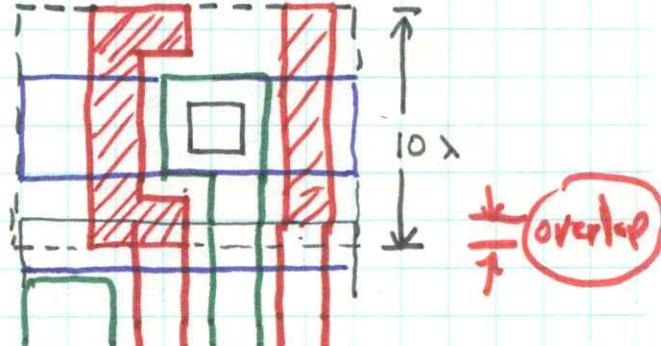
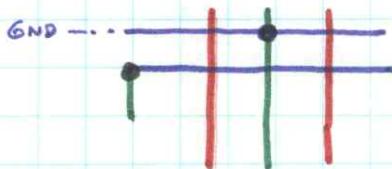
- WHAT ABOUT GND Return at edges of PLANES:

Could just use:



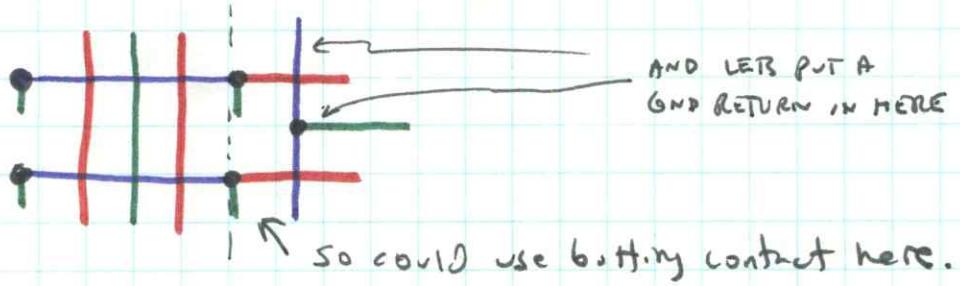
- BUT LET'S BE COOLER! IN A GIANT PLA, we'll need more GROUND RETURNS.

LETS MAKE A CELL THAT WE COULD USE AT INTERVALS, INSERTED AS ROWS, IN "AND" PLANE (cols in "OR" plane).

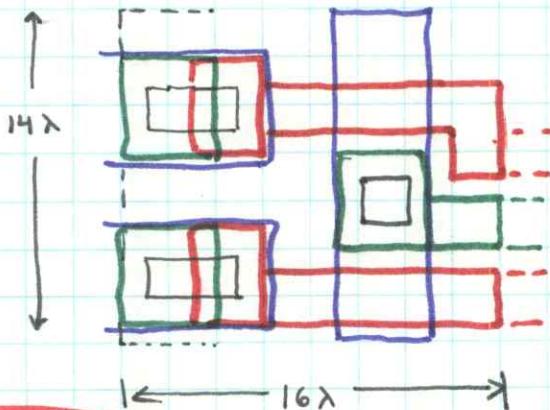


- LETS DESIGN CELLS TO CONNECT THE PLANES:

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  $14\lambda$  HIGH, WITH RED WIRES GOING OUT AT PROPER PLACE TO ENTER A "TILTED" PLA CELL PAIR:

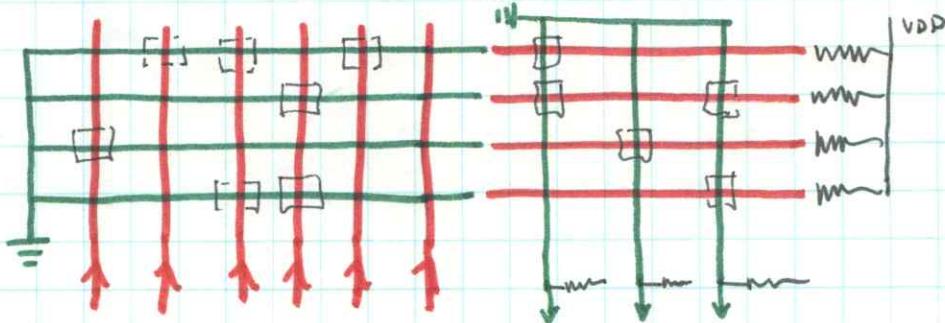


**WHILE SEEM SIMPLE,  
NOTE, THESE ARE  
FATHER TRICKY,  
MINIMUM LAYOUTS**

**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 RIGHT FOR SPACE, AND YOU CHECK  
YOUR DELAYS & PULLUP/PULLDOWN RATIOS CAREFULLY.

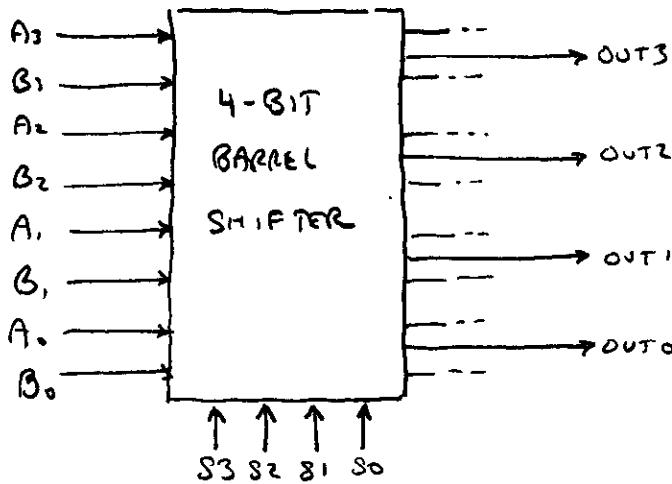
## THE BARREL SHIFTER

- 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 kludgy using purchased parts.

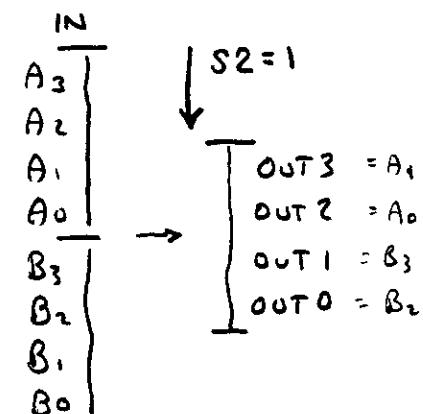
SHOW ON  
SLIDE

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.

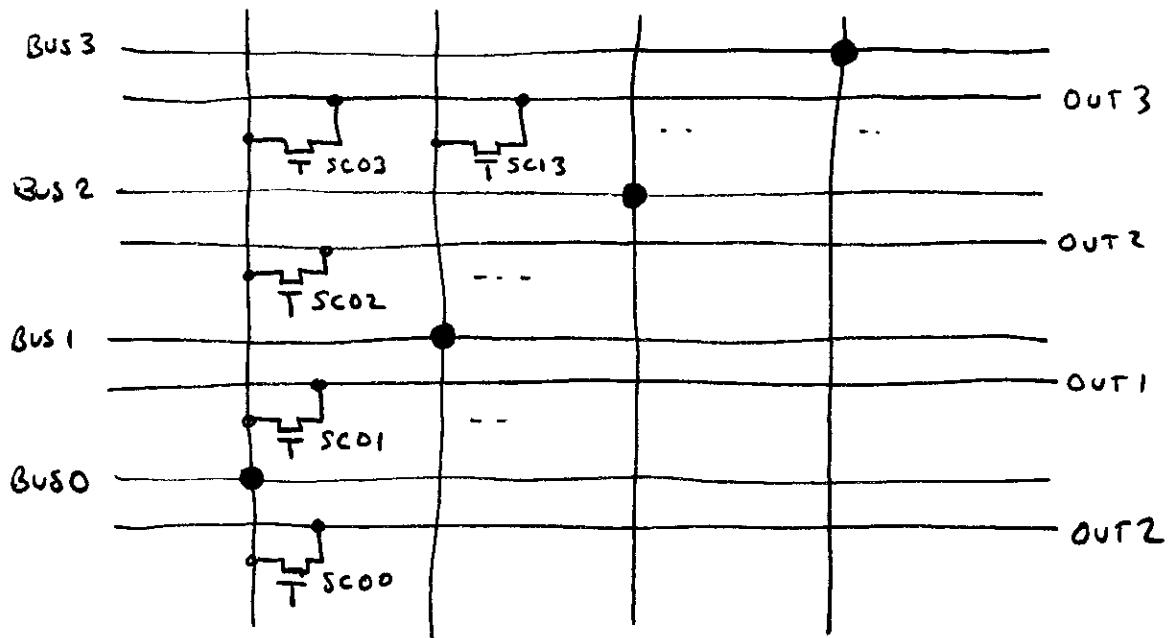
- Let's Describe and Design a FOUR BIT Barrel Shifter, [8 in 4 out] so we can keep the details in control & visualize what's happening.
- However, this design is directly extensible to a 16 bit Barrel Shifter.
- Block Diagram: Context: Embedded in a system. Two 4-bit Buses run right thru it!



CONCEPTUAL PICTURE OF FUNCTION:



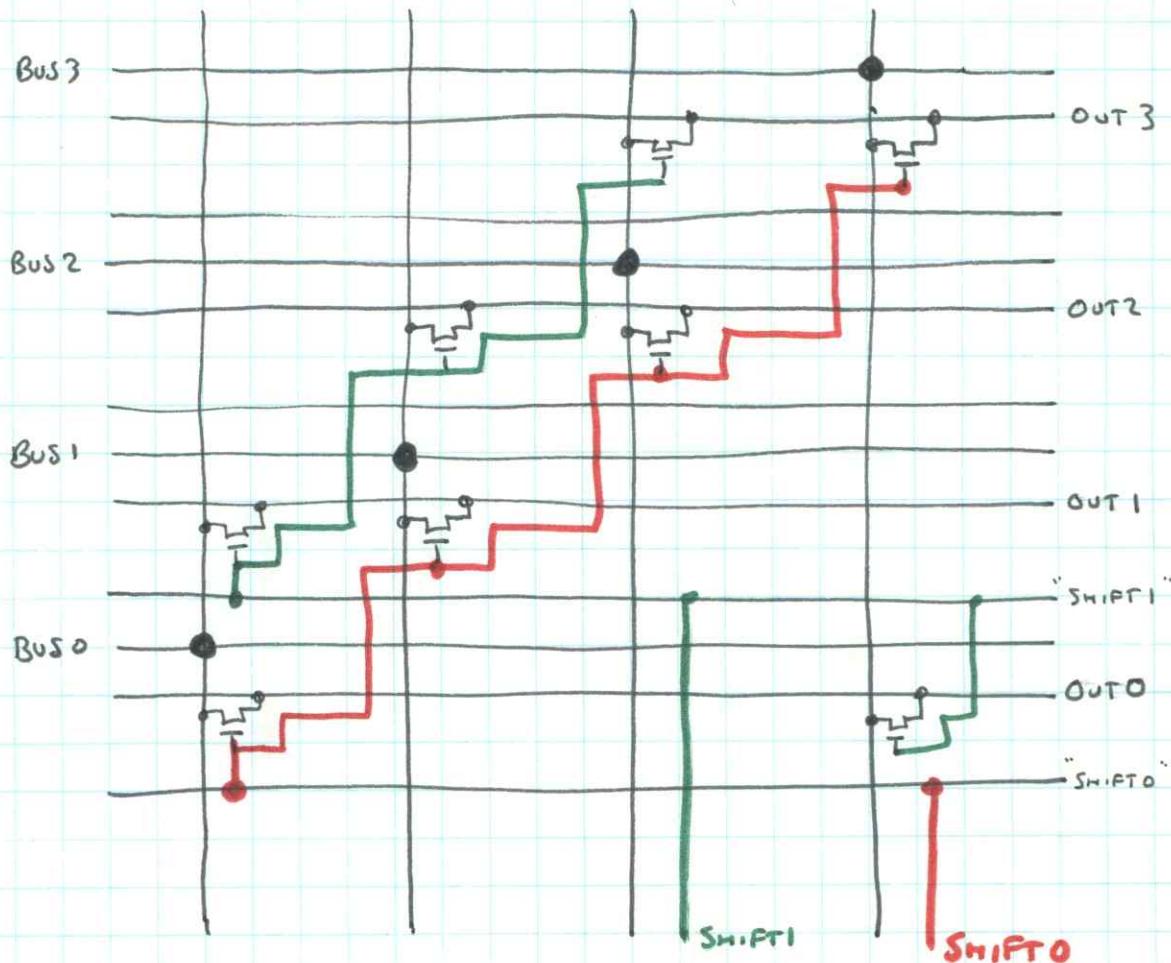
- NOW, WE NEED A WAY OF CONNECTING ANY BUS BIT WITH ANY OUTPUT BIT. THUS WE MUST HAVE DATA PATHS RUNNING VERTICALLY.
- A SIMPLE CIRCUIT IS A  $4 \times 4$  CROSSBAR --- THIS GIVES US SOME STARTING IDEAS:



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

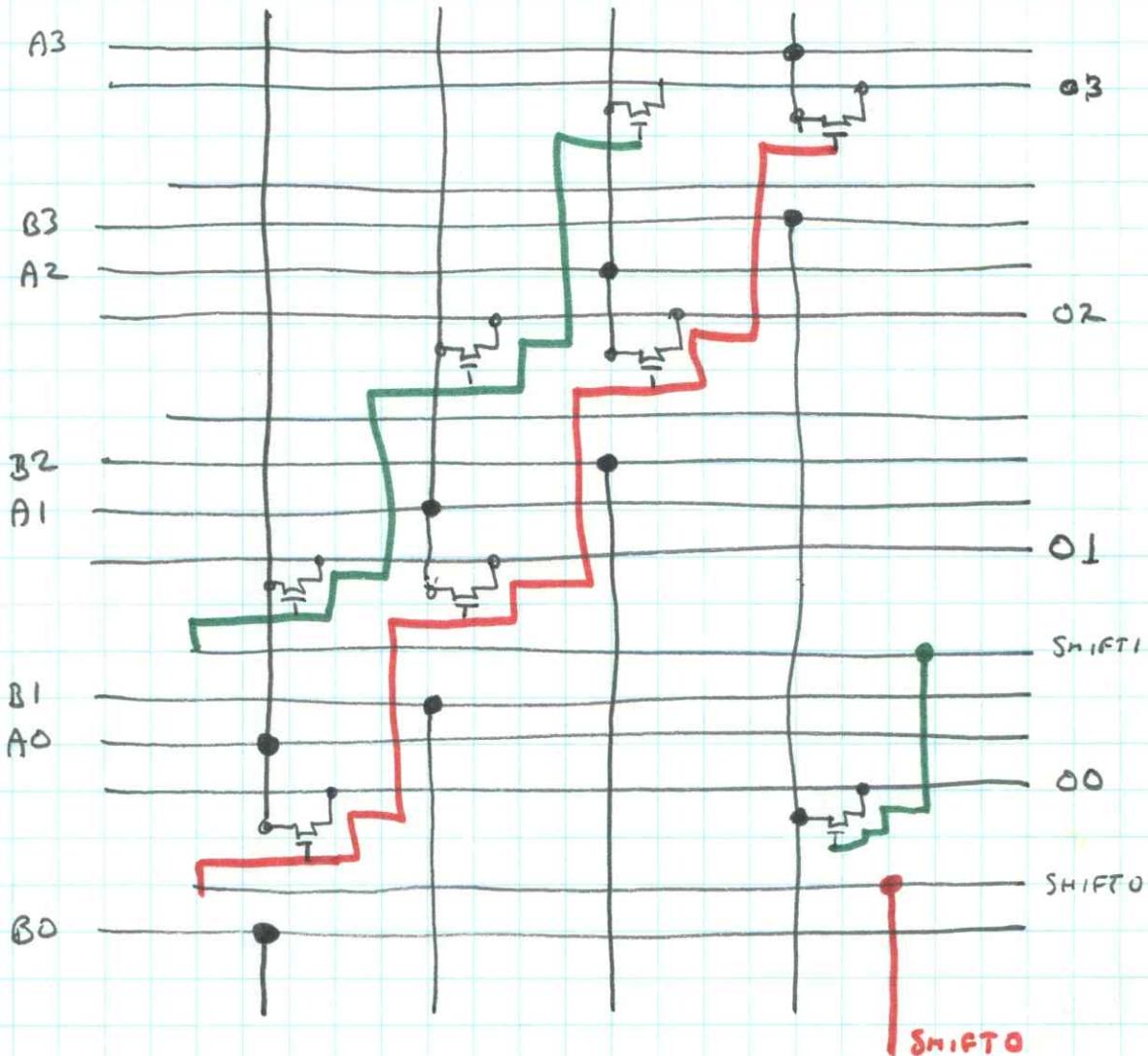
(cont.)

DRAW SAME DIAGRAM: WITH A FEW MORE LINES:



- DRAW IN ALL FETS TO CONNECT BUS<sub>i</sub> TO OUTPUT<sub>j</sub> ( $\text{SHIFT} = 0$ ) THEN HOOK THEIR GATES TOGETHER, ; TO A LINE CALLED SHIFT0
- DRAW IN ALL FETS TO CONNECT BUS<sub>i</sub> TO OUTPUT<sub>j</sub> ( $\text{SHIFT} = 1$ ) THEN HOOK THEIR GATES TOGETHER <sup>it1</sup> TO LINE CALLED SHIFT1
- 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<sub>i</sub> to OUT<sub>i</sub>. CONNECT GATES TO SHIFT0  
 > ALL " " BUS<sub>i</sub> to OUT<sub>i+1</sub>. " " " " SHIFT1

, etc.

- Note How Now ON SHIFT1, AO → O1

$$\begin{array}{l}
 \text{AO} \rightarrow O1 \\
 \text{A1} \rightarrow O2 \\
 \text{A2} \rightarrow O3 \\
 \text{B3} \rightarrow O0
 \end{array}$$

A3 ↑ Shift0 OK

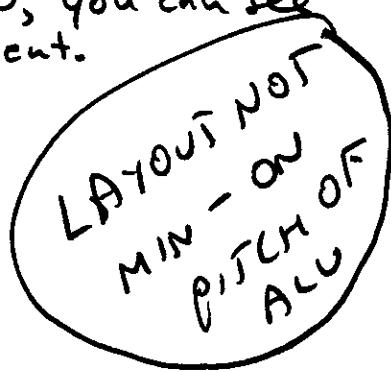
$$\begin{array}{l}
 \text{B3} \rightarrow O0 \\
 \text{B2} \rightarrow O1 \\
 \text{B1} \rightarrow O2 \\
 \text{B0} \rightarrow O3
 \end{array}$$

~~Shift0~~

?   
 ~~A3~~ looks strange!   
 OK!

AYOUT: The Barrel Shifter is one structure that seems easier to think of in circuit form. Try stick diagramming 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
- Adds 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 OUT PUPS.  
BASICALLY A SIMPLE TO CHECK LAYOUT



[THIS IS A VERY CLEVER SUBSYSTEM. I DON'T KNOW WHO ALL WORKED ON IT, BUT MOST OF THOSE NAMED IN THE SECTION "OM PROJECT AT CALTECH" HAD SOMETHING TO DO WITH IT.]

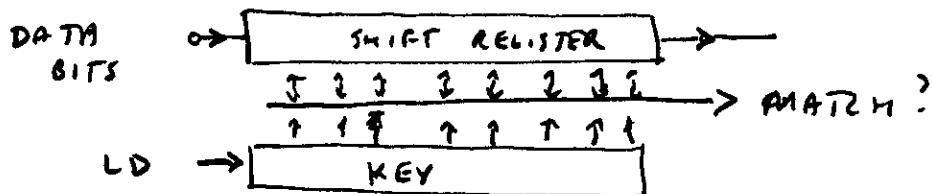
(A COUPLE OF MONTHS AGO I TALKED WITH DELL LATIN AND HEADS UP INTEL'S NEW ARCHITECTURE GROUP IN DALLAS --

(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 19<sup>TH</sup> CENTURY: A PERIOD OF GREAT MACH INVENTIONS: COLT'S REVOLVER, THE IDEAS ASSOC. w/ INTERNAL COMBUSTION ENGINES, ETC.

SO, WHILE MAYBE MUCH OF THIS WILL BE AUTOMATED, IT WILL BE (AT FIRST) AT HIGH OR LOWER LEVELS. BUT AT THE JUNCTURE OF SYSTEMS WITH THIS NEW TECHNOLOGY--- I THINK WE'RE GOING TO SEE A LOT MORE INTRIGUING INVENTIONS

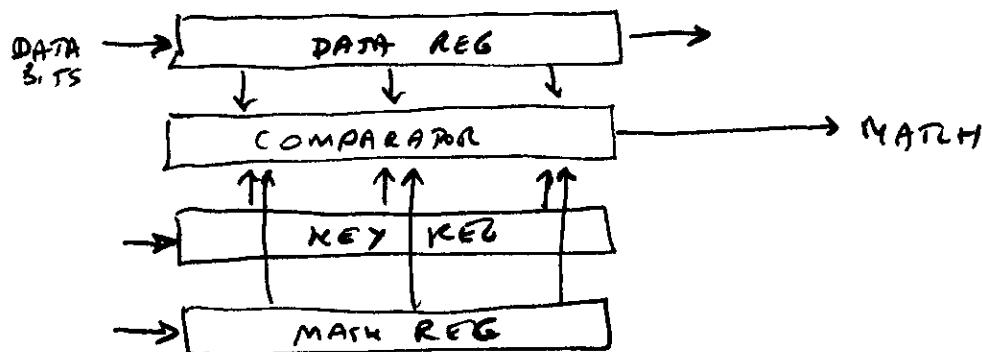
## A SERIAL BIT STREAM COMPARATOR:

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

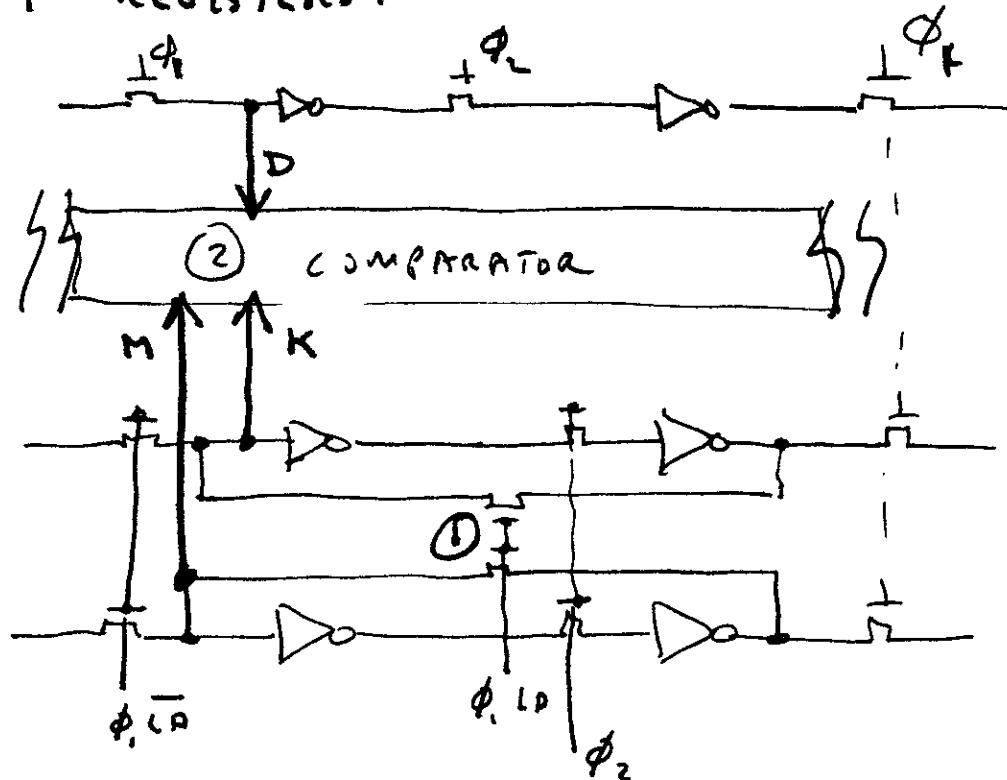
So:



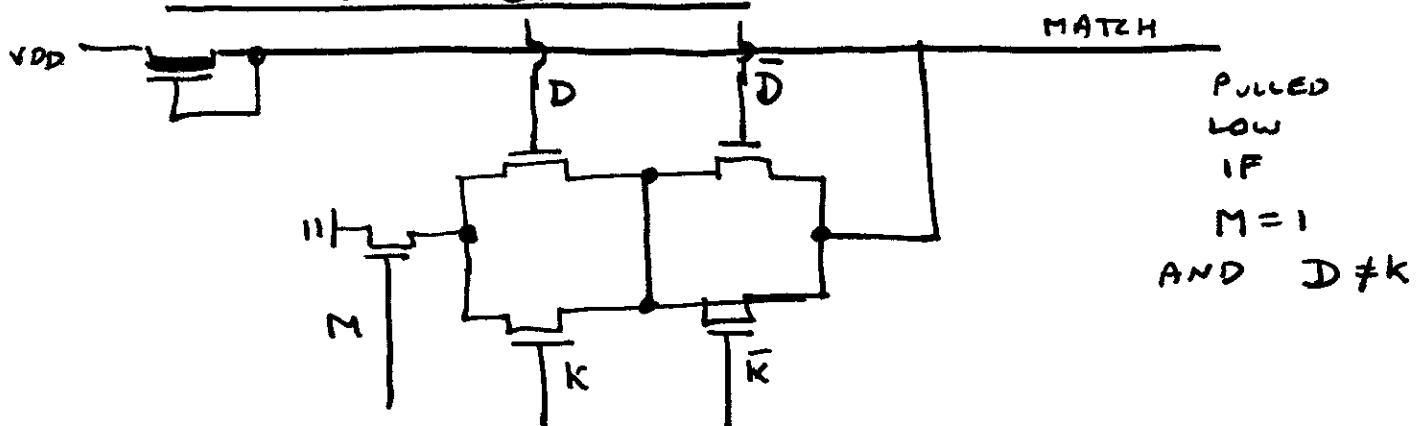
I.E., ONLY TRY TO MATCH THE SUBSET MARKED BY THE MASK BITS.

## 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:

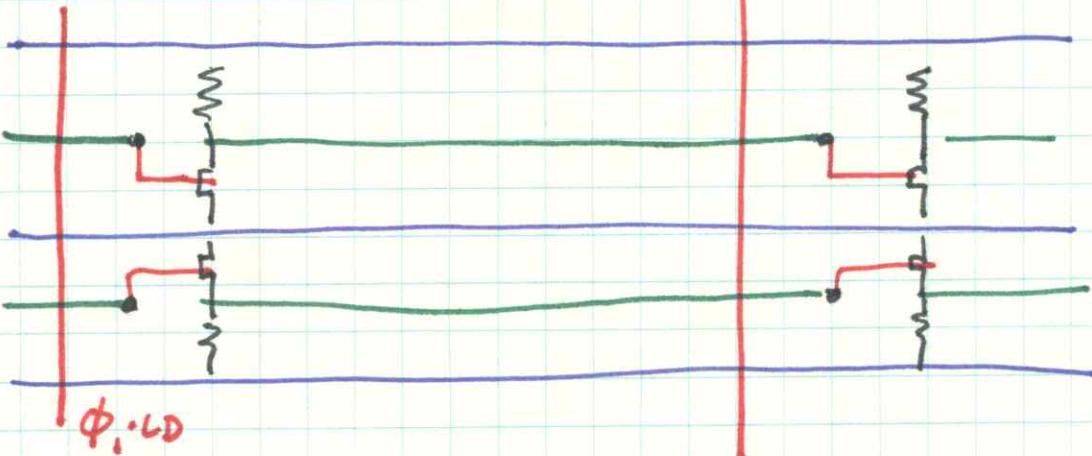
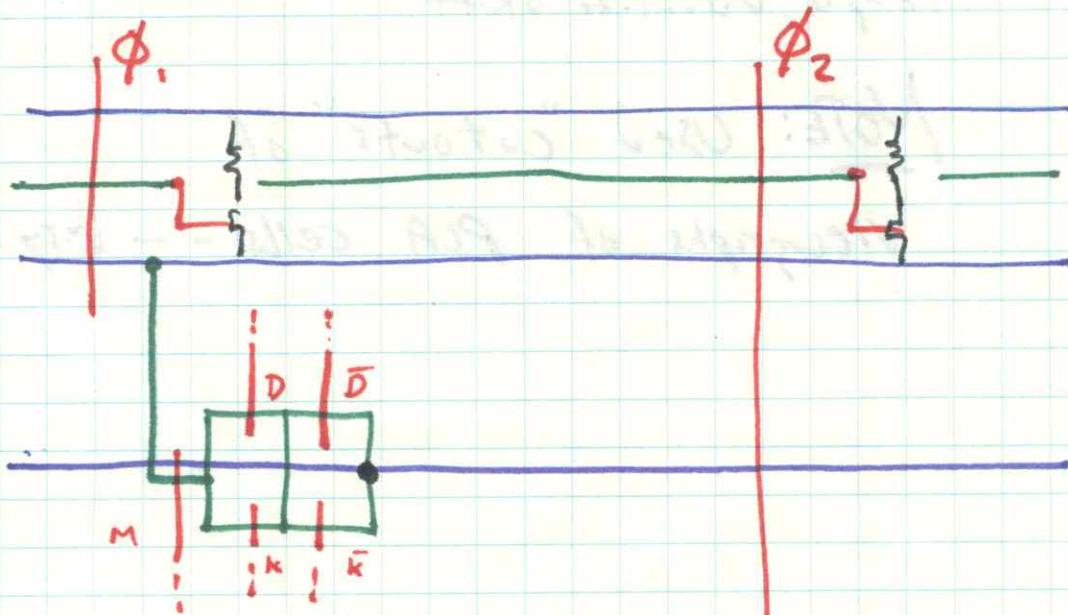


- MAKE LOWER TWO LOADABLE ①
  - BRING MASK, DATA, KEY TO COMPARATOR ②
  - NOW DESIGN COMPARATOR:



## HAND OUT H.W. #4

- NOW, HOW MIGHT WE STACK 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  $\phi_1 \cdot LD$  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 LAS 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 solution 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 ~~formats~~<sup>formats</sup> do. There were things unspecified in the problem: actually how to start the machine. Are one-bit messages allowed? These 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 initialized 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 bonding 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 + 5 only)

Lynn Bowen  $324\lambda^2$

Problem 11.

Gerald Raylance  $1302\lambda^2$

Michael Colm  $1357\lambda^2$

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 1, REPEAT TO GET MASKS,  
PROCESS, 3, 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:

~~(that were trying to describe a kind of the two dimensional object code ...)~~ We could use something like a macro-number where we define Macros and then call them to insert code in place.

Let's look at an informal example:

SLIDE OF SRC CELL

SHOW CELL ENCODING.

SLIDE OF CODE

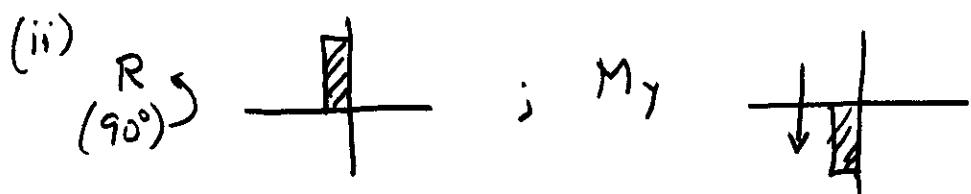
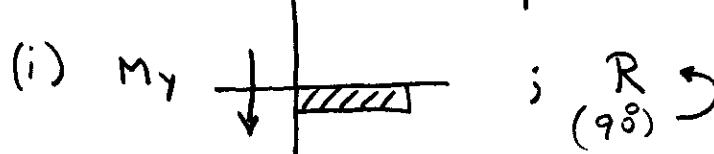
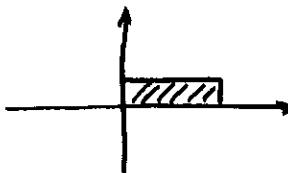
SHOW ITERATION, TRANS.  
TO GET AN ARRAY ...

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 MIRRINGS, 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.

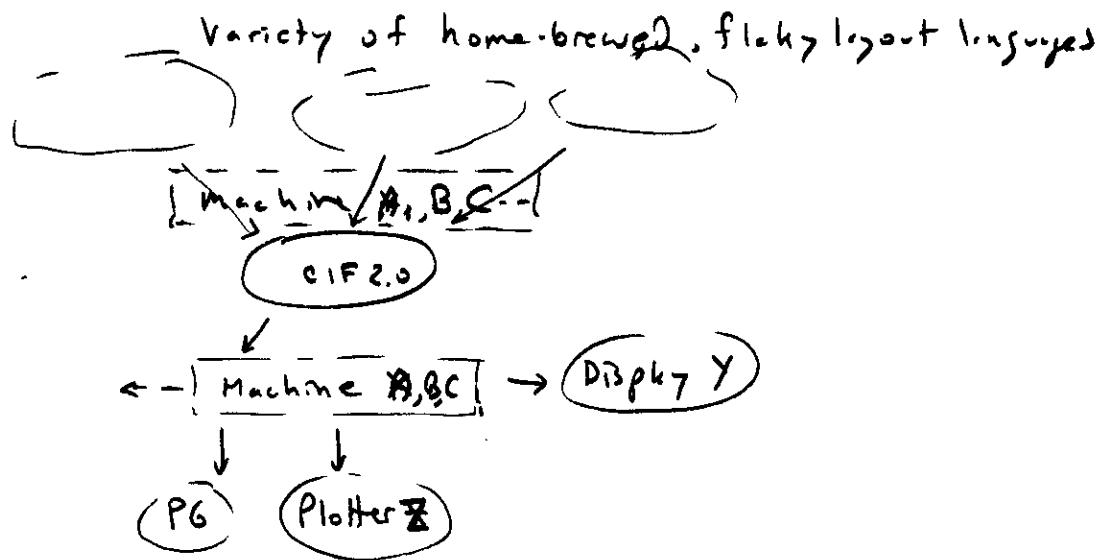
BACKGROUND:

The is no generally <sup>accepted</sup> ~~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. ~~FOR~~ ACTUALLY, THE SYMBOLIC LAYOUT APPROACH IF DONE AT A HIGH ENOUGH LEVEL, HAS ADVANTAGES NOT DIRECTLY AVAIL. IN STRAIGHT INTER-GRAFICS 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 ENHANCE THEIR LAYOUT CODING 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 in CIF2.0:  
Interpret

- Distances: Integer values in units of hundredths of a  $\mu\text{m}$
- Coordinates: Right handed coord. system, inc. 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  
]

- Recommended using shorter encoding:

BOX: B25 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;      or      LND;  
                                   LNP;  
                                   LNC;  
                                   LNM;  
                                   LNI;  
                                   LNB;  
                                   LNG;

- 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 layers 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 \* 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 mm. Or, use  $\lambda = 3\text{mm}$ , qtid. 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.
 

$C 57 M X R -1,1 T 10,20;$
- alternatively:
 

$C 57 M X R -1,1 T 10,20;$
- NOTE:  $C 1 T 500 0 M X$  adds 500 to the X coords, then mirror in X.  
 $C 1 M X T 500 0$  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  $3 \times 3$  transformation matrix:
 
$$\begin{bmatrix} X' & Y' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & 1 \end{bmatrix} T$$
- The transformation matrix  $T$  is simply the product of all the primitive transformation specified by the coll: i.e.  $T = T_1 \ T_2 \ T_3$ , etc.
- The primitive transformation matrices ~~are~~ obtained by using the following templates:

(Cont.)

## Primitive Transformation Templates:

$$T_{ab} \quad T_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix}$$

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

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

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

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.

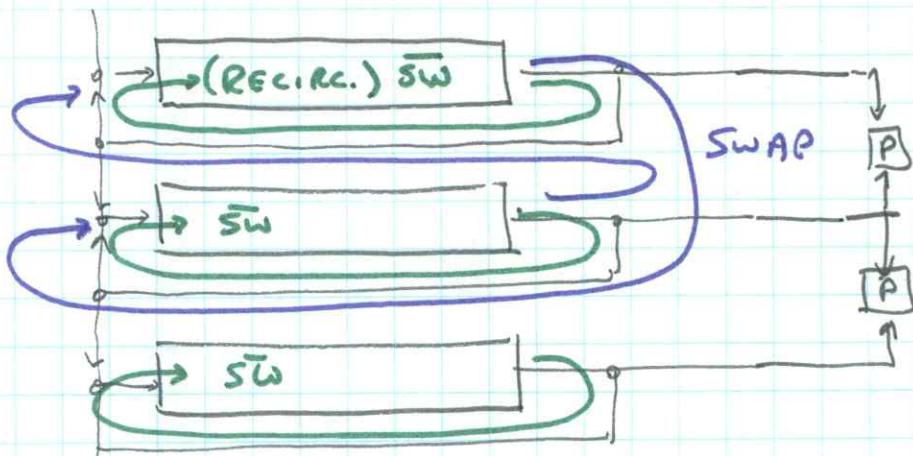
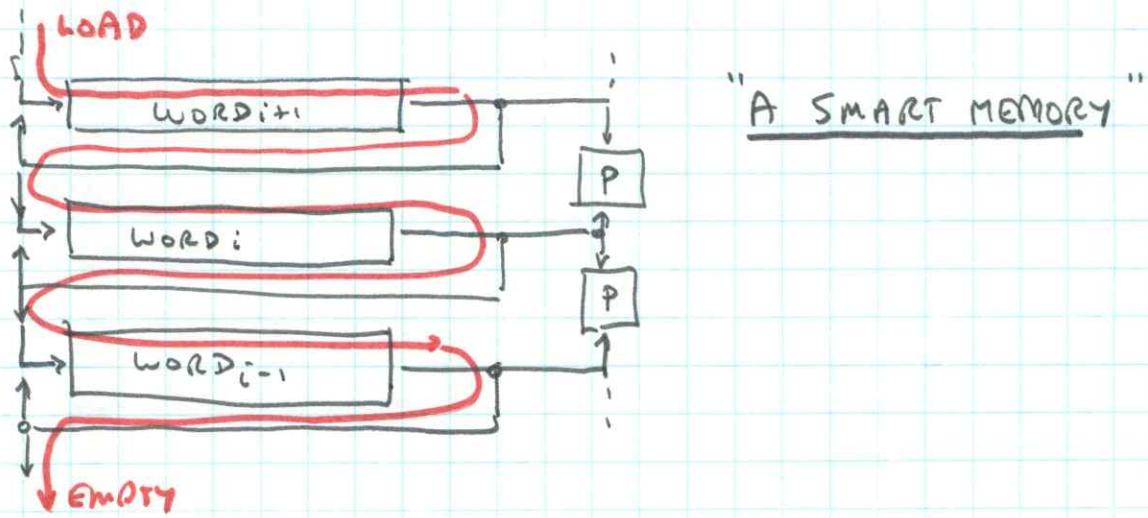
COMMENTS: Enclose in parentheses: (comment);

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

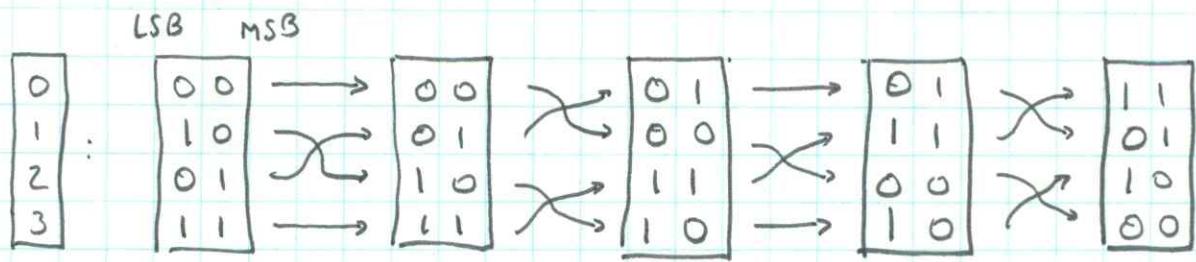
## DISCUSS NEXT HW ASSIGNMENT:

- 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

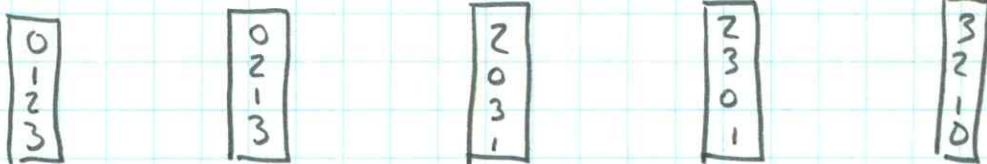
USE HW SET AS NOTES. DEVELOP STRUCTURE:



THE CONCEPT OF THE BUBBLE SORT: SUPPOSE LOADED:

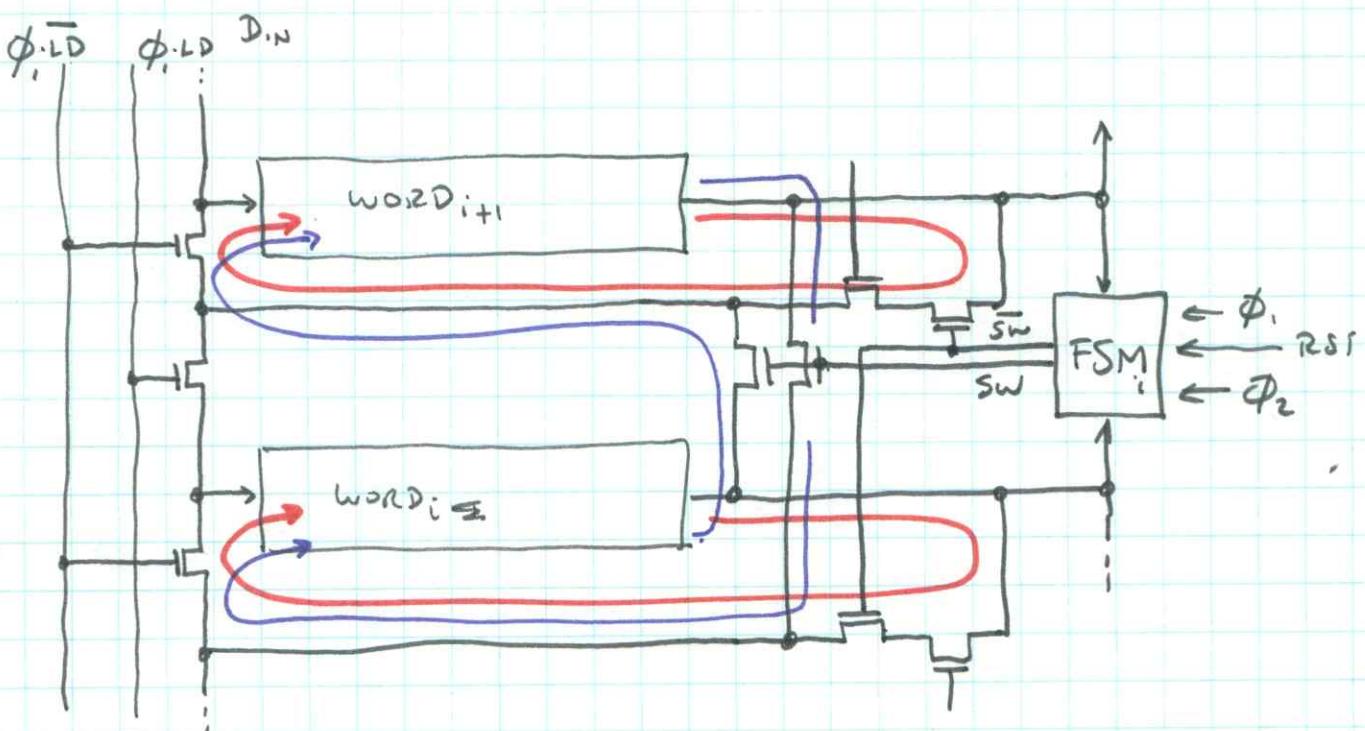


i.e:



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 thru this, in some detail.

But all you should do for this assignment is  
the circuit design of FSM's.

Be sure to correctly apply the  $^{2-\phi}$  clocking methodology,  
doing the right things in the right places.

- 
- 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. WHILE 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  $\frac{1}{2}$  OF FINAL. ON BASICS. THE HW ASSIGNED THIS TIME IS LAST BEFORE MIDTERM.

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

MULTIBIT COMP	}	~4600	Dean Brock	5280	Guy Steele
1 BIT VERT SLICE		~4600	Andy Boughton	5300	Alan Snyder
IN $\lambda^2$		~5160	Randy Bryant	5328	Siu Ho Lam
		5202	David Otten		

LAB: Glen Miranker (and) will open today ~3:30 to begin text editing  
Bill Henke: present more on CIF subset well 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!

(IN HW #5)

ON SLIDE OR

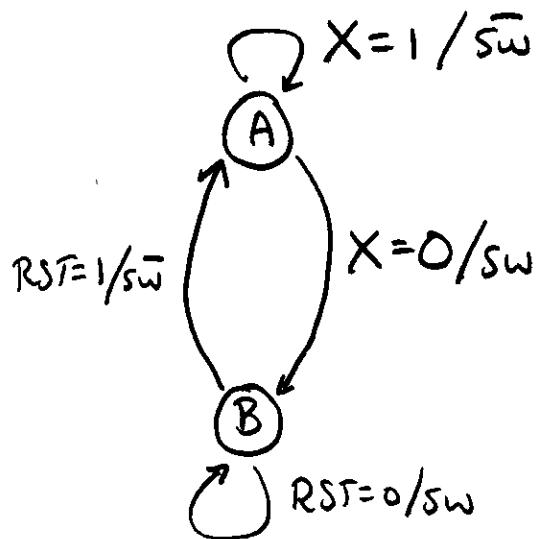
- HAVE THE SORTER FIGURE (ON BLACKBOARD.)

- 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 simple cases. Unfortunately there was a superficial error in the transitions listed.

- Lets correct that error and look at a simple circuit implementing that FSM.

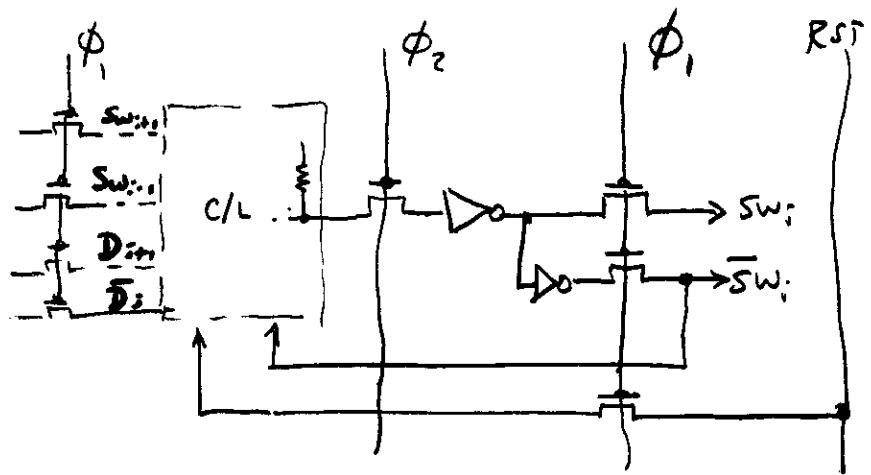
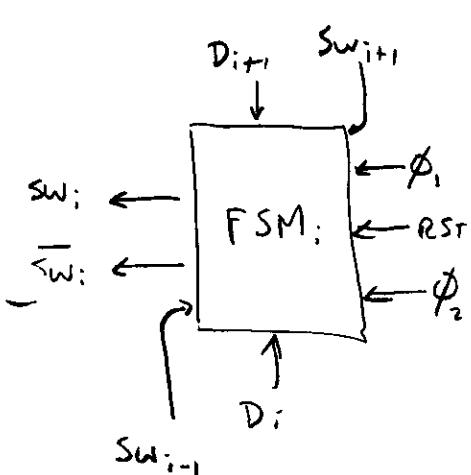


$$X = \text{sw}_{i+1} + \text{sw}_{i-1} + D_{i+1} + \overline{D}_i + \text{RST}$$

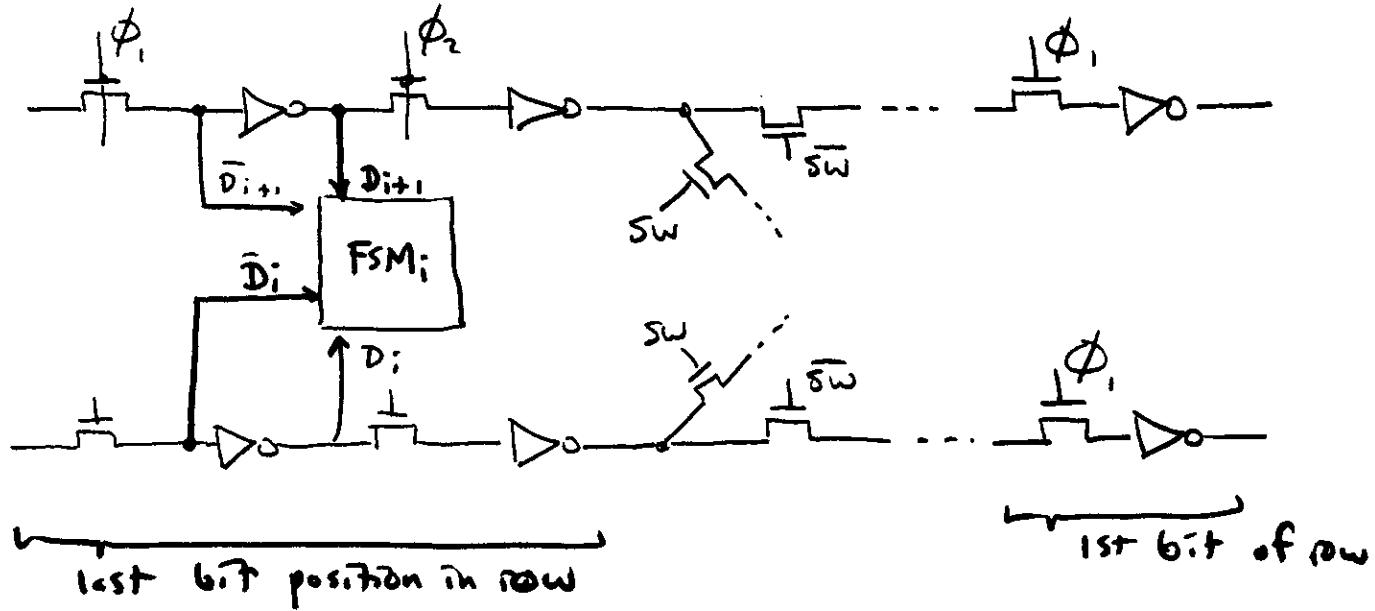
STATE A: NOT SWAPPING

STATE B: SWAPPING

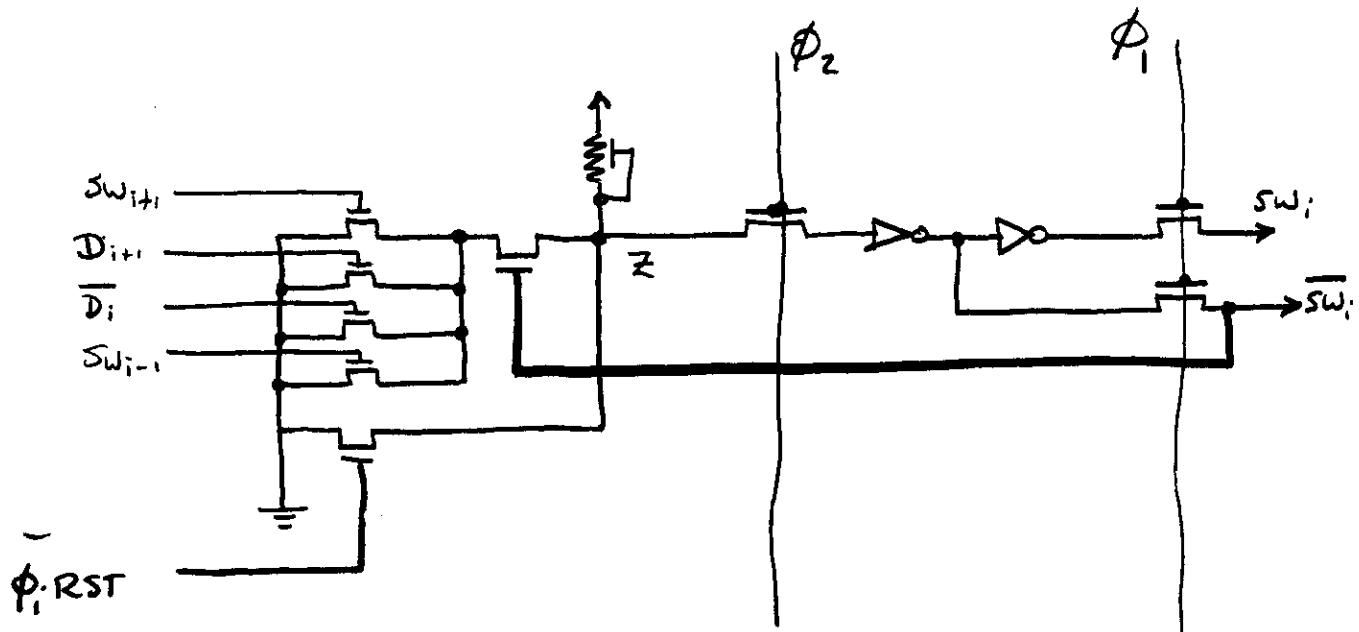
- FORM FOR INSERTION INTO THE SORTER:



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



- NOTE ALSO THAT THE OUTPUTS  $\bar{SW}_i$ ,  $SW_i$  ARE SET UP INTO THE SWITCHES DURING THE FOLLOWING  $\phi_i$ . ALSO, THEY INPUT ADJACENT FSM'S DURING " " $\phi_i$ .
- NOTE THAT A ROW SHOULD CONTAIN WORD<sub>i</sub> + 1 BITS
- HERE IS A SIMPLE CIRCUIT IMPLEMENTING FSM<sub>i</sub>:



- 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: ① Not swapping ( $y_{ct}$ )

- ② Swapping
- ③ 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 (1) even though they previously  
 encountered a (0).

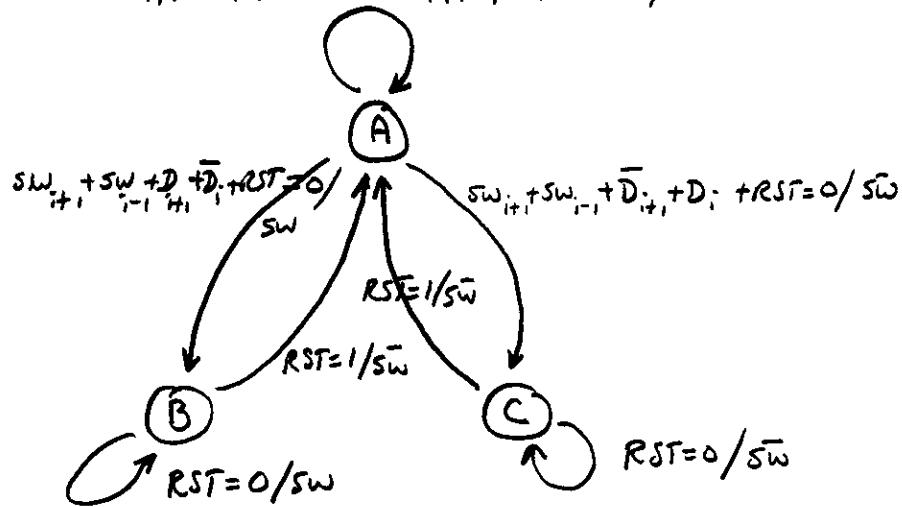
- So, we must add a state to "Remember Not To Swap"  
 If we encounter  $D_{it_1} = 0, D_i = 0$ .
- A STATE DIAGRAM FOR THE CORRECT THREE STATE FSM:  
 and an extension of the previous CIRCUIT diagram to implement  
 the three state FSM: is given in the next HW  
 assignment.

$$SW_{i+1} + SW_{i-1} + D_i, \bar{D}_i + \bar{D}_{i+1}, \bar{\bar{D}}_i + RST = 1 / \bar{SW}$$

5

### STATE DIAGRAM:

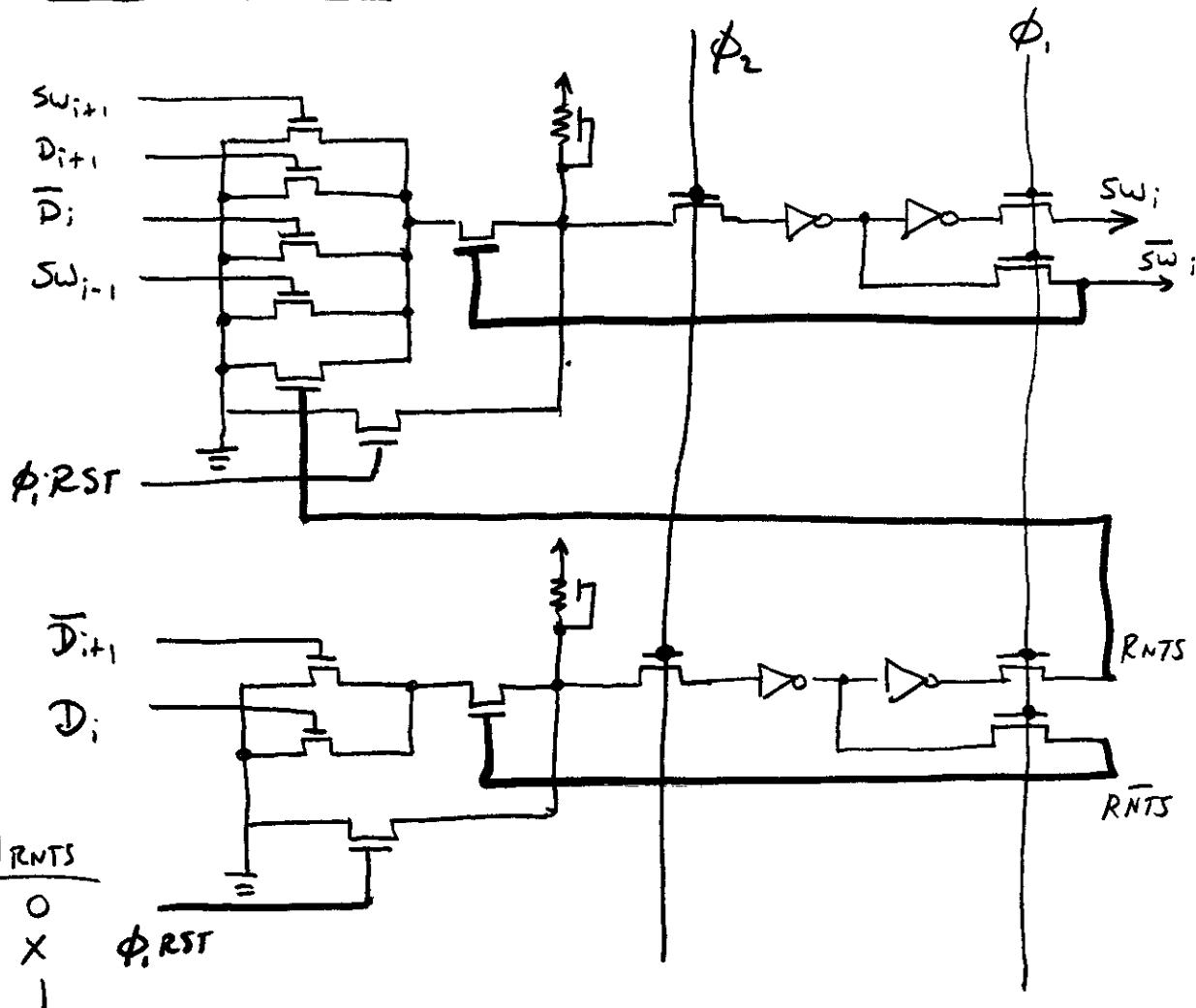
- A: NOT YET SWAPPING/RESET
- B: SWAPPING
- C: REMEMBER NOT TO SWAP



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)

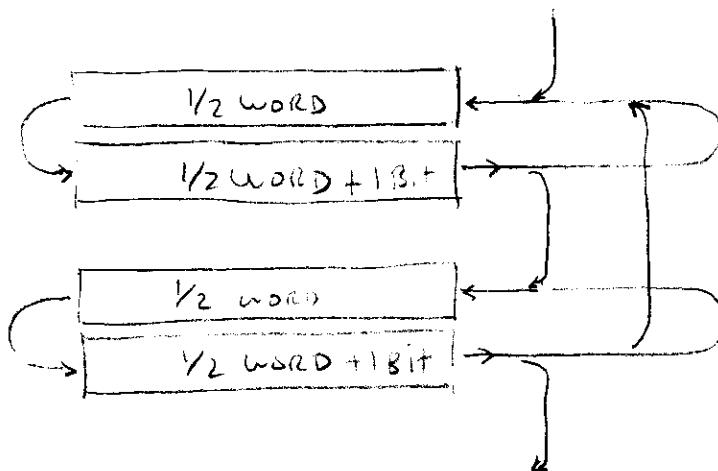
### ONE POSSIBLE CIRCUIT; IMPLEMENTING FSM:



- 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:

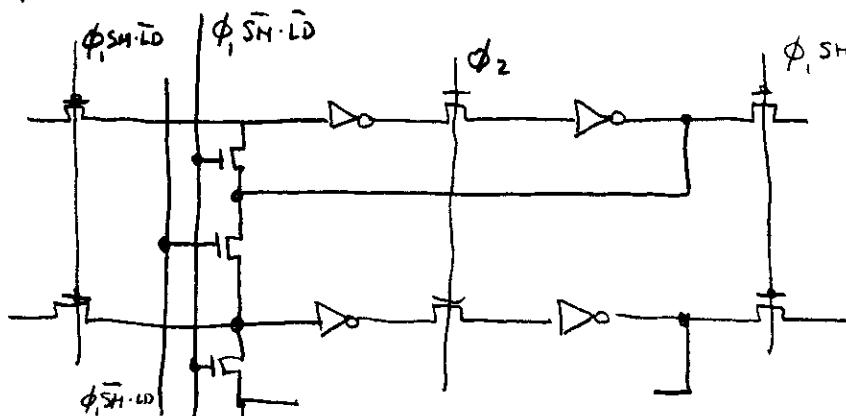
WRAP-AROUND EACH SR, BRINGING THE TWO ENDS TOGETHER. OF COURSE THIS STARTS TO KLUGE-UP THE SWAPPING IN FSM, BECAUSE NOW THE LOAD CIRCUITRY IS THERE ALSO:



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

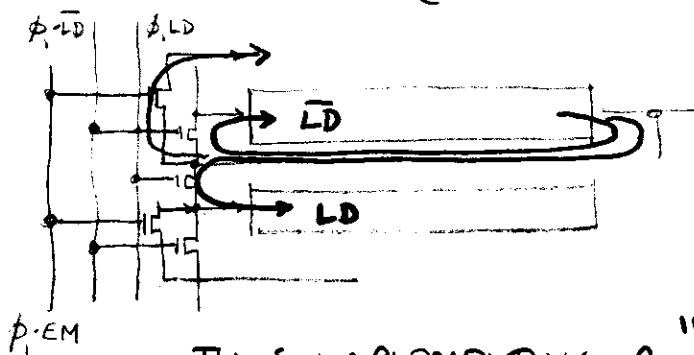
[MAYBE USE 8:1 MIN  
PULDOWN SR  
HOWEVER]

- TO EXPLORE TENTATIVE STICK DIAGRAMS: STICK DIAGRAM ON A GRID IT'S A GOOD WAY TO GET A FEELING FOR DENSITY. (LET RED OR GREEN RUN RIGHT BY BLUE, etc.).
- 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  
KLUGES UP  
THE SR ARRAY]

- NOTE ALSO THAT IF WE ADD ANOTHER PATH AT THE "LEFT" (CONCEPTUALLY) AT 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 A WHILE!



## 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 Diagrams, Alg., Stck Dsg of subsections.

21 NOV: Tentative Layout

7 DEC: Project Report

> I'll make selection for inclusion by 28<sup>th</sup> 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.  
Will favor: Those completing early, well checked.  
Interesting design or appl. concepts  
Well subsetted for first testing

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

- 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 Chaos net interface. Like ~ 4 or 5 students to participate. This could be a very exciting project --- but also might be a hard one. Tom Knight & Clark Holloway of the AI Lab will be able to meet with a group next week to describe few, give handout on current MSI revision. 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  $P_n(A) = \frac{(NA)^n}{n!} e^{-NA}$

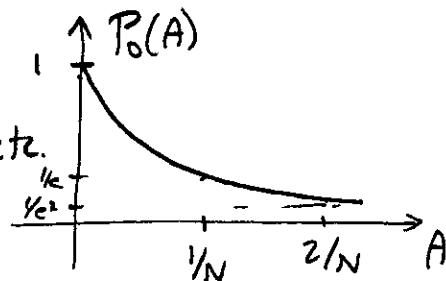
Data Analysis for Sci & Eng  
Meyer, Wiley Publ.  
'75

In particular,  $P_0(A) = e^{-NA} = \text{prob. b. i. 3 of zero defects in an area } A.$

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

Chips with  $A \approx \frac{1}{N}$

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  $\sim 5\text{mm} \times 5\text{mm}$ , yield will be  $\sim 10\%$  or so.  
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 is, 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 tosses some random errors into your code!  
That's what we have to deal with in integrated systems.

- Maybe some clues about how to approach those 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 PENETRY 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 Y, 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 BEG. ACCESS **E-BEAM**

Also, the industry doesn't take risks. It's always working on next year's real product, and the processor 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 LET'S 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

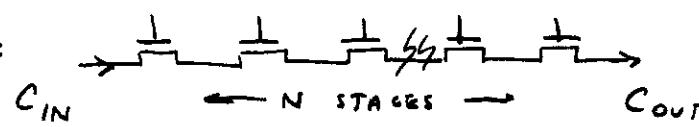
AH! BUT DON'T GET TOO CARRIED AWAY: BE CAREFUL: YOU'VE GOT TO GET LEVERAGE:

EXAMPLE: At a module yield  $\sim 1/2$ , for area A, if double A to get self-test + connection net, then yield  $\rightarrow \sim 1/6$ .

So if start with 100, get only 50 (@ 2A per) of which only  $\frac{1}{6}$  th now work

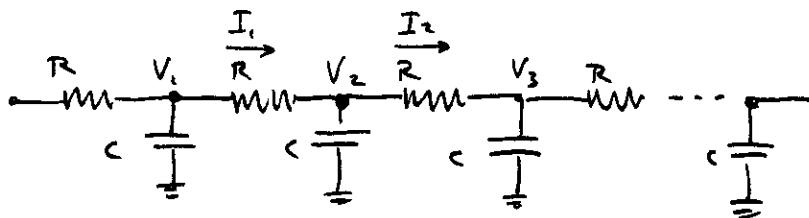
$\sim 8$  equiv chips. [Play with the numbers to see]

## 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:



$\left[ \min R = \text{res of one pass-trans}, \min C = \min C_g. \text{ Parasitics} \right]$   
 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  $R \approx C$  per unit length, then this reduces to differential form:

$$RC \frac{dV_2}{dt} = \frac{\Delta V_{1-2}}{\Delta x} - \frac{\Delta V_{2-3}}{\Delta x} = \frac{\Delta^2 V}{\Delta x^2} \quad \left\{ \begin{array}{l} \text{charge in } \Delta V \\ \text{for change in } x \end{array} \right.$$

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

where:  $R = \text{res/length}$   
 $C = \text{cap/length}$

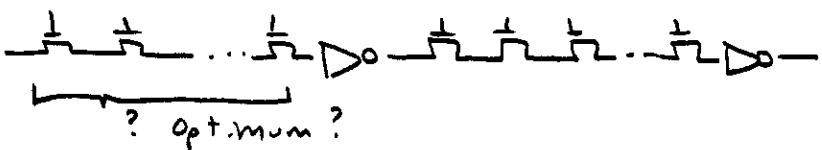
Slide

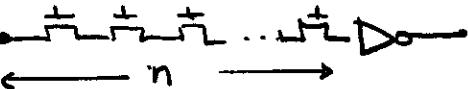
- This is the well known Diffusion Equation.  
Its solution are complex, but in general the time required for a transient to propagate in such a system is proportional to  $X^2$ .
- This can be seen qualitatively: Doubling N doubles both R & C in the system, multiplying an inherent time constant by 4.
- One extra pass transistor: Adds little delay to a small chain  
But may add a lot of delay to an already long 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  $\approx RCn^2 + T_{inv}$

- AVERAGE DELAY / STAGE  $\cdot RCn + T_{inv}/n = f(n)$

we find that minimizing  $f(n)$ , the min occurs when:

$RCn^2 \sim T_{inv}$

so that's how to choose  $n$

- Right now:  $T_{min}n^2 \sim T_{inv} = \frac{9}{2} \cdot 2 T_{min}$ ,  $\therefore n^2 \sim 9$   
and  $n \sim 3$ . We'll usually use  $n \sim 4$

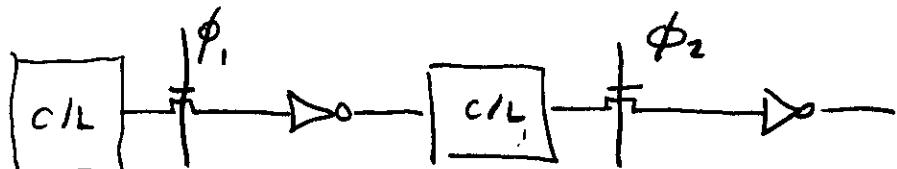
- IN CLOCKED STAGES: IT'S AN ARCHIT. QUESTION. CAN USE UP TO THE MAX DELAY ON A ⚡ WITHOUT PENALTY

## TRANSIT TIMES & CLOCK PERIODS

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

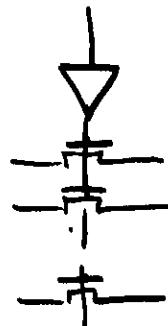
Consider:

R-R  
Transfer  
stage



- Normally the C/L + clock gate: IN PASS TRANSISTORS.
- Level Restoring will be inverters:  $k \sim 8$ . Max delay  $\sim 8T$
- Normally THESE will be MATCHED:  
C/L + clock:  $\sim 8T$ , INV. MAX  $\sim 8T$
- Thus TOTAL OF  $\sim 16T$  per phase  $\times 2$  for STRAYS.
- Thus  $\sim 30T$  per clock phase.

> BUT CTL Lines typ. 2.0ly drive  
10 to 30 Pass Transistors.



• Even if SUPER BUFFERS, if  
YN30, Driver delay is  $\sim 9T$

$$\therefore T \sim 100T$$

• Must add  $\sim 8T$  to operate drivers

>  $\therefore$  Total CLK  $\phi \sim 50T$ .

In 1978:  $0.3 < T < 1.0 \text{ ns}$  so  $T \sim 30 \text{ to } 100 \text{ ns}$ .  
(For compact synch. d. syst)

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

For example in OMZ,  $\phi$  is  $50T$  but  $\phi_2$  has a delay  
of  $100T$  for each 4-bit ALU block, so  $\phi_2 \sim 400T$ .

$$\therefore T(\text{OMZ}) \sim 450T \quad \text{or} \quad (135 \text{ ns to } 450 \text{ ns} = \text{fclk period})$$

## (\*) 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  $\sim \frac{1}{2}$  to  $\frac{2}{3}$  will get done / look ok. to go on chip set.

So probably:  $\sim 12 - 15$  projects will go on chip.

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

Area  $\sim 60\text{ mm}^2$ . Probably  $\sim 10\text{ mm}^2$  will go for  
scribe lines, alignment marks, test patterns, etc.  
Maybe a bit more for "packing inefficiency".

So: maybe  $45\text{ mm}^2 / 15 \text{ projects} = 3\text{ mm}^2 / \text{project}$ .

So: A large project will be  $2 \times 2\text{ mm} = 4\text{ mm}^2$ .

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

Try for  $2\text{ mm}^2$

Let me know early if you'll want  
to go over  $4\text{ mm}^2$

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

: Priorities?  
C/F 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

WE'LL CONSIDER  
AT LEASTTO THE COMING CHANGES: 3 areas:

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

- OVERVIEW/REVIEW OF DES & IMP  
"ARTIFACT FLOW"

SLIDES

- Now, HOW ARE WE ACTUALLY GOING TO DO ALL THIS?
- DRAW FLOWCHART ON BOARD \*  
MANY OF THE ANSWERS, IN DETAIL, ARE IN THE GUIDEBOOK.

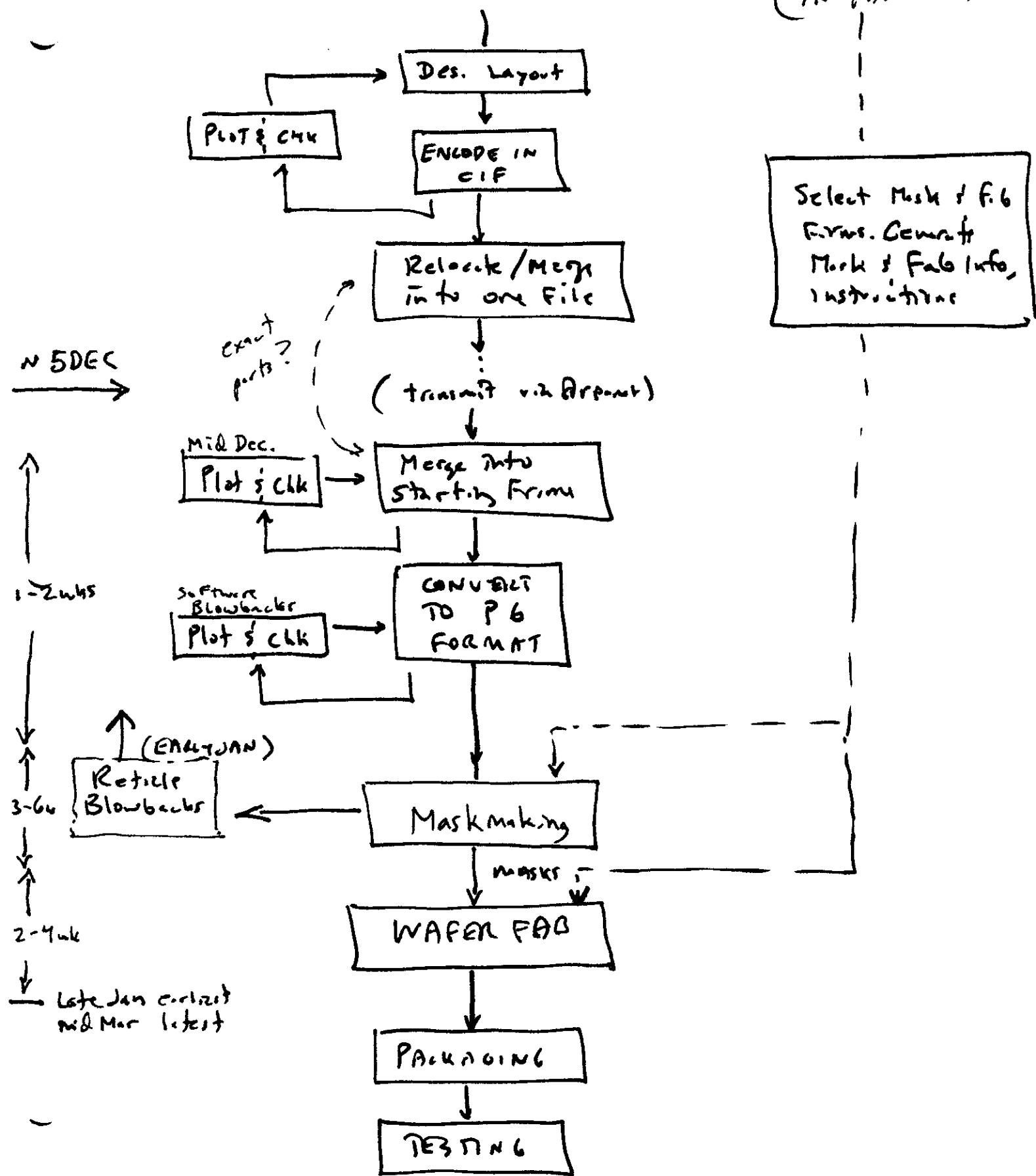
(UNIV. IMPL.)

MIT. PROJ. CHIPSET

IMPLEMENTATION:



(IN PARALLEL)



- 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 rather 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 ADHOC

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 firm sticks as few rules, R.T. CHECK

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

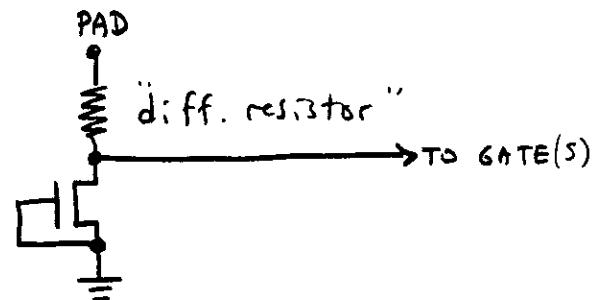
(p100-144) Bob Baldwin, Dick Lyon

- A bit about creating a Cell Library. 100-101
- Table of Contents 102-103
- SOME USEFUL CELLS: Pads, PLA cells, Log.2/Arithmet2, & STARTING FRAME (disc. later)
- PADS: Input w lightning Arrestor, Output w 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 through, current will flow dropping the overvoltage 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.

Trade-off max speed ( $\times e$ ) for simplicity, less area, lower power than the DM drivers/pads.

- PADS: SIZE:  $126\mu m$  (114 if oversize). BIG enough to be easy to Bond.

## ELECTRICAL SIMULATION: Another Useful Design Aid.

P 19-25 DICK LYON

- Very common use of computers in traditional structured 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-TRI, NODE WITH HIGHEST FANOUT, CONTROL DRIVERS, OUTPUT DRIVEN.
- TWO WIDELY USED SIMULATORS: MIDL (SIAMF.) SPICE (BECKMANN) Tend to suffer from Univ. Batch "card oriented" culture of origin. 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 2nsec RISE/FALL + inverter condition external]
- > OUTPUT PRODUCED IN LINE PRINTER FORM. SLIDE [certainly Node 1 to node 6 delay ~ 11nsec @  $T = 27\text{ nsec}$  [optimistic] But helps 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 some integrated design system, operating off same or machine generated variants of same data base, with att. p.t.d to user interface. BUT TRADE-OFFS VS IMPL TIME
- Const. of such int. sys. important requirement NOT 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 Maskmaking.

Most Mask houses use GCA-Mann PG & Photoreject Equip.  
The photorejectors yield a limit to field exposable  $\approx 1\text{cm}^2$

Thus, we normally make ~~at~~  $10\times$  reticles, and at most cover a  $1\times 1\text{cm}$  area on the wafer with these.

Now the package we use is std. I 40 pin - which will hold  $\approx 6\text{mm} \times 6\text{mm}$  chip. So if we  $1\text{cm} \times 1\text{cm}$ , must be able to scribe within it. (more later).

We must provide artifacts for:

- > alignment marks used in fab sequence
- > CD's used in maskmaking (lines of known widths)
- > Scribe lines
- > maybe "fiducials" used in photorejecting
- > maybe "parity marks" used in photorejecting

We must also provide information regarding

plate polarities, dep. on whether n+ or p+ resist  
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 (self-aligned) "Channel Stop" region is ion-implanted (p+) 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 -

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 well use conducting epoxy to glue the chips into their package. Then connect ground lead to package ground.

- OK: 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.
- (CH4 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 > pitch 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 MASKS & 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  $\approx$  6k for masks, (mostly PG time), and  $\approx$  2k for FAB of  $\approx$  20 wafers.  $\approx$  20 is a minimum run. If not more, less /water.
- We'll get enough chips so that every participant can 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 QUESTIONS: DESFILE - CHIPS IN  $\approx$  3 days.

[ONE OF THE VERY IMPORTANT CHANGES COMING IS THE GENERAL AVAILABILITY OF EAST INKAOUND 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 GUIDEBOOK

- 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  $2\lambda$
- 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: ~ 2 K to fab ~ 10 to 20.

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

Specs: Sample SPEC sheet for proj. set 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 frame to allow us to measure  $N$  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  $N$ , 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.

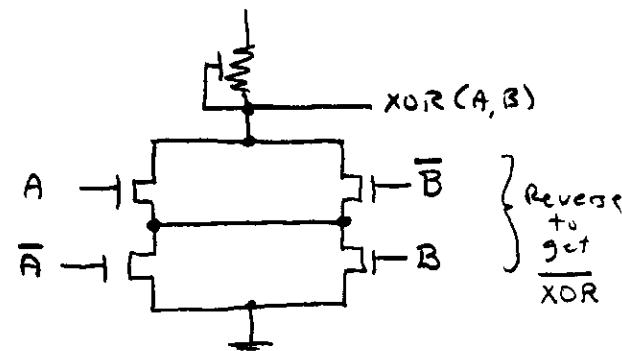
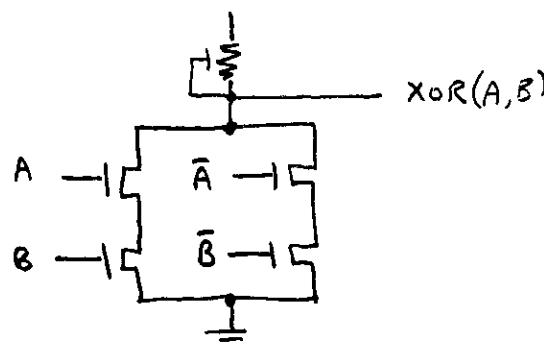
ANOTHER IMPORTANT GATE EASILY IMPL. IN MOS IS THE XOR GATE "EXCLUSIVE-OR"



		(XOR)	( $\overline{XOR}$ )
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

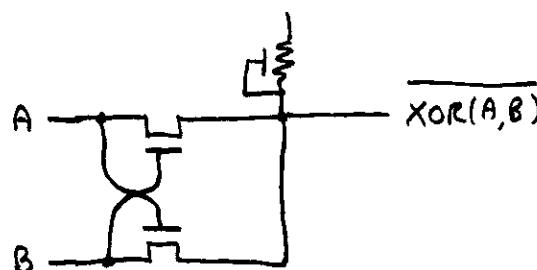
XOR  $\Leftrightarrow$  "MOD 2 SUM"

AS IN MULTI COMPARATOR:



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

IS:



Describe fcn:

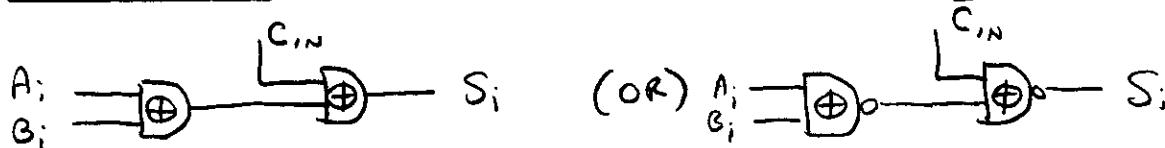
If  $A = B = 1$ , OUT = 1

$A = B = 0$ , 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. Portability of 2 chips of System: DATA PATH, CONT. **SLIDES**
- In later lectures I'll be describing this system in some more detail.
- It is architecturally 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. It's like today to review the basic ideas of the stored program computer --- the classical general purpose computer sometimes referred to as the von Neumann machine (as a concept).
- What is a general purpose, St. prog. Computer? [First 1/2 CH 6]

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 1/2 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 set of outputs indep of act. in Data Path. Could be used for impl. a digital filter - data in at left, fixed set of ops, data out at right.

- Enhance this a b.t: **Fig 2**

To make control sequencing possibly, be fan 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 FSMs, 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 PRUG. MACH.

**FIG 4 a**

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 poss. bilities:

- 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, DATA PATH, MEM) we can mechanize any of a number of different algorithms of interest to a number of diff users.

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

Programs operate on data also located in the memory.

- The Machine func as follows

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

> In Partiz FSM state, called Fetch N<sub>x</sub> 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 starty state" for op type
- ③ Fetch Operands      state seq. to fetch opernds ...
- 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 mult.

- 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, Decade, etc.

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

[While many states, transitions are usually simple 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 Definable when 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 writeable 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 flag b.t.s, h next state lines, & i-6,7 instructions, thus need  $2^{(i+f+h)}$  words.  
But can usually easily reduce by inserting logic in feedback path:

- In figure **Fig 7**, some logic will call the "Micro program counter P-th" is inserted in the path between the <sup>bus</sup>memory and decoder.

This type of control is generally referred to as MICRO PROGRAMMED Control, whether wr.able or Read only memory 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  $\mu$ -CODE MEMORY.  
ITS PURPOSE: TO REDUCE THE AMOUNT OF  $\mu$ -CODE MEMORY REQ'D.  
DOES THIS BY:
  - MAPPING THE  $F + N$  bits of state into smaller #, then decoded to address  $\mu$ -code memory.
  - Reduces  $N$  by allowing complex funs within  $\mu$ PC to be specified with just a few bits of control info.
- THE CONTROLLER CHIP DESC. IN CHG IS THE  $\mu$ PC PATH PORTION OF A MICRO-PROG. CONTROLLER FOR OMEZ.  
(GO BACK TO OV. SLIDE)
- ALTERNATIVE WAY TO VIEW FIG 7 :
  - > EXAMINE LOOP FORMED BY  $\mu$ PC, Decoder,  $\mu$ -CODE MEM
  - > VIEW  $\mu$ -code mem address as an "Inst ADDRESS" and wires from  $\mu$ -code mem to  $\mu$ PC 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 fun capab. lity 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 FETR, EXEC, STORE, etc.



6.978 SEMINAR

TUESDAY OCT 31

- TODAY: Douglas Fairbairn, XEROX PARC.

Seminar: Interactive Graphics Aids for Integrated System Design

- ④ Next Time: Midterm Exam: Bring your colored pens/pencils, Books, Notes, Scratch Paper

- ④ Projects: Returning Project Assgn #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 ~ 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 project. Actually, that helps provide the CONTEXT for smaller designs.
- > The List may help you identify possible collaborators.

- ④ TODAY I'M GOING TO NAME DOUG FAIR. OF XEROX PARC JOIN US TO PRESENT A SEMINAR ON — .

DOUG IS A MEM. LCT STAFF AT PARC, AND PRES. IS ACT. MGR OF THE LSI SYSTEMS AREA.

HIS RES. INTERESTS RANGE FROM COMPUTER ARCH - ESP OF PERSONAL / DISTRIBUTED COMPUTERS - TO INTEGRATED SYSTEMS ARCHITECTURE & DES. METHODOLOGY.

DOUG WAS ONE OF THE FOUNDERS OF XEROX'S LSI SRS. 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	T E A M	C O L L A B	S I P E	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	WRITABLE 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 #14NOVEMBER 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 NEWS RE PROJ. SET IMPL.

- NOTE: ON NOV 21 (TUES) DICK LYON OF XEROX PARK  
~~RE~~ WILL PRES A SEMINAR ON VLSI  
IMPL. OF SPEECH PROCESSING FUNCTIONS.

HE WILL PRES. SOME BASIC BUILDING BLOCKS ADDRS, MULTIPLIERS, MEMORY SUBSYSTEMS, OUT OF WHICH ONE CAN BUILD <sup>DIG.</sup> SIGNAL PROC. SYSTEMS. HE WILL THEN DISCUSS AN EXAMPLE PROJECT CHIP NOW IN DESIGN: A ~30 CT ~~ALL~~ DIGITAL BANDPASS FILTER BANK WHICH WILL FIT ON ONE CHIP EVEN IN PRO. TECHNOLOGY.

HE WILL GIVE A PRES. OF THE UV. SYS. DES OF AN ISOLATED UTTERANCE RECOGNITION SYS. WHICH SHOULD FIT ON ONE OR A FEW CHIP WITHIN A FEW YEARS. (FOR VOICE INPUT TO COMPUTER)

- NOTE: CALVIN 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, MOSE BRENN)
- 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:

Problem 1 (a)  $C_L = 0.8 \text{ pF}$

I GRADED FAIRLY HARD. A FEW WHO HAVE DONE VERY WELL ON HW DID POORLY. MAYBE HAD A BAD DAY. DON'T BE DISCOURAGED. OUR HOME LNS DID...

$$C_g(\min) = 4 \times 10^{-4} \text{ pF} \times (6 \times 6) \text{ mm}^2 = .0144 \text{ pF}$$

$$Y = \frac{C_L}{C_g} = \frac{0.8}{0.0144} = 55.6$$

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  $\times 2$  inv. per cell (4 per cell pair)

- However, only half can be on at one time maximum.
- First of each pair is approx 5  $\mu\text{s}$  long when "on"
- Second is "  $6\frac{1}{2}$   $\mu\text{s}$  "

- So worst case is when all "1st" inverters "on".

$$\text{In this case, } I_{\text{total}} = 256 \times \frac{5}{5 \times 10^{-4}} = 25.6 \text{ mA}$$

- Wires can carry max  $1 \text{ mA/mm}^2$  But are 1mm thick, so:

$$\therefore \text{WIRE} \geq 25.6 \mu\text{m wide}$$

-3

Grading: Calc errors -3 to -4; Invalid Assump: -4 to -6; Forgot pull-down R

Problem 2: Most people did O.K. Most PLA's matched the log.2 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  $1\frac{1}{2}\lambda$  from neighboring enh. mode transistor gate region **SLIDE**

Some said metal not over cuts, b.t look closely, O.K.

(b) Several Major Errors:

**SLIDE**

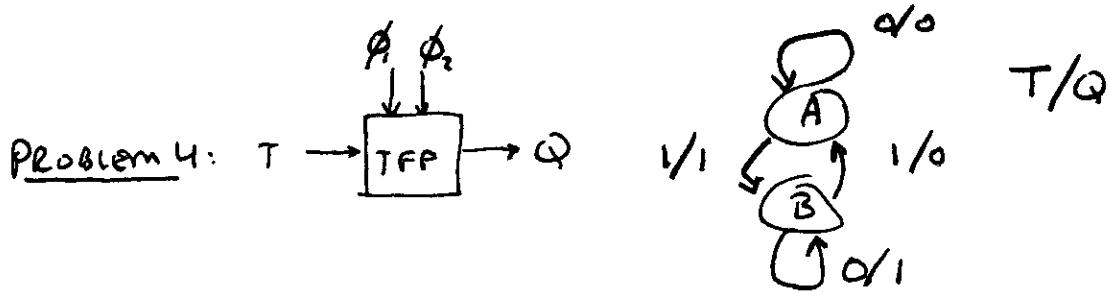
- > Ratio should be 8:1, not 4:1.
- > No implant over pullup to make dep. mode.
- > Metal over Bitting contact is  $2\lambda$  not  $3\lambda$  from GND METAL.
- > Some DIFFS ARE  $2\lambda$  Sep. not  $3\lambda$ .

(c) EXAMPLES OF REAL IMPROV. IN DESC. LAYOUTS OVER WHAT CIF CAN DO:

- > DIG AT CORNERS OF BOXES / NOT CTC
- > DIRECT ITERATION (AS IN INFORM. DESC. LANG IN CHAR)
- > PASS PARAMETERS TO CALLED SYMBOL: USE FOR EXAMPLE: SCALING
- > USE #'S OTHER THAN OUR INTEGERS, ALLOWING EASY SPEC OF SIZES IN  $\lambda$ .
- > CALL SYMBOLS BY NAME INST OF #.

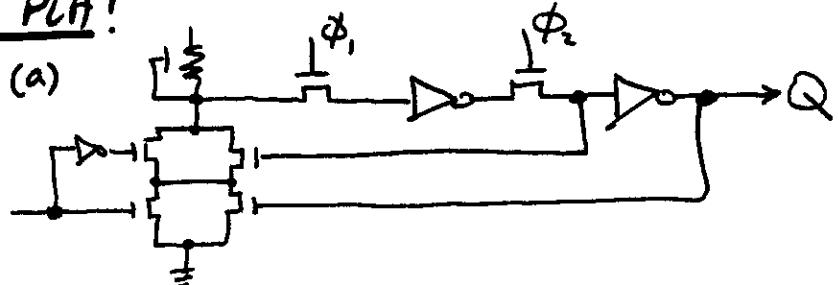
, etc.

(4)

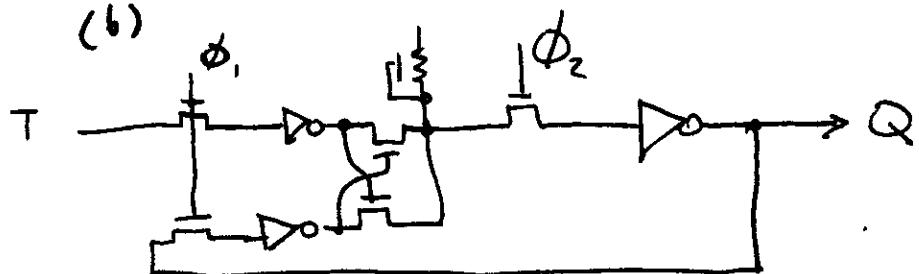


MANY SOLNS: A few used PLA!

- Some based on XOR:

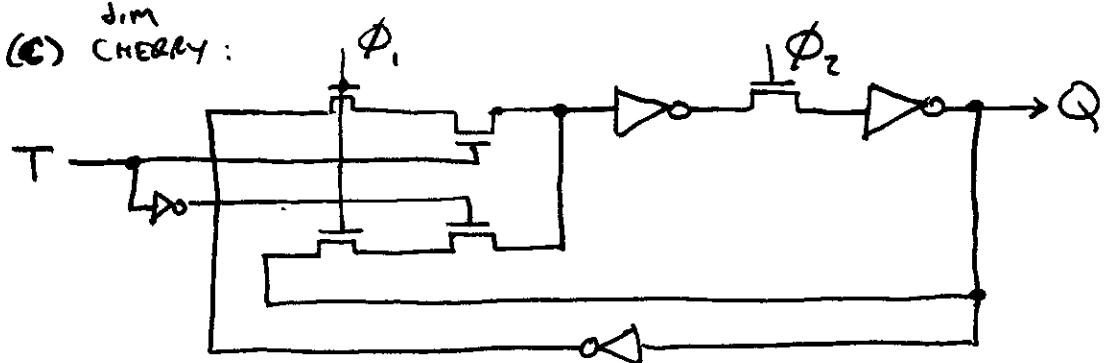


(b)

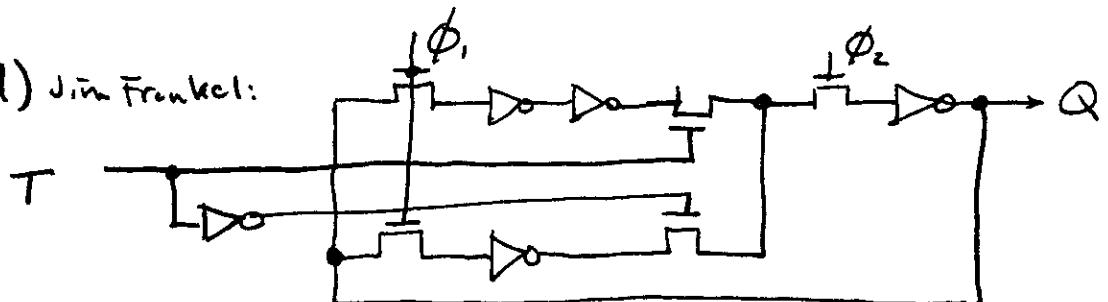


- Others: (c) Jim Cherry:

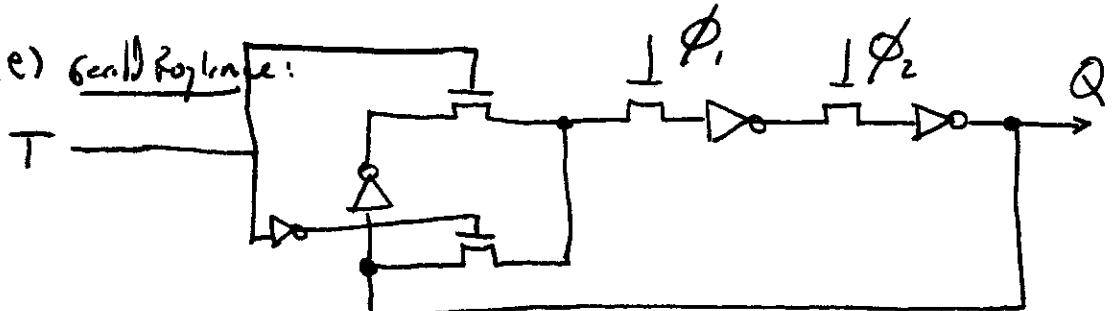
Based on Directly  
Selecting  
inverted or  
non-inverted  
feedback  
Based on  
value of T



(d) Jim Frankel:



(e) Gerald Royston:

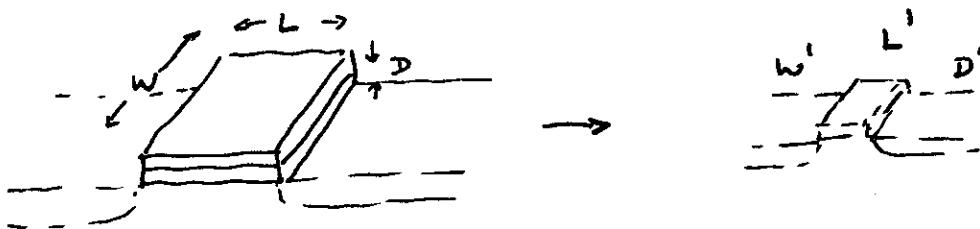


## WHAT HAPPENS AS WE MAKE THINGS SMALLER?

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  $\alpha$ . WHAT WILL BE EFFECTS ON ELECTRICAL PROP OF DEVICES?



- Distances divided by  $\alpha$ :  $W' = \frac{W}{\alpha}$ ;  $L' = \frac{L}{\alpha}$ ;  $D' = \frac{D}{\alpha}$
- Also, to keep all Electric Fields same as before, we reduce VDD and set all thresholds to be reduced by same factor  $\alpha$ .
- 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  $\mu\text{m}$  feature sizes ( $\lambda = 4\text{e}\mu\text{m}$ ).
- Now what happens to  $T$ , to  $C_g$ , to  $I_{ds}$ , etc?

- Remember Eqn 1(CH.1):  $T = L^2 / \mu V_{ds}$

$$\therefore \frac{T'}{T} = \frac{(L/\alpha)^2}{(V/\alpha) \cdot L^2/\alpha} = \frac{1/\alpha^2}{1/\alpha} = 1/\alpha$$

$$\therefore T' = T/\alpha$$

Transit time of m.h devices, i.e. circuit speeds all scale down linearly with  $\alpha$

- Gate Capacitance:  $C_g = \epsilon WL/D$

$$\frac{C'_g}{C_g} = \frac{(L/\alpha) \cdot (W/\alpha)}{(D/\alpha)} \cdot \frac{1}{WL/D} = 1/\alpha$$

$$C'_g = \frac{C_g}{\alpha}$$

- Note, however, that Capacitances in general scale up by  $\alpha$  if given in terms of  $C/mm^2$ . Since for given absolute  $W \in L$ ,  $D$  gets thinner.
- Note: Resistances in general scale up by  $\alpha$  since lines get thinner vertically. However, the  $R/D$  of transistors remains  $\sim$  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)  $I_{ds} = \frac{M \epsilon W}{LD} (V_{gs} - V_{th})(V_{ds})$

$$\text{so } I_{ds} \propto WV^2/LD$$

$$\frac{I}{T} = \frac{WV^2/\alpha^3}{LD/\alpha^2} = \frac{1}{WV^2/LD} \propto \frac{1}{\alpha}$$

$$I_{ds} = \frac{I_{ds}}{\alpha}$$

- So, current per device goes down by  $\frac{1}{\alpha}$
- However, # devices per unit area goes up by  $\alpha^2$
- $\therefore$  Current Density over the chip (if uniform design or repeated designs, etc) goes up linearly with  $\alpha$

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  $1mA/\mu m^2$  current limit.

One solution: Increase H/W of wires by  $\alpha^2$  as scale down. Some of this can be done, but might not 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  $\alpha^2$ .  
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 ~~without 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  $T \propto T'$ .

Thus,  $P_{sw} \propto \frac{WV^3}{DL}$ ,  $\boxed{P'_{sw} = \frac{P_{sw}}{\alpha^2}}$  per device

So both dc & sw power remain const. 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  $\log_2$  pulldowns were turned on.

[ $P_{sw}$  diss. 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  $\gg$  larger than wafer thickness  
 (i.e. macroscopic dimensions), the following rough  
 rules of thumb apply:

$< 1 \text{ watt/cm}^2$	no problem at all
$2 \text{ watt/cm}^2$	} somewhere in here, you begin to
$4 \text{ watt/cm}^2$	} need special heat sinking
$> 8 \text{ watt/cm}^2$	begin to need forced cooling of some kind to remove heat.

> As in all thermo problems, all depends on next higher  
 context. One isolated chip in well heat sinks  
 package may do well in still air at  $4 \text{ w/cm}^2$ , 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  $> 2 \text{ watt/cm}^2$  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  
 some 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.

$$\begin{aligned}\text{CELL AREA} &= 19\lambda \times 21\lambda = 57\mu\text{m} \times 63\mu\text{m} = 57 \times 10^{-4}\text{cm} \times 63 \times 10^{-4}\text{cm} \\ &= 3591 \times 10^{-8}\text{cm}^2 = \underline{\underline{3.6 \times 10^{-5}\text{cm}^2}}\end{aligned}$$

$$\text{Power} = I \cdot V = \frac{5V}{60k\Omega} \cdot 5V = \underline{\underline{4.2 \times 10^{-4}\text{Watts}}}$$

(ACTUALLY THE  $10k\Omega$  is FOR FAST PROCESS; IS OPTIMISTIC)

$$\text{PWR/AREA} = \frac{4.2 \times 10^{-4}}{3.6 \times 10^{-5}} = 12\text{W/cm}^2. \text{ BUT IN SERIES ONLY } 1/2 \text{ ON AT A TIME}$$

$$\therefore \text{PWR/AREA} \approx 6\text{W/cm}^2$$

SO IF YOU FILLED A CHIP WITH THESE, YOU'D GET IN TROUBLE!

- AS AN aside, ONLY  $\approx 2K$  of these ~~and~~ in pairs would fit on a  $\approx 5\text{mm} \times 5\text{mm}$  chip.

So how do we get 48K, 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 DE3 METH IN TEXT, WON'T HAVE PROBLEMS. REGIONS OF WIRES AND PASS GATES DON'T DISS. DC POWER, i.e. COMPL. FOR REGIONS OF DENSER ~~same cells~~. LEVEL REST. CELLS.

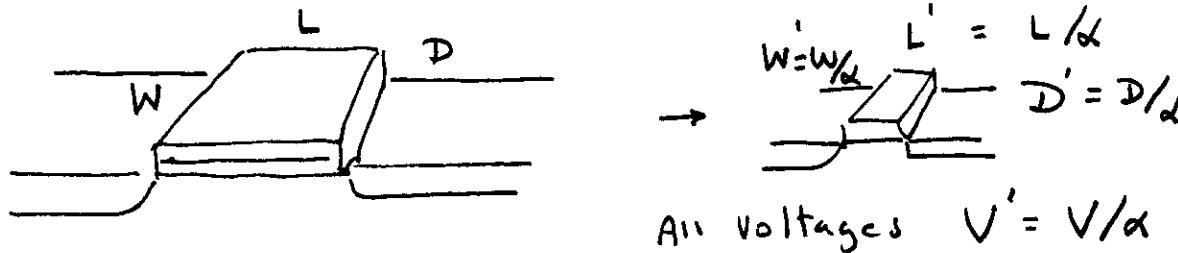
BUT, IF MAKE REALLY BIG ARRAYS OF SR's, STACK CELLS, ETC. CALCULATE PWR DENS. TO BE SAFE.

- IN ANY EVENT: SCALING DOES NOT MAKE THIS WORSE



LECTURE # 15NOVEMBER 9

- Collect Proj. Assign. # 2
- LAB / PROJ: Use only Boxes at right  $\times$ 's. Later software won't support more.  
Also: We have priority in Lab > 3 P.M.  
POINT OUT # COSTS: ONCE DO RIGHT, SAME  
FOR LATER COPIES AT A  $\frac{1}{\alpha}$  price.
- TODAY: CONTINUE SCALING.  
DISCUSS LIMITING FACTORS.
- SUMMARIZE SCALING SO FAR:



Using this simple scaling by  $\alpha$  of all dimensions including vertical, and all voltages ( $V_{DD}$ ,  $V_m$ ,  $V_{DSS}$ ) we found:

$$T' = T/\alpha \quad ; \quad I_{ds}' = I_{ds}/\alpha \quad ; \quad \text{PROBLEM ENCOUNTERED!}$$

[ex.: Eq 1:  $r = \tau^2 / \mu V_{DS} \therefore \downarrow$ ]

But # of devices per unit area goes up by  $\alpha^2$   
So mean current density: CURRENT INTO AREA OF CHIP  
Goes up by  $\alpha$ . This is not problem, since in our scaling wires would get thinner. Current limited wires would either have to have aspect ratios increase by  $\alpha^2$  (which can't go on for long) or get proportionally wider.

$$C_g' = C_g/\alpha \quad ; \quad \text{But } \underline{\text{capacitances}} \text{ in general scale up by } \alpha \text{ if given in terms of } C/\mu m^2 \text{ since vertical dimensions shrinking (oxide getting thinner).}$$

Resistance/scale up by  $\alpha$  since lines getting thinner.

(Except note that we can't really do that with curr. lim. metil.)

However, if calculate it out, find that  $R/D$  of FETS will stay about the same.

So another problem:  $R/D$  of poly, diff getting proportionally larger while  $R/D$  FET staying same. This is aggravated by crystal clumping in POLY --- makes effect worse as scale down.

DC Power Dissipation: Per Device  $P_{dc} = I \cdot V$

$$P'_{dc} = \frac{P_{dc}}{\alpha^2} ; \# \text{Dev. / Area goes up as } \alpha^2$$

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 <sup>2</sup>	no prob.
2 W/cm <sup>2</sup>	begin to need
4 W/cm <sup>2</sup>	
> 8 W/cm <sup>2</sup>	reasonable heat sinking need way to remove heat: forced cooling

We examined ≈ worst case in our methodology:

Larg array of shift registers are in Fig 8b ch 4.

Found power/unit area ≈ 10 W/cm<sup>2</sup>.

BUT NOTED THAT WE USE PASS-T LOGIC BETWEEN THESE AND USUALLY DON'T USE SUCH SHORT BUCKLES, etc.

So normally we don't need to worry. But should calculate if in doubt. TRY TO STAY < 2 W/cm<sup>2</sup> IF CAN.  
(over line areas to further)

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:

$P_{sw} = \text{energy stored on capacitance divided by the clock period or time between successive chargings / discharging.}$

$$\text{But } T' \propto T. \text{ Thus, } P_{sw} \propto \frac{CV^2}{T} ; T \propto \frac{L^2}{V}$$

$$\therefore P_{sw}' \propto \frac{WL}{D} \cdot V^2 \cdot \frac{V}{L^2} = \frac{WV^3}{DL} \quad \therefore$$

$$P_{sw}' = \frac{P_{sw}}{\alpha^2}$$

So: Since  $P_{sw}$  per device goes down by  $\alpha^2$ , and # dev./area goes up by  $\alpha^2$ , 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 binding it into the design, or sometimes can be controlled dynamically. This reflects an important metric of device performance: SWITCHING ENERGY per DEVICE

$E_{sw} = \text{power consumed by device at max clock freq, multiplied by the delay: i.e., it is a "power} \times \text{delay" product.}$

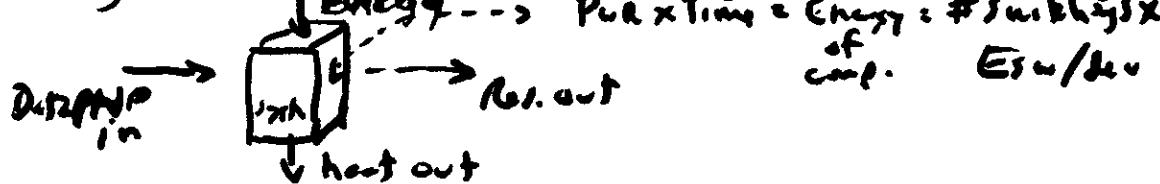
Rationale: In a sense, to do any computation, we must switch a large collection of switches, switching them in a particular order, and some number of times.

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 those limit ultimate performance:

No matter how much power we put in, we can't reduce delays below the minimum value of  $T$ .

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.  $\text{Min. Power} \times \text{Time} = \text{Energy} = \# \text{switches} \times \text{Energy}$



## How Does Switching Energy Scale?

Our basic FET switches have  $E_{SW} \propto CV^2$

and ∵

$$E'_{SW} = \frac{E_{SW}}{\alpha^3}$$

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 1/100 th as many chips)
- > Power Density remains constant.
- > All voltages reduced by factor of 10
- > Current / Area increased by factor of 10  
(current)
- > Time delay / stage decreased by factor of 10
- > Power-Delay Product decreased by " " 1000  
of Devices

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/ D rel to FET)

Forgetting possible design mechanics, with problems:  
 CAN WE SIMPLY SCALE DOWN WHOLE DESIGN? Yes but:  
 CONSIDER

(6)

- Delays to outside world: (Read SPACE vs TIME in CH1)

What is effect of scaling on output driver design; delays?  
 We can't just scale them down: the outside world stays  
 b.g. Remember the result in chap 1:

Min Delay when use factor of  $e$ , and  $N = \ln Y = \ln \frac{C_L}{C_{g\min}}$

$$\text{In this case: Min Tot Delay} \approx T e \ln \left[ \frac{C_L}{C_{g\min}} \right]$$

Now, scale everything down by  $\alpha$ , including Voltages.  
 (This we do scale even in the external world)

$$T' = T/\alpha; C_{g'} = C_g/\alpha; \therefore Y' = \alpha Y$$

DERIVE

$$\therefore t'_{\min} = t_{\min} \cdot \frac{1}{\alpha} \left[ 1 + \frac{\ln \alpha}{\ln Y} \right]$$

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.)

$$\text{Derive *: } t_{\min} = T e \ln Y$$

$$t'_{\min} = T' e \ln Y' = \frac{T}{\alpha} e \ln(\alpha \cdot Y) = \frac{T}{\alpha} e [\ln \alpha + \ln Y]$$

$$t'_{\min} = \frac{T e \ln Y}{\alpha} \left[ 1 + \frac{\ln \alpha}{\ln Y} \right] = \frac{t_{\min}}{\alpha} \left[ 1 + \frac{\ln \alpha}{\ln Y} \right]$$

ALSO, BRIEFLY MENTION BIPOLAR TTL, I<sup>2</sup>L,  
OTHER MOS:CMOS

(7)

SO, SCALING PRODUCES SOME GREAT EFFECTS.  $\uparrow$

$\downarrow$  CONT.

BUT WE SHOULD ASK: HOW SMALL CAN WE MAKE NMOS THESE DEVICES AND STILL HAVE THEM WORK?

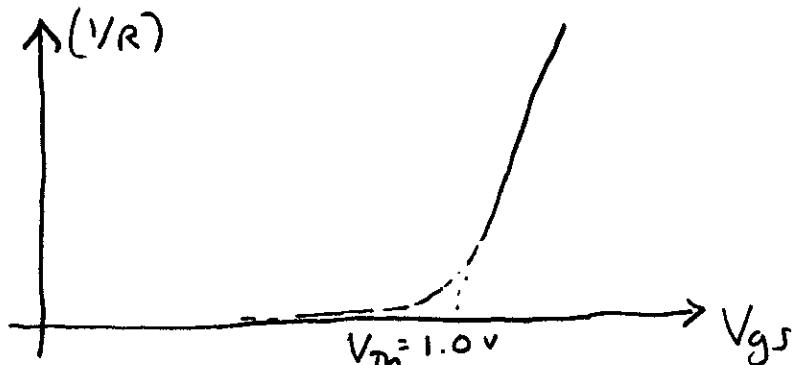
WE MUST SUSPECT THAT THERMAL, STATISTICAL, QUANTUM EFFECTS ARE ULTIMATELY GOING TO MEAN THINGS UP!

QUESTION: IF PATT'S, FAB WERE NOT LIMITING US, HOW SMALL COULD WE MAKE FET'S 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<sub>GS</sub>, IT IS NOT SIMPLY A STRAIGHT LINE RUNNING DOWN TO V<sub>TH</sub>, BUT HAS AN EXPONENTIAL TAIL:



Below V<sub>m</sub>, the conductance  $1/R$  is not zero but depends on V<sub>GS</sub> and temperature:

$$\frac{1}{R} \propto e^{(V_{GS} - V_m)/(kT/q)}$$

T = absolute temp

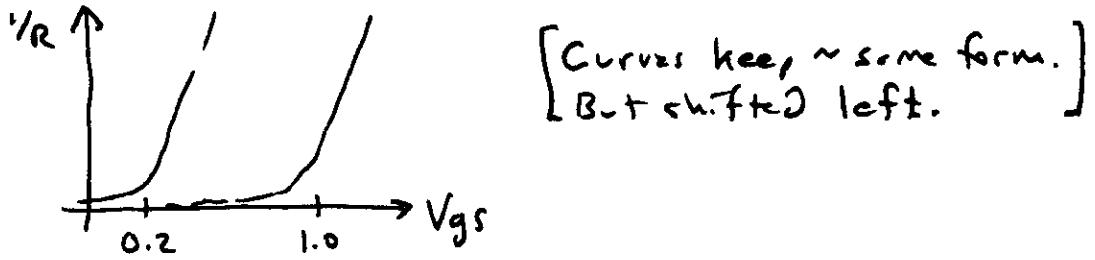
k = Boltzmann constant

q = charge on electron

At room temperature,  $\frac{kT}{q} \approx 0.025$  volts

Thus, at present threshold voltages, an off device, below threshold by perhaps 0.5 volts, is below threshold by  $20kT/q$ . 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, 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:



Now the OFF FET is down ONLY BY  $4kT/Q$ ,  
 $\therefore$  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, 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 voltage, say when VDD reaches about 1V. But this also causes us power dissipation problems.  
**EVEN NOW** voltage scaling should be done to int. nos PERC.  
Could reduce prob by Reducing T. See interesting <sup>say</sup> paper by Gaenslen, Riedout, Walker CRL where very small MOSFETs were op. at liquid N temp and measurements confirm improvements.

But the TRL says nos at 5°

"existing 25  
ratios of  $\frac{1}{2} kT$ "

## ALL ENERGIES CAN BE SCALED AS $\propto kT$

- Ah, related to low temp operation: Side point:  
 - So → Reducing temperature also reduces  $E_{SW}$ . Figures quoted were at room temperature. But be careful!

CH 9 shows interesting comparison: FET's vs J.s.-J.J.s:

Won't go into full detail, read if you are interested:

Although switching energy at Device T 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  $T_{switching}$ .

Now, A COUPLE 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 FET's down to  $\sim 1/2 \mu m$ , gaining the  $\alpha^3$  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 required for a computation, must use T at the heat sink temperature. Refrig. dev. to reduce energy of computation is  $\propto \log_2$  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  $\approx$  room temp.). J-J has advantage of trading the power-delay trade off to lower values of delay ( $\sim 1/30$  best FET's). J-J's at quantum limits at  $\sim 1\mu m$  sizes. can't be scaled down smaller. ---

So the factor of 100,000 quoted by IBM CS is of switching energy is wiped out by  $\times 1000$  (possible) improvmt in FET's by scaling, and  $\times 100$  due to the perpetual motion machine error.

MAYBE DON'T OVERDO IT

## JUST MENTION THESE BRIEFLY:

(10)

### 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  $\alpha$ .

Results from granularity; statistical distribution of substrate impurity charges which determine the threshold voltages.

At same time, # devices increasing. If pull-up threshold goes one way, and pull-down 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 (not all FETs being within threshold limits) = 0.9, may require  $V_{DD} \approx 0.7\text{V}$ .

- Quantum Effects. Gate oxide is already only  $1000\text{ \AA} = 0.1\mu\text{m}$  thick. Positional uncertainty for electrons is related to uncertainty in momentum by

$$\Delta p \Delta x \approx \hbar$$

For energy barrier of  $\sim 1\text{eV}$ , calculating corresponding  $\Delta p$ , we find  $\Delta x$  is about  $.001\mu\text{m}$ . 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:

	<u>1978</u>	<u>MID-80'S</u>	<u>19XX</u>
MIN FEAT. SIZE ( $2\lambda$ ):	$6 \mu m$	$1 \mu m$	$0.3 \mu m$
$T$ :	0.3 to 1.0 ns	$\sim 0.05$ to 0.15 ns	$\sim 0.02$ 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:	$\sim 30$ to 100 ns	$\sim 5$ to 15 ns	$\sim 2$ to 4 ns
CLK. PERIOD ( $\sim 100T$ )			

- > The mid 80's column we'll probably reach without major hassles. Voltage will be scaled to 1/2 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 50N 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 COURSE, WE ARE STILL LEFT WITH THE PROBLEMS:  
WAVELENGTHS  $1.4$  to  $0.7 \mu m$  MV  $\approx 3 \mu m$   
OF LIGHT! HOW DO WE MAKE SYSTEMS THIS SMALL?



6.978 LECTURE #16.NOVEMBER 14.

TODAY: MORE ABOUT THE FUTURE: HOW THE PATT & FAB TECHNOLOGIES MAY BE IMPROVED TO ACHIEVE SCALING TO LIM. DIM.

FIRST: PROJECTS: NOTE: NO OFF. HRS. TMW. 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 by Friday early):  
Rae McClellan, Dave Levitt, Dave Shover, Scott Westbrook, Moshe Bain, Martin Freeman.  
(@locksmotor)
- 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 FET, no OVG will be placed. You will therefore <sup>not</sup> be able to probe any large ( $> 25\mu m \times 25\mu m$ ) 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 <sup>Functional</sup> AND gates and <sup>Functional</sup> inverters!
- A word about 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 CLOCK. However, in normal room light, this time may be reduced to tens of micro seconds. Avoid light when long clock periods are used. Dyn. Mem. Chips packaged in 814 degree packages because of this effect.

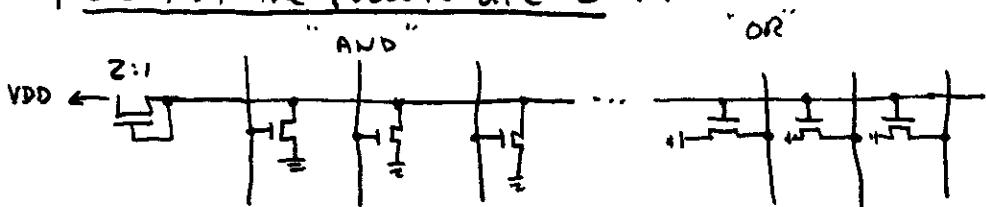
- "2 weeks to get the preliminary report of your lab submission. I would like to check IF you want to get your final chip back."

## 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 Mirricker 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 makes 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 pull-ups 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's area. We use a  $4\lambda \times 4\lambda$  DIFF box to place a transistor rather than a  $2\lambda \times 4\lambda$  box. Sketch on SLIDE

Thinking  
→

- FINAL: I had originally planned to give a Take Home Final Exam. However, I now think that such a final would be antithetical 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 <sup>more</sup> weight than the M.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 i.e. 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  $\sim 0.4$  to 0.7  $\mu\text{m}$ . UV is  $\approx 0.2$  to 0.3  $\mu\text{m}$ .

- LET'S EXPAND OUR VIEW OF THE FUTURE FROM THE SIMPLE TABLE GIVEN LAST TIME OF SIZE / PERF. AS FUNCTION TO INCLUDE HOW THESE CHANGES CAN BE BROUGHT ABOUT.  
NOTE: ALTHOUGH AS SYSTEM DESIGNERS WE WON'T NORMALLY 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 GRAN FILM EACH YEAR ---

AS WE PROCEED FROM 6  $\mu\text{m}$  to 0.3  $\mu\text{m}$  Feature Sizes,

- |   |                            |
|---|----------------------------|
| ① CIRCUIT & DEVICE TECHNOLOGY<br>& DESIGN RULES & CONSTRAINTS<br>② PROCESSING TECHNOLOGIES<br>③ PATTERNING TECHNOLOGIES | } WILL ALL BE<br>CHANGING. |
|---|----------------------------|

LET'S 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 size 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<sup>nMOS</sup> 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 plan of subsystems and compiling stuck bypassed cell arrays from building blocks into target design rules - with aids helping check for current / power density limits, etc.

So, as first 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":

- (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 dosage, 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 track and flipped into position in a tool, 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 thinner 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.

- LET'S SKETCH THE COMING EVOLUTION IN P&I TECH BY BEGINNING WITH WHAT'S 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  $2\lambda$  (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.

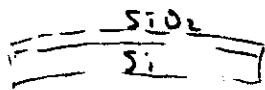
Such projection exposure will be good down to  $\approx 1\mu\text{m}$  feature sizes. Below that we'll need an ALT IN UV light. But more than feature size to worry about.  $\lambda$  also result of  $1\mu\text{m}$

- BUT EVEN AT  $\approx 2\mu\text{m}$  Feature Size ( $\lambda = 1\mu\text{m}$ ) 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  $\text{SiO}_2$  is grown. Thermal Coeff of exp of  $\text{SiO}_2$  is  $\approx 1/10$  that of Si:

As wafer cools, Si shrinks more than  $\text{SiO}_2$ . So, wafer won't be flat, but will be convex on the  $\text{SiO}_2$  side



Now if cooled slowly, may be possible to relieve the stress induced by the diff in ~~length~~ 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 we'll 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. Steps, repeat with local alignment on the wafer to cover entire wafer.

When  $\lambda < 0.5\mu\text{m}$ , Feature Size  $< 1.0\mu\text{m}$ , we no longer can use UV light (wavelength  $\approx 0.3\mu\text{m}$ ), 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 ~~as small as 250 Å~~, but there are problems which will likely limit use of Direct writing with E-Beams to feature sizes  $> \sim 0.5\mu\text{m}$ . 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  $250\text{\AA}$ , 10KeV, resist  $0.4\mu\text{m}$ .  
AT SPACINGS OF  $2\text{nm}$ ,  $1\text{nm}$ ,  $0.5\text{nm}$ ,  $0.3\text{nm}$ .

- > Associated Problem: Exposure at any point depends on exp. at neighboring points. This proximity effect requires pattern dependent exposure corrections at small values of  $\lambda$ .
- > As  $\lambda$  decreases, the time to "write" the whole wafer rapidly increases. Time (exp.) per wafer gets very large at small  $\lambda$ .

- However, although time / w-far 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  $\approx 0.5 \mu\text{m}$ , and how do we get high wafer throughput in manufacturing at  $< 0.5 \mu\text{m}$  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'S 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  $\approx 0.02 \mu\text{m}$ . X-RAYS OF  $\approx 100$  to  $1000$  eV range (Wavelengths  $\approx 0.001$  to  $0.001 \mu\text{m}$ )  
 [SO, WE CAN USE THE STEP & ALIGN TECHNIQUE TO ULTIMATELY SMALL DIMENSIONS USING OPTICAL INTERFEROMETRIC ALIGNMENT AND X-RAY EXPOSURE.]

- WE ARE THUS BACK TO MAKING MASKS:

(see 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 X-RAYS OCCURS. INTERACTIONS OF X-RAYS WITH MATTER TEND TO BE ISOLATED, LOCAL EVENTS. ANY ELECTRONS PRODUCED WHEN X-RAY 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 SHOWS PAD WITH PERIOD OF  $\approx 0.3\text{ }\mu\text{m}$ .

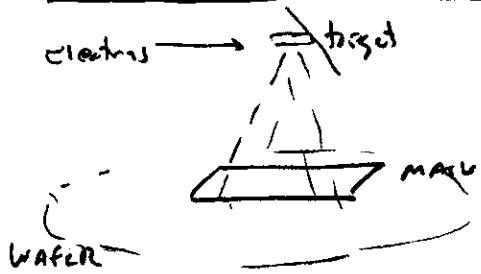
SLIDE

- 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 put close to source to get higher intensity, resolution is poor. If furthermore, exposure takes 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 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 one chip could be exposed at each vertex every few seconds. Achievable values of  $\lambda$  in both the feature size and alignment sense are down to  $\approx \lambda = 0.1\text{ }\mu\text{m}$ .

- 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  $\approx 1-2 \mu\text{m}$  Feature size  
AND PROJ. EXP + STEP 1 ALIGN ON WAFER WILL GET AROUND ANY RUNOUT PROBLEMS.

- THIS ALSO ELIMINATES STEP 1, REP. IN MARKMAKING. COULD HAVE REASONABLE TURNAROUND IF OPTIMIZED FOR THAT (TREND TO MAKING OPT MASKS BY E-BEAM: SIMPLER, QUICHER, 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  $\lambda \approx 0.3 \mu\text{m}$  Features 0.5 to 0.6  $\mu\text{m}$

- CENTER PATH: POSSIBLY THE ULTIMATE MANUFACTURING PATH FOR DEMEST SYSTEMS. CLEARLY WORKABLE FROM PATTERN FEATURE SIZE 1; ALIGNMENT STANDPOINT DOWN TO  $\lambda \approx 0.1 \mu\text{m}$ . (FUTSIX 0.8  $\mu\text{m}$ ) 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. envirn at  $\lambda = 0.5 \mu\text{m}$ , then manufacturing full systems at  $\lambda = 0.2 \mu\text{m}$ .



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<sup>+</sup> TO ~4<sup>+</sup>.
- ALSO, OFFICE HOURS TMRW FROM ~11 TO ~4.
  - HOW MANY MIGHT BE INT. IN USING LAB OVER THANKSGIVING?  
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 ? SO THEY CAN GET DINNER. ]

MEMORY CELLS & SUBSYSTEMS

- Those presently get lots of att'n because of the huge market and money being made manufacturing memory chips.
- Presently, general purpose computing done by Von-Neumann type machines, where memory is made distinct from processing in the hardware  $\boxed{\text{CPU}} \leftrightarrow \boxed{\text{Mem}}$ .

So much effort is made to increase speed ( $T_p$ ) of CPU's, 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 competitive 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  $\text{MEM} \rightarrow \text{PROC}$ .

(C) Because of the # of best LSI designers preoccupied with Memory cell design, it is a CLASSICAL ENGINEERING ART.

All those 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 & power dissipation will require the use of electrical simulation.
- (e) The ultimate I-T memory cell which will show requires very clever(experienced) circuit design in its interface and support circuitry.

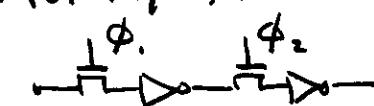
- So this is a diff. design environment than that discussed in this course. Rather than doing lots of designs per unit time, designing hierarchically to build big systems, and being design methodology constraints to keep out of trouble, MEMORY DESIGN involves highly optimized (lower level) process and circuit design.
- Note also: Changes in way we <sup>might</sup> do computing will be discussed during weeks of Dec 4-8:  
DEC 5 Carri Mead (H. Con. Pres.) DEC 7 Wayne W. Liner (<sup>Rec.</sup> machine)  
DEC 8 Carlo Sequin, U.C.B., X-TREE [How many can come?]
- To Prepare for That Week: Read CH 8 p1-9, p31-32, p57; Skim p33-56.  
(notes, results)
- Also: Make a copy of, and at least skim thru the recent highly important paper by John Backus of IBM - "Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs"

- 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 SRcells: 

$$\text{Area} \approx (21\lambda \cdot 2) \cdot 19\lambda = 126 \mu\text{m} \times 57 \mu\text{m}$$

"78λ = 3 μm  
use all metal layers"

- How many Bits could we place in a big chip, say

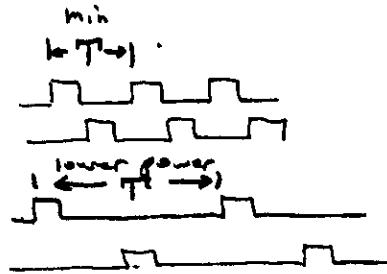
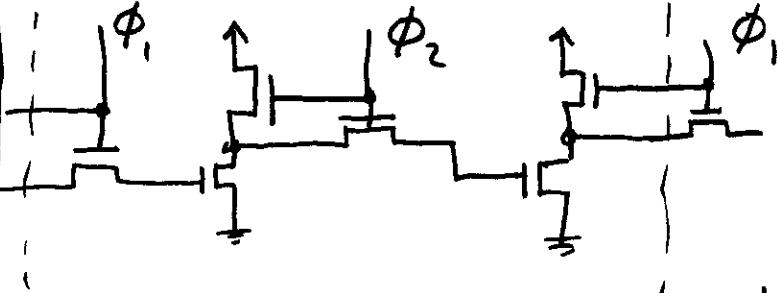
$$\# \text{BITS} \approx \frac{4000^2}{126 \cdot 57} \approx 2K \text{ bits}$$

4 mm × 4 mm ?  
also use for  
large transistors

- Remember we previously calculated a power dissipation of  $\sim 8$  to  $12 \text{ W/cm}^2$ . This was rather conservative (high) and also could be reduced by using longer pullups, narrower pullups to perhaps  $3$  to  $6 \text{ W/cm}^2$ . 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 larger 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)

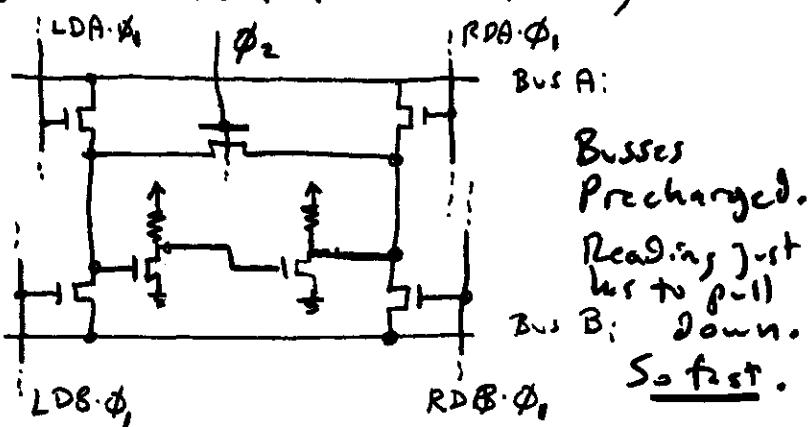
# (SEE THE TI Book FOR MORE SR TRICKS!)

- 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's; Then progress thru series of denser RAM cell designs

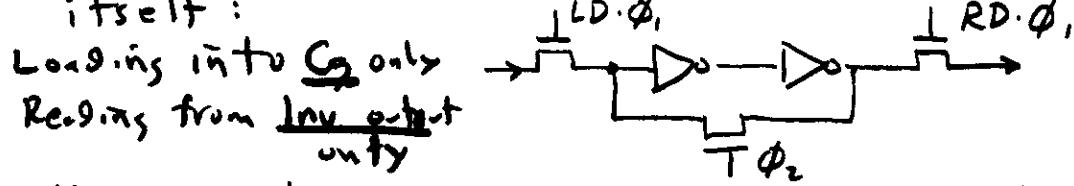
## OM REGISTER CELL:

IF 2-BUSES: 9-T's / BIT  
IF 1-Bus 7-T's / BIT

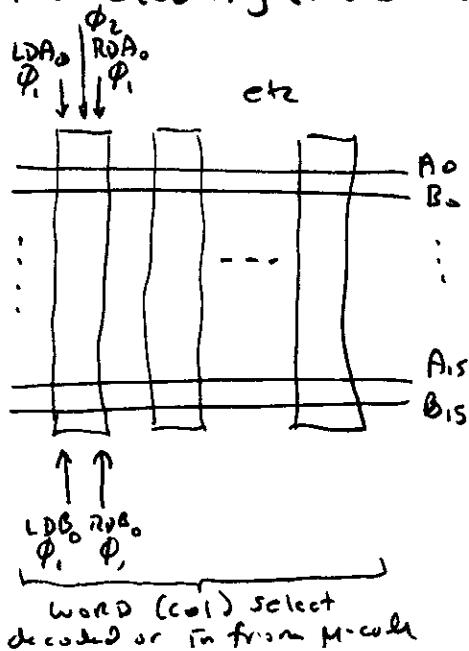
## NOT STATIC



Basically very simple. Really like SR which feeds back on itself:



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  $\sim 54 \times 48 \lambda$

$$\therefore \text{In } 4 \times 4 \text{ mm: } \frac{(4000)^2}{162 \cdot 144} \approx 700 \text{ Bits}$$

EVEN LESS THAN SR

Even if used one Bus (7-T's/cell)  
could only get  $\sim 1K$   
in 4x4mm.

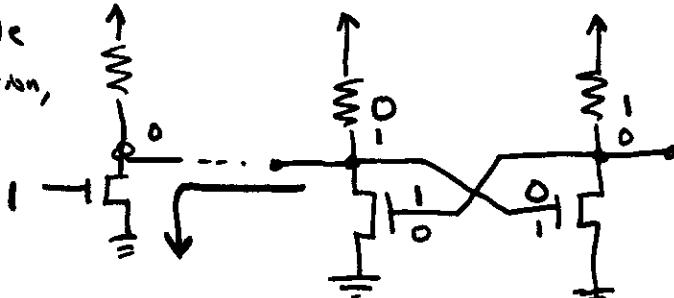
NO Problem with PWR.  
BUT MUST BE CLOCKED

## HOW TO MAKE A STATIC RAM

THE SIMPLEST STATIC RAM CELL IS BASED ON CROSS-COUPLED INVERTERS:

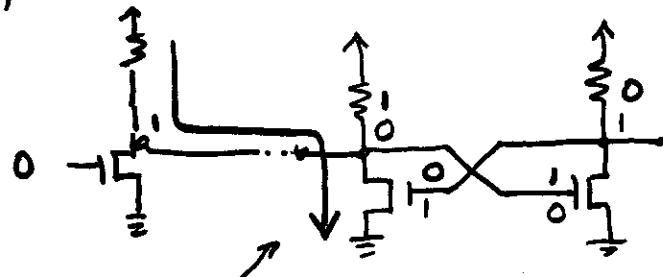
Same as our Reg cell, but no clocked pass-T in feedback path:

- If either input/output node is pulled down, by ext. action, other side turns off, latching the pulled down side ON.

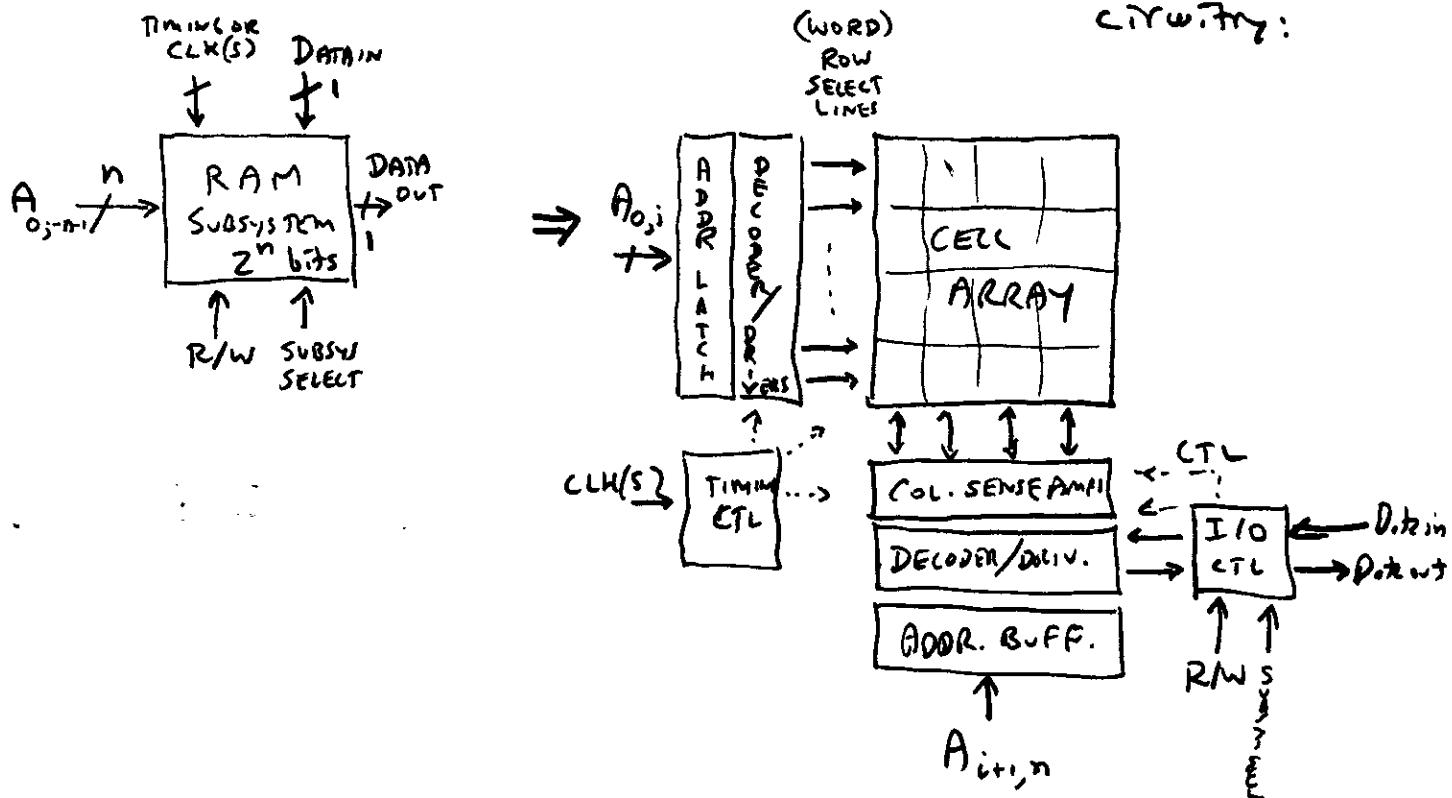


- Will hold state indefinitely (STATIC) unless power goes off.

- Note: Ratios have to be right: external driver has to have wider pull-up in order to source enough current to raise input node above threshold. Usually use double rail input



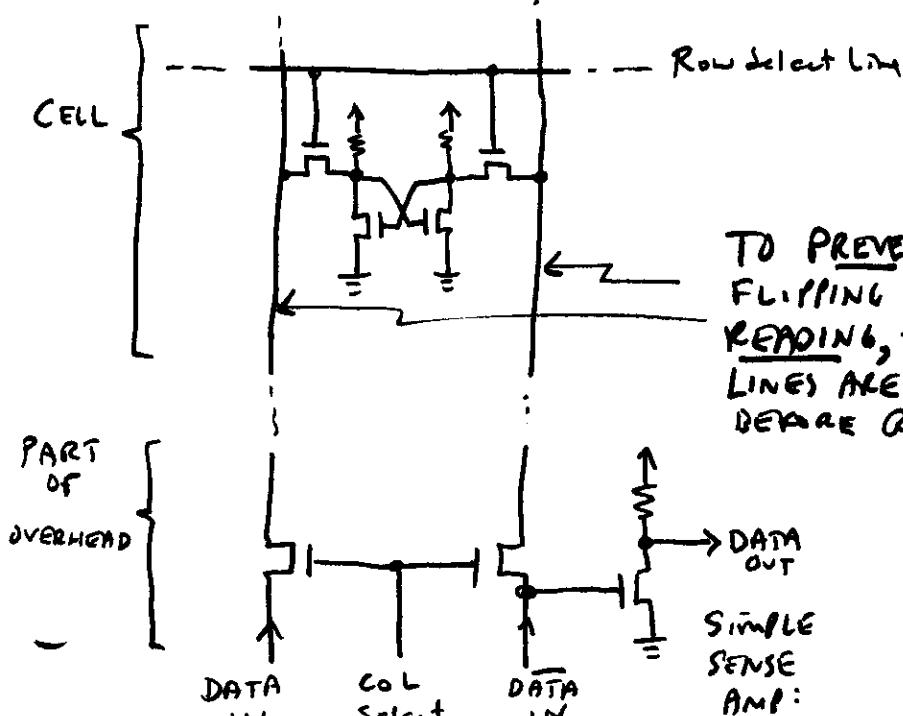
IN GENERAL: MAKING RAM SUBSYSTEMS: There is overhead circuitry:



## 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  $n \dots$  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 - Price to interface.

### THE 6-T STATIC RAM:



TO PREVENT  
FLIPPING WHILE  
READING, THESE  
LINES ARE PRECHARGED  
BEFORE READING

SIMPLE  
SENSE  
AMP:  
JUST AN  
INVERTER

Area: Maybe  $125\text{ }\mu\text{m} \times 125\text{ }\mu\text{m}$   
So:  $\frac{(4000)^2}{125 \cdot 125} = (32)^2 = 1\text{ K}$   
per  $4\text{ mm} \times 4\text{ mm}$   
Not counting OVERHEAD

i.e. ~ same as  
single bus REG CELL.

If on-chip subsystem,  
COULD READ/WRITE  
WHOLE WORD AT ONCE

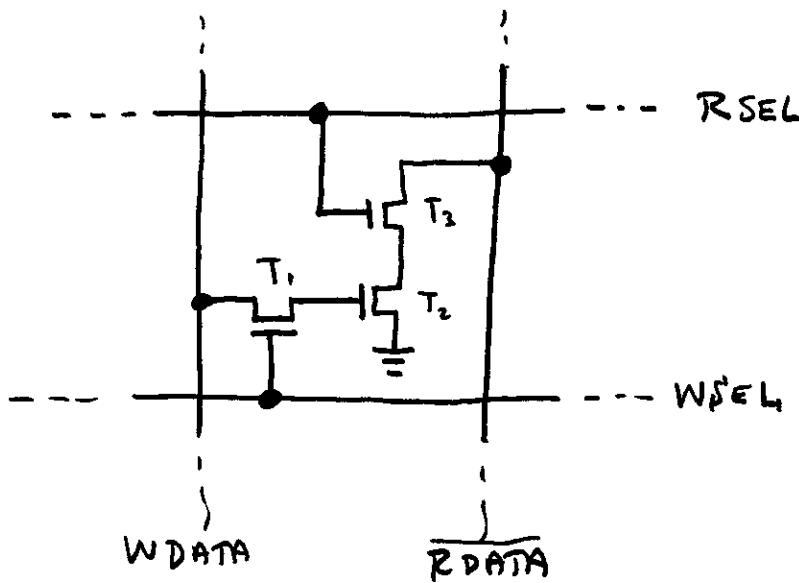
NO PWR/AREA PROBLEMS  
SINCE NOT DENSE ENOUGH!  
(IN BITS/AREA)

- 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  $\propto$  #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-Wires per cell.
- Data stored on gate of T<sub>2</sub>
- To WRITE: Put data in onto WDATA. Raise WSEL for a while, then lower.

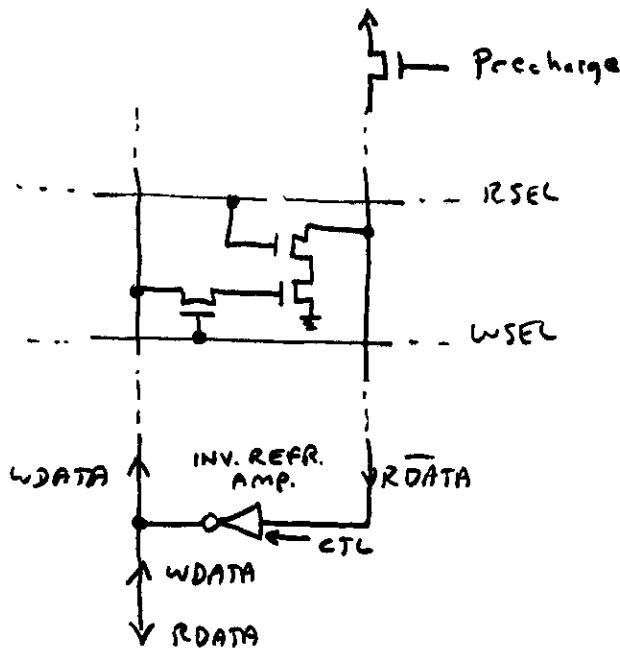
- TO READ:  $\Rightarrow$  RDATA is Precharged high.

$\Rightarrow$  RSEL is then turned on. The RDATA line is then discharged only if input to T<sub>2</sub> is high.

- $\Rightarrow$  Note: Output is complement of input
- $\Rightarrow$  Note: The Read is non-destructive
- $\Rightarrow$  Note: The cell must be periodically refreshed.

## 3-T RAM (cont.)

- Let's 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 STICKS, FINALLY MANY LAYOUT ALTERNATIVES.  
(NOTE ADJ COL. OR ROWS CAN SHARE A GND RETURN)
- DENSITY:

Conservatively:  $60\mu\text{m} \times 60\mu\text{m}$  per cell. So, # in  $4\text{mm} \times 4\text{mm}$ , (not counting overhead)

$$\text{IS: } \frac{(4000)^2}{(60)^2} \approx \boxed{4K \text{ bits}}$$

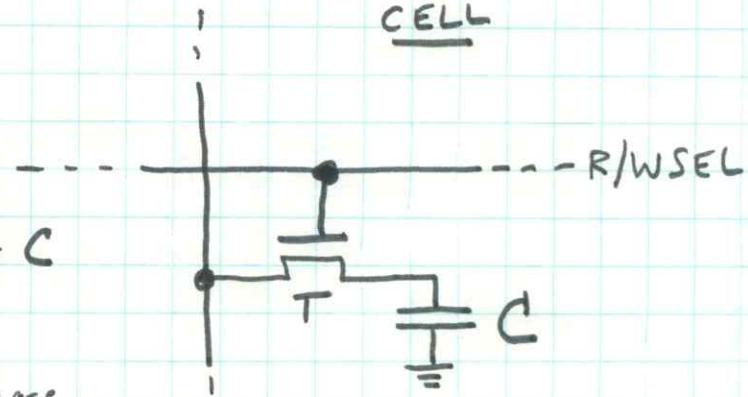
- PWR: No problem: switching power, and short dissipation thru  $T_2 - T_3$  if energy stored on RDATa.

4 Times as Dense as STATIC RAM.  $\approx$  SAME PWR/PERA.

SUPPOSE THIS IS STILL NOT DENSE ENOUGH!

THE ULTIMATE (AT LEAST NOW) IS THE SO-CALLED  
L-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 <sup>sense</sup> amp inputs) have capacitance  $C_L$ , then when close T to r/wd, the voltage on C divides:

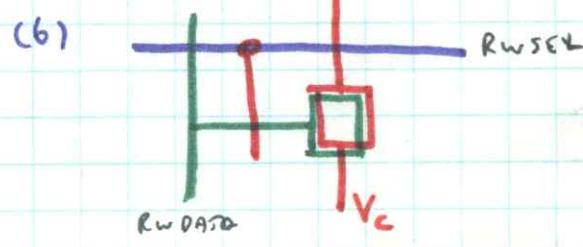
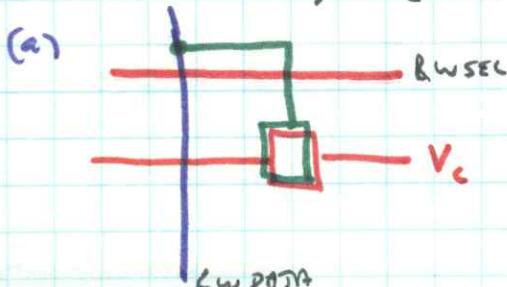
$$\text{Ex: If } C_L \text{ low, } C \text{ holds } V_{DD}, \text{ then a } \perp \text{ on } C_L \approx V_{DD} \left( \frac{C}{C+C_L} \right)$$

- In Real RAMs  $C_L \gg C$ . Maybe by  $\times 20$ ,  $\times 50$ ,  $\times 100$ ! So this gets really hairy!

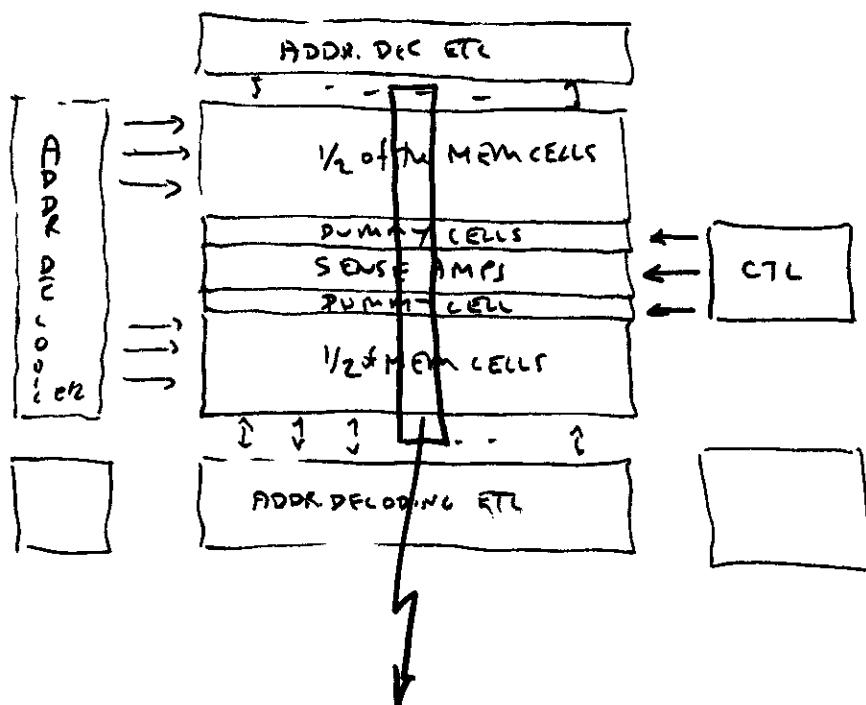
> Need special form of sense amplif. circuity.

> Also, Readout is DESTRUCTIVE, & MUST Rewrite after every read, not just to refresh.

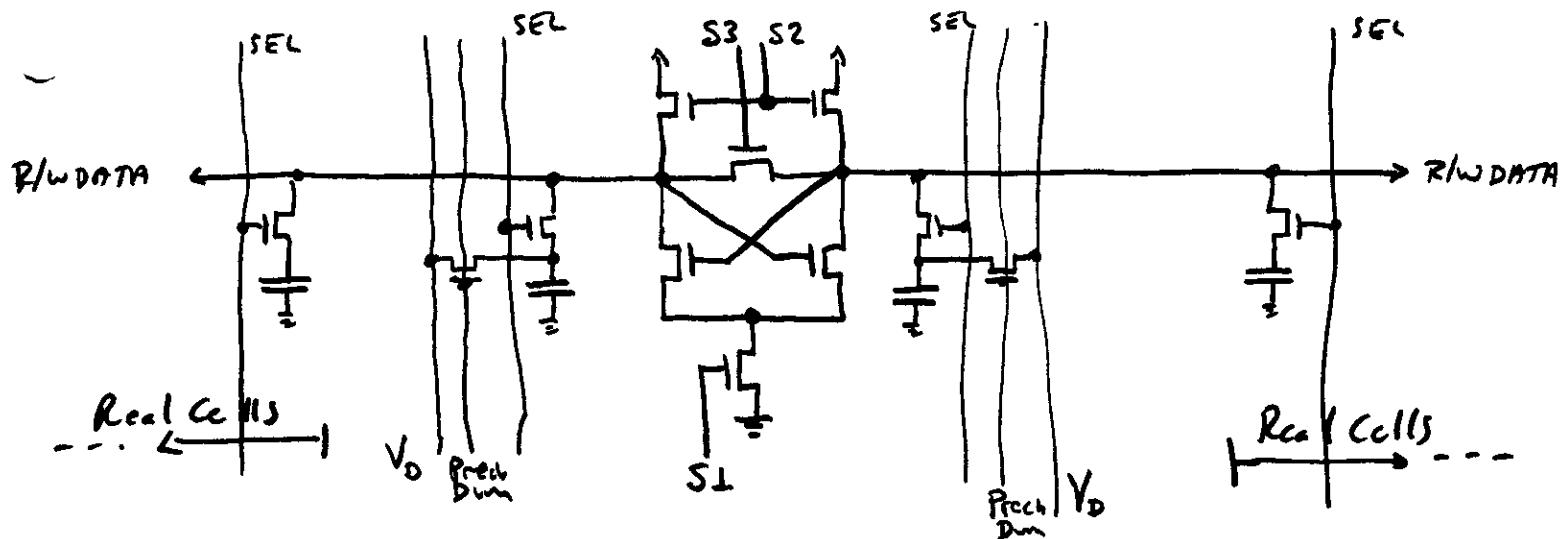
- LAYOUTS: Tricky: I think that to eliminate a Back-to-Read contact, the Capacitor input is on Green, & otherwise goes to Red to some voltage  $V_C \neq GND$ . 2 Poss.b.lities:



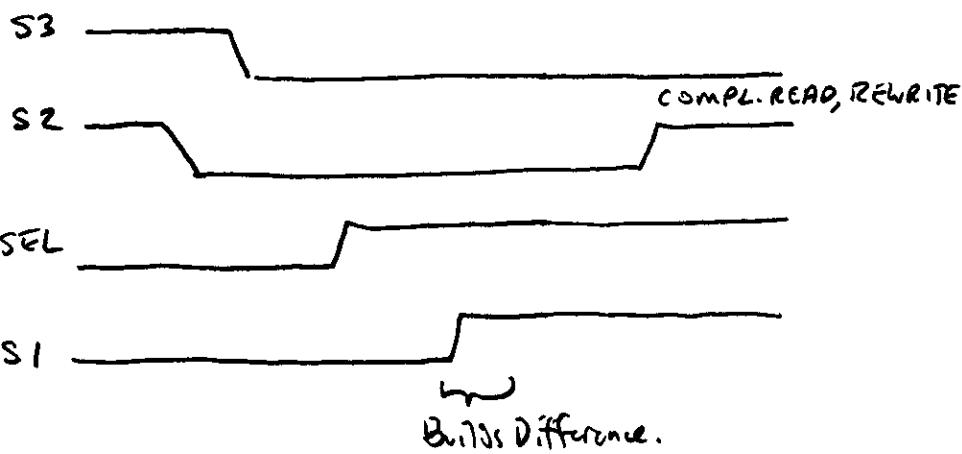
POSSIBLE WAY TO SENSE / REWRITE: ORGANIZE SUBSYSTEM FURTHER AS FOLLOWS:



TILT OVER ONE COLUMN OF THIS ARRAY:



• Dummy Cells Precharged to some  $V_D$  Then to READ / REWRITE:



- When  $S_2, S_3$  on,  $S_1$  off everything prech. high
- When one side high on  $SEL$ , other side discharge faster!
- When  $S_2$  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  $C_L \gg C$ .
- VERY HIGH OVERHEAD DESIGN COMPLEXITY. BUT LOOK AT DENSITY: CONSERVATIVELY: '78  $\lambda = 3\text{ }\mu\text{m}$
- CELL SIZE  $\approx 30\text{ }\mu\text{m} \times 30\text{ }\mu\text{m}$ , SO IN  $4\text{mm} \times 4\text{mm}$ :  

$$\frac{(4000)^2}{(30)^2} \approx 16\text{ K bits}$$
 not counting overhead.
- Actual 16K RAMS MADE WITH  $\lambda$  ON ORDER OF  $\sim 2.5\text{ }\mu\text{m}^+$  ( $5\text{ }\mu\text{m}$  wires). The new 64K RAMS will be similar and use  $\lambda \approx 2.5\text{ }\mu\text{m}$  or a little smaller, (i.e.  $\sim 2.5\text{ }\mu\text{m}$  wires).

### IF TIME: INTRO SOME MATH 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 having "larger wires". **SLIDE 3**
- If do this, we may pay a penalty /bit in that area/bit increases as the Branching Ratio  $\alpha$  decreases (i.e. we branch more often). **SLIDE FIG 5**
- But analysis shows that: Area-Time Product has a minimum for some value of  $\alpha$  **SLIDE FIG 6**
- And that Energy/Access: Also has min at same **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 CARMER PROBLEMS DISCUSSED ON DEC 5



## SEMINAR :

NOVEMBER 21

ON BOARD: 6.978. TODAY: SEMINAR BY RICHARD LYON, M.R.S. X RPAC  
" 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 dimension or connection point changes adjust that we might find slight errors. We'll fix them if you mark.
- Digitize so that Box centers and edges fall on 1/2 micron grid and no finer. i.e. numbers in centimeters should end as --- .50.
- I'll be in today & tomorrow. I'll be away during Thanksgiving. If group want to use lab - 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, SDRNs, etc.
- WE'VE TESTED THE ARPANET FOR SENDING FILES TO PARC --  
I HAVE PLOTS BACK FROM PIECES OF DESIGNS BY STEELE, RICHARD, SPENCER, FORT.

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.

H.I.3 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 PARC?]

- 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 / FILE NAME.
- HAVE ROOM FOR CATEG 1, 2, + a couple of more.  
WILL COMMIT AREA AS I SEE PLOTS, GET FINAL DIMENSIONS
- 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 PadVdd:  
SEMICOLON MISSING: (12 ITEM) ;  
LNG15; Box ---;

optional: one of the boxes is slightly too small if/sober weird if plotted on large scale.

optional: Show PadIn:

You could simply replace PadVdd, PadIn 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	DIA.CIF
✓ ✓ ROYLANCE	ZZZZZ.CIF
✓ ✓ LEVITT	F.F
<hr/>	
✓ ✓ PEREA	BYTELSI
✓ ✓ BROCK, BOUGHTON, BRYANT, LEUNG (DATA MANIP)	BROCK.CIF
✓ ✓ HIRATSUKA	<del>COMPARE.CIF</del> 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)	<del>SP-TESTS</del> <i>&lt;LSI.GOLDIKENER&gt; ROUGH</i>

3. WORK IN PROGRESS. MIght GET ON CHIP

~~HIRATEKA - SP-TESTS~~

✓ ✓ FRAEMAN	PROJECT.CIF
RUBENSTEIN, AZOURY, BOWEN (CFT PROJ).	
✓ ✓ FRANKEL	(FRANKEL.CIF)

4. INSUFF. INFO: ✓ RUBENSTEIN, AZOURY, BOWEN (CRT CONT.)  
REYNOLDS  
McCLELLAN

*<LSI.RUBENSTEIN> CRT*

W = \*SENT TO PARC

- TODAY I'M VERY PLEASED TO INTRODUCE  
RICK DAVIES, MEMBER OF THE RES. STF, XEROX PARC,

RICK 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 <sup>BETTER</sup> "CALIBRATED" SIMULATIONS.



LECTURE # 18NOVEMBER 30

- FINAL PROJECT DETAILS / PLANS
- PRESENTATIONS (<sup>10 to</sup> ~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 PEREZ'S. (MENTION MULT. PARTNER PROJ)

- CRAIG OLSON'S SUGGESTION: MAY EDIT AND GROUP CTS  
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 SPER  
FIELD, FOR EX, DIGITAL FILTERING, GIVE REFS RATHER  
THAN EXT. TR. ON DIG FILTERING.

INCL. COMMENT ABOUT WHAT YOU LEARNED FROM ACTUALLY  
DOING THE PROJECTS 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 ~5 OR SO ---  
MAYBE AT THE FACULTY CLUB -- WOULD REALLY  
LIKE TO HAVE STUDENTS COME TO THAT --- TALK INF.  
WITH CARVER.

-----  
DATE: 29 Nov 1978 4:52 PM (WEDNESDAY)  
FROM: LYON  
SUBJECT: MIT PROJECTS  
To: CONWAY, FAIRBAIRN, ABELL  
cc: LYON

(SOME  
SAMPLE  
MSG  
TRAFFIC)

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

PL0FO\0\FGO

LOGOUT JOB 21, USER CONWAY, ACCT 1, TTY 24, AT 11/29/78 1711  
USED 0:0:13 IN 0:5:34

TENEX(FTI)^

LENET

3 DLS

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). THE 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 SET 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

I'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 PROBLEM 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. ROYLANC 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).

ABELL

- STATUS:> Net was down in early evening of Tuesday.  
 > Files were shipped later, successfully  
 > Problems with F1S ( $C_1F \rightarrow ICARUS$ )  
 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, BROCK, COLN, FRANKEL, GOLDINGER,  
 LAM, LEVITT, OLSON, PEREA, RUBENSTEIN,  
 SNAUER, STERN, ; YANG  
 > F1S problems with  
 ROYLANC, STREET, BALDWIN, FREEMAN, HIRABUNA.  
 > ICARUS problem with SNYDER

- DIAGNOSTICS OR CORRECTIONS: SEE ME FOR DETAILS

BAIN : MISSING SEMICOLON

BROCK: IN UPPERRIGI RAND LOGIC: Some pullups h.c. channels not long enough

CHERRY, FRANKEL, FRANK, PEREA, YANG : STUFF AFTER  
END (;) ?

OTTEN: IMPLANT IN PULLUPS TOO SHARP

SNYDER: PUT DS -- DF around whole design, recoll.

~~ROYLANC~~: MISSING SEMI AFTER COMMENT.

~~BAIN~~: 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 F.  
DF;  
This banked steeler's plot.
- NOTE ASYMMETRY IN PADOUT IT LOOKS SYM. BUT ISN'T!  
SLIDE
- EXTEND IMPLANT 1.5  $\lambda$  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 fuel 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 PROCESSES within each appended file. Place delete definition statements between files for #'s  $\geq 100$ . 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 plot projects w/ this disease. So: Must fix to see merging effects.
  - > Contingency Plans:

FINAL RULES OF THE GAME:

SHOW BOARD  
OF PROJECT PASTELUPS

- PUT COMMENT AT/NEAR BEGINNING WITH YOUR NAME IN IT:  
(PARC FILE = LNAME.CIF);
- PUT ALL LIBRARY SYMB. DEF'S NEXT. ( $\# \leq 99$ )
- PUT YOUR SYMB. DEF'S NXT. USE #'s  $\geq 100$
- PUT DS, DF AROUND ALL THIS. USE #  $\geq 100$
- 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: prof. 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 CIFRN
  - > sign off on Th.B.
- 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 / THUER, MESSAGING WILL BE DONE AT PARC.
- POSSIBILITY OF LAST ITERATIONS OCCURRING (IF FIND SMALL PROBLEMS) ON THURSDAY.  
WILL BUMP OFF PROJECTS WITH OBVIOUS GROSS ERRORS (SUCH AS PADS NOT HOOKED UP --- VDD/GND SHORTS etc.) AND REPLACE WITH OTHERS.
- ~~May need help with project messaging / transmission  
Any volunteer (no --- announce on Tuesday)~~





DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

C A M B R I D G E , M A S S A C H U S E T T S 0 2 1 3 9

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.
- ~~Computer architecture~~ pictorial, Burroughs, one of the architects of the B1700.
- Now man. Res. Staff. at PARC, doing some very ~~interesting~~ work in Computer Architecture.
- The Subject of his seminar is:

- 
- Projects all successfully trans. HD to PARC, successfully merged into ICARUS design file. Now being conv. to PC front end (checkes) -- then on to maskmaking. Tell me 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  
Dept of EE & CS  
University of C.I.T., Berkeley

- Carlo earned his Ph.D. in Physics at the University of Basel, in Switzerland.
- He then joined Bell Labs as a Member of the Tech. Staff, in the early 70's and was one of the principal Charge Transfer Detectors, 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 <sup>since very</sup> been very 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 ---

- 
- Contents:
- Syndic: Address, NID, Quidit as trees are merged.
  - Corby: " " as trees deleted.
  - Lenny/Brent: Deadlock loops in Fifos.

## 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 ultimately 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 MEETING12 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; Hiratsuka;  
Lam; Levitt; Olson, <sup>other</sup>Perca; Roylance;  
Shaver; Snyder; Steele; Stern; Yang;  
one by Bowen, Azoury, ; Rubenstein ;  
one by Goldknecht ; 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 up 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 Nov. and 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. Jon Allen.

Dimitri Antonatos will discuss plans/procedures to be used in packaging

Jon Allen will discuss plans for electrical testing.

### HANDOUT QUESTIONNAIRE FOR INFO

\* LET'S 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 Argonet M. Box 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 --- PAGE ---

ANY QUESTIONS ON PROJECTS ? IAP ? ETC.

---

LOOKING AHEAD : OPPORTUNITIES :

Grad. work; Teaching; Research; Development; Entrepreneurial  
[ ; right now: helping the arena grow; participating in ]  
expanding 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 them after class to see if you think they're worth buying]

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: <sup>rec.</sup> Buy

- ① 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.

- ② A. S. Grove, "Physics and Technology of Semiconductor Devices". a bit dense, but still the classic on process technology and device physics. Excellent Reference.

- ③ 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 "Semicon. Memory Design & Application". If you're interested in memory subsystems, consult this reference.

- Zvi Kohavi: "Switching & Finite Automata Theory" (Gardiner, NY)

- ④ Bell & Newell "Computer Structures: Readings and Examples".

A classic encyclopedia work on computer architecture.

Lots of history and examples. A new edition is about to come out - a recommended buy. Not much of anything on int. circ. / int. syst. but still very interesting.

- ⑤ AND, OF COURSE: NEXT SUMMER, BE SURE & BUY MEAD & CONWAY

- LET'S TAKE 10 MINUTES -- FILL OUT COURSE FEED BACK 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 PRIVILEGE TO TEACH SUCH A FINE GROUP OF STUDENTS.

INDIVIDUALLY AND AS A GROUP YOU'VE ACCOMPLISHED FAR MORE, & IN MUCH LESS TIME, THAN <sup>IN PRED</sup> AT ANY OF THE OTHER SCHOOLS. I'M VERY PROUD OF YOUR ACHIEVEMENTS.

---

I HAVE A FEELING THAT SOME GREAT THINGS WILL BE DONE ~~BY STUDENTS~~ 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.



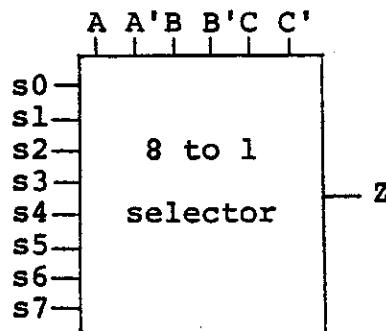
## **IV. Homework and Project Assignments**

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Homework #1. Due: Tuesday, Sept. 19, '78.

- 1(a). Construct a color-coded stick diagram representing the design of an integrated MOS structure which implements the following combinational logic function:

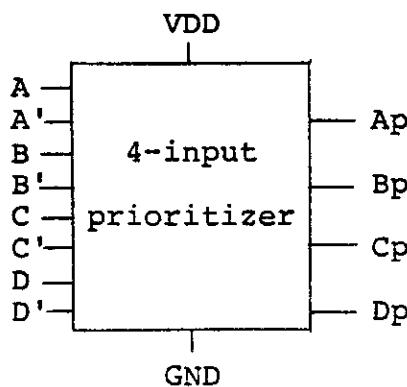
truth table:



A	B	C	Z
0	0	0	s0
0	0	1	s1
0	1	0	s2
0	1	1	s3
1	0	0	s4
1	0	1	s5
1	1	0	s6
1	1	1	s7

- 1(b). Draw the transistor circuit diagram corresponding to your stick diagram solution for 1(a).

- 2(a). Construct a stick diagram design for an MOS structure implementing a 4-input prioritizer function as described below:



truth table (X=don't care):

A	B	C	D	Ap	Bp	Cp	Dp
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

Hint: you'll probably need the GND(logic 0) input, and some solutions use the VDD(logic 1) input.

- 2(b). Explain the principle behind your solution to 2(a), i.e., the basic idea of how your design works. Could your design be expanded in some natural way to implement prioritizers having more inputs? How?

M.I.T., Department of Electrical Engineering & Computer Science

6.978. Homework #1. (continued). Due: Tuesday, Sept. 19, '78.

- 3(a). Design a logic gate implementation of the 8 to 1 selector function described in 1(a). Use NAND and/or NOR logic gates each having two or more inputs. Use the symbols:



- 3(b). Now construct a color coded stick diagram representing the design of an integrated MOS structure which implements your logic gate solution to 3(a), and thus which implements an 8 to 1 selector.

- 3(c). Compare your solutions to 1(a) and 3(b). Any comments? What did you learn from this comparison of these two different approaches to designing a logic function in MOS?

Reading Assignment: Study the following sections in Mead & Conway:

Chap1: Introduction,  
The MOS Transistor,  
The Basic Inverter,  
Inverter Delay,  
Basic NAND and NOR Logic Circuits

Chap2: Introduction

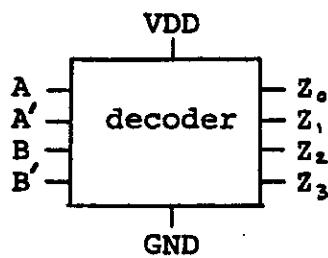
Chap3: Introduction,  
Notation,  
Combinational Logic

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Homework #2. Due: Tuesday, Sept. 26, '78.

4. Design, and illustrate in mixed notation (similar to that in Fig. 5b., Chap. 3, in Mead & Conway) a register array stage which can shift words up one bit, or pass them straight thru, or shift them down one bit.

- 5(a). Design and stick diagram a logic block which decodes a two bit binary number into a four bit unary number. Use whichever of the two truth tables you prefer.



A	B	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Table 1 (or) Table 2

- 5(b). Draw the transistor circuit diagram corresponding to your solution to 5(a).

- 6(a). Generalize the stack cell idea (illustrated in Fig. 10a., Chap. 3, in Mead & Conway) to create a cell which can shift data in 2-dimensions. Sketch your solution in mixed notation as in Fig. 10a, using the additional control signals SHU(active in phase 1) and SHD(active in phase 2). Show at least two cells in adjacent rows.
- 6(b). Stick diagram an nMOS structure which implements the cells of your solution to 6(a). This nMOS cell should be have a form so that it may be repeated both horizontally and vertically in a regular array of cells ( see for example the stick diagram in Fig. 10b in the text ).

Reading Assignment: Study the following sections in Mead & Conway:

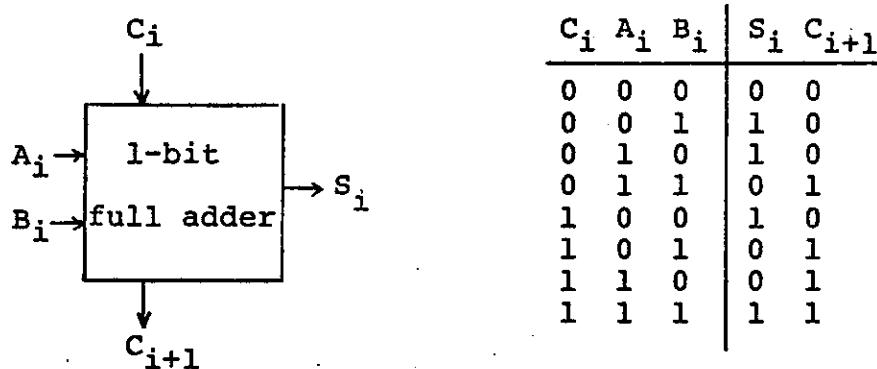
Chap3: Two Phase Clocks, The Shift Register, Relating Different Levels of Abstraction, Implementing Dynamic Registers, Designing a Subsystem.

M.I.T., Department of Electrical Engineering & Computer Science

6.978. Homework #2 (continued):

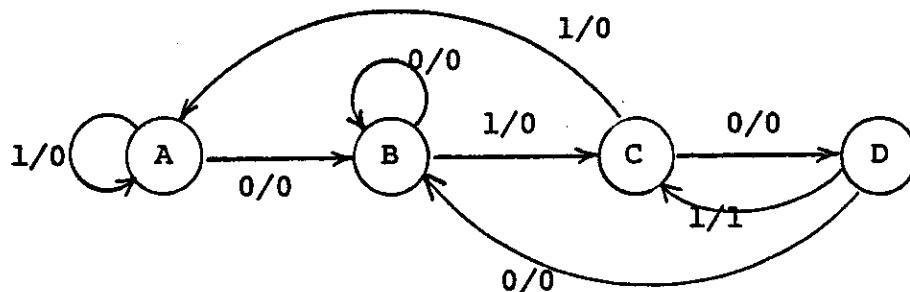
Read & Study: Chap3: The Programmable Logic Array, Finite State Machines.

7. Construct the stick diagram for an nMOS PLA which implements a one-bit stage of a full adder. Include the phase-1 and phase-2 clocked registers which precede and follow the PLA logic (O.K. to use logic symbols for the inverters). Clearly indicate the topology of the carry-in signal path and the carry-out signal path. The full adder function is:



8. Following are a state diagram and symbolic transition table for a (0101) sequence detector (it is assumed, arbitrarily, that the detector starts in state A).

Construct an encoded transition table and then stick diagram an nMOS PLA finite state machine which implements the encoded function. The x/y labels on the arrows indicate the Input/Output values associated with the indicated transitions.



Into Inreg in Phase 1

Into Outreg in Phase 2

State	Input	Next State	Output	Comment
A (start or startover)	0	B	0	may start 0---
	1	A	0	no good, stay at A
B (seen one or more zeroes)	0	B	0	stay if zero
	1	C	0	--01, so go to C
C (seen --01)	0	D	0	ok, go on to D
	1	A	0	fail, go back to A
D (seen --010)	0	B	0	fail, go to B
	1	X C	1	success, output 1, and go to X C

M.I.T., Department of Electrical Engineering & Computer Science

6.978. Homework #3. Due: Tuesday, October 3, '78.

- 9(a). A wire in our system carries messages in a coded binary serial form. When things are running right, all strings of 0's are of even length and all strings of 1's are of odd length in this code. There is a reset wire which carries a signal which indicates the last bit in a message. This reset signal also remains on for many cycles when our system is started, prior to any messages being sent on the message wire.

We wish to design a finite state machine to hang onto the message wire which will output a 1 if it sees an error in a passing message. For example, in the following message bit sequence (M), with the message ends marked by a reset bit (R), we find the indicated errors (E):

R:	1 1 1   0 0 0 0 0 0 0 0 0 0 0 0 1   0 0 0 0 0 1   - - -
M:	X X X   0 0 1 0 0 0 1 1 1 0 1 1 0 0   0 0 1 1 1 0   - - -
E:	X X X   0 0 0 0 0 1 0 0 0 1 0 1 0 0   0 0 0 0 0 1   - - -

MSG1    MSG2

Construct a state diagram showing the states, state transitions, and output under various possible input and state conditions, for a finite state machine specified to produce the error signal E. Hint: watch out for the various message terminating situations illustrated by MSG2.

Now construct a symbolic transition table corresponding to your state diagram. Place some comments next to the various transitions to help others interpret the table.

- 9(b). Construct an encoded transition table from your symbolic table above. Now, stick diagram an nMOS PLA finite state machine which implements the function specified in your encoded transition table.
- 9(c). Which of the above took longer to do, 9(a) or 9(b)? Which of these parts did you consider most difficult?

Reading Assignment: Read & study:

Chap2: Patterning, The Silicon Gate n-Channel Process, Design Rules.  
 Chap4: Introduction

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Homework #3 (continued):

10. Figure 8b in chapter 4 of the text shows the layout of one cell of a shift register. This cell is repeatable in a regular array with an area per cell (if adjacent rows are mirrored to share VDD and/or GND) of 21 lambda by 19 lambda =  $399 \text{ lambda}^2 = 3591 \mu\text{m}^2$ . ('78 lambda =  $3 \mu\text{m}$ ).

For a particular application we need a more compact layout, and are willing to trade power vs delay over a wide range as long as we keep  $Z_{pu}:Z_{pd} = 8:1$ .

Create a more compact layout which still satisfies the design rules given in the text. Plot your layout on a grid of lambda = 1/4 inch. For this problem you may use lines at 45 degrees if you wish (normally in this course we will produce layouts using only lines at right angles). Color your layout so as to clearly differentiate between the various layers.

What is the area per cell of your shift register when repeated in a regular array (in  $\text{lambda}^2$ ) ?

11. Using only lines at right angles, and using colors to identify the layers, layout on a lambda = 1/4 inch grid the stack cell having the topology given in figure 10b in chapter 3 in the text.

Your layout should be repeatable in a regular array. Note that the two halves of the cell shown in the text are the same, but one part is just rotated 180 degrees. So, you need only show the layout for one half of the cell, but carefully producing the layout so that it will properly share VDD and/or GND and abut properly with the rotated half cell. Use min-sized pulldowns, for low power.

What is the area per (full) cell, in  $\text{lambda}^2$ , for your stack layout when repeated in a regular array?

Reading Assignment: Read & study:

Chap2: Electrical Parameters, Current Limitations in Conductors.

M.I.T., Department of Electrical Engineering & Computer Science.

6.978 Homework #4. Due: Thursday, Oct. 12, '78.

- 12(a). Figure 1 on page 2 of this assignment sketches the block diagram of a serial bit string comparator. The function of this subsystem is as follows: A stream of data bits is clocked through a Data Register, which might be 64, 128, 256, or more bits in length. Each clock cycle, the data bits in the register are compared with a previously loaded pattern of bits in a Key Register. However, only a subset of the bits are actually compared, namely those in positions marked by a pattern of bits in a Mask Register. The comparison is made in a Comparator, which has a Match Line running through it. If any of the data bits in positions marked by the mask bits are different from the key bits, then the Match Line is "pulled low". Otherwise it is left high.

One possible MOS design for the bit string comparator is suggested by figure 2, which shows a one-bit vertical slice of a design, in mixed notation.

Stick diagram an MOS structure which implements this one-bit vertical slice. You might try several alternative approaches, keeping in mind the future implications on layout, before you settle on a final stick diagram.

- 12(b). Construct a layout diagram corresponding to your stick diagram for 12(a). The one-bit vertical slice should be repeatable horizontally. Use only lines at right angles.

There are many ways to simplify the plotting of the layout of this slice: Note that the two bottom registers are (almost!) mirror images of each other. So you don't need to repeat the full plot of both of these. You might do the important cells on a  $\lambda = 1/4$  inch scale, and then carefully construct an overall plot on a  $\lambda = 1/8$  inch scale. But, whatever ideas or tricks you use, be sure that the design rules can be checked, and that the locations of all objects can be determined from your plots.

Reading Assignment: Read & Study:

Chap 1: Driving Large Capacitive Loads, Super Buffers.

Chap 4: Patterning and Fabrication

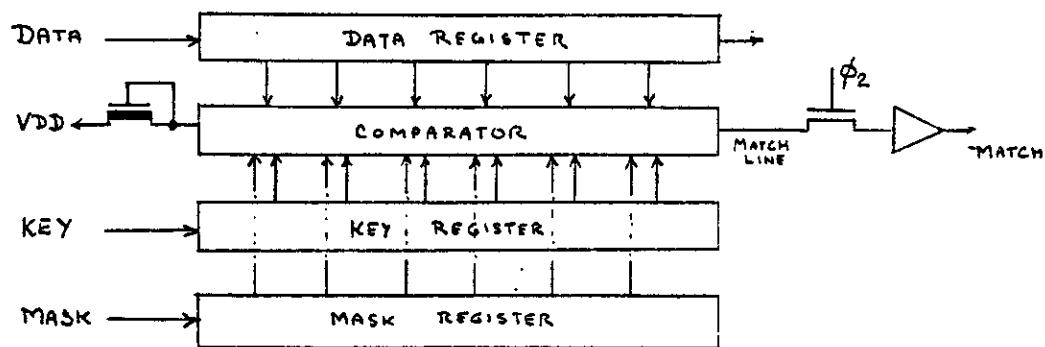


Figure 1. A Serial Bit String Comparator Subsystem.  
(Block Diagram)

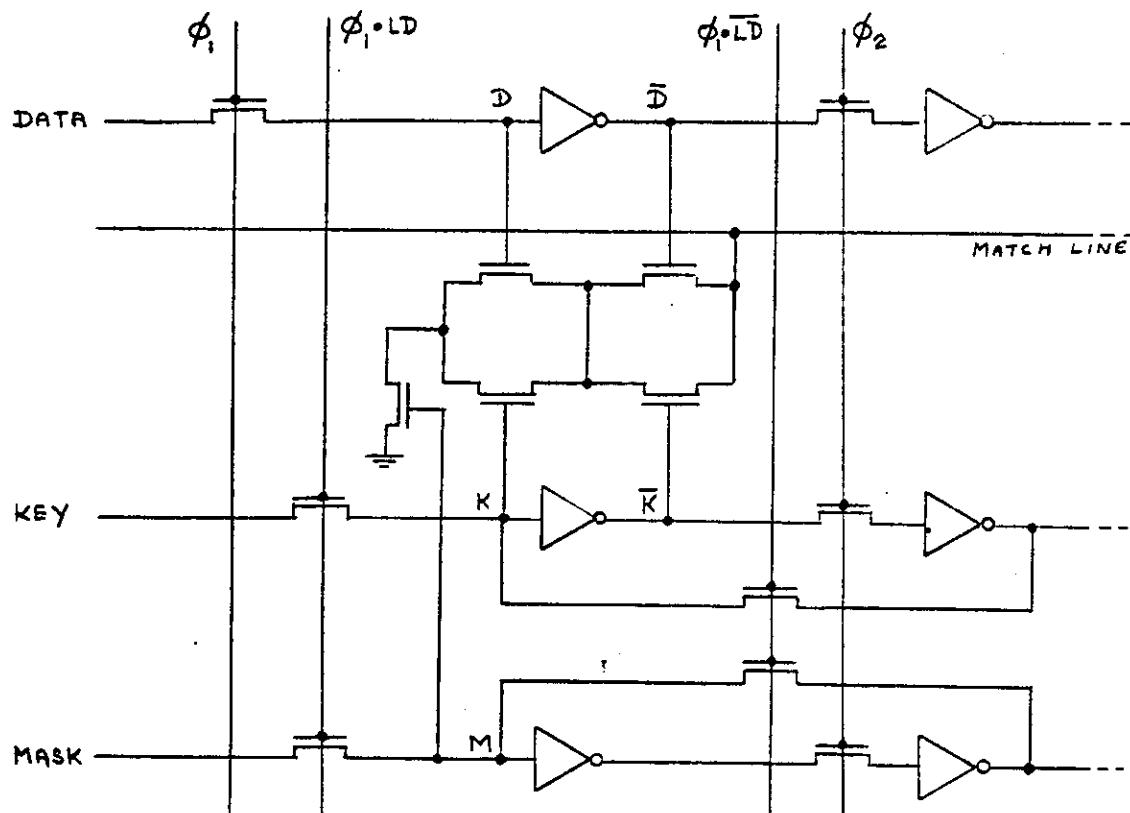


Figure 2. A One-Bit Vertical Slice thru One Possible  
nMOS Circuit/Logic design of the  
Serial Bit-String Comparator.

M.I.T., Department of Electrical Engineering and Computer Science.

6.978. Introduction to VLSI Systems.

This Handout Contains:

1. Homework Assignment #5.
2. Project Assignment #1.
3. Example CIF-code.
4. List of CIF errata in text.

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Homework #5. Due: Tuesday, Oct. 17, '78.

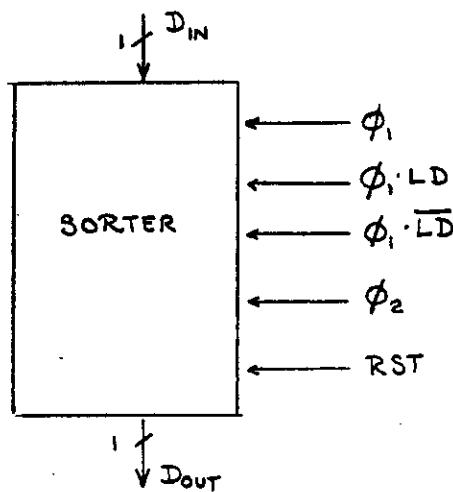
Note: Keep a copy of your solutions to this HW.

13. Construct a CIF-code description of the layout of the two dual-port register cells pictured in Figure 22a in chapter 5 of the text. However, modify the layout as needed to eliminate the use of boxes at other than direction (1 0).
14. Sorters of various types provide interesting examples of "smart memory" subsystems having important applications. Sorters are often fairly directly mappable into integrated nMOS structures.

Suppose we wish to design a serial input, serial output bubble sorter having the block diagram shown below. This subsystem is loaded with a bitstream of  $N$  words of  $m$  bits each. We then wish to sort the words with the largest rising to the top. It is then unloaded at the bottom with this unloading perhaps overlapped with the loading of another  $N$  words.

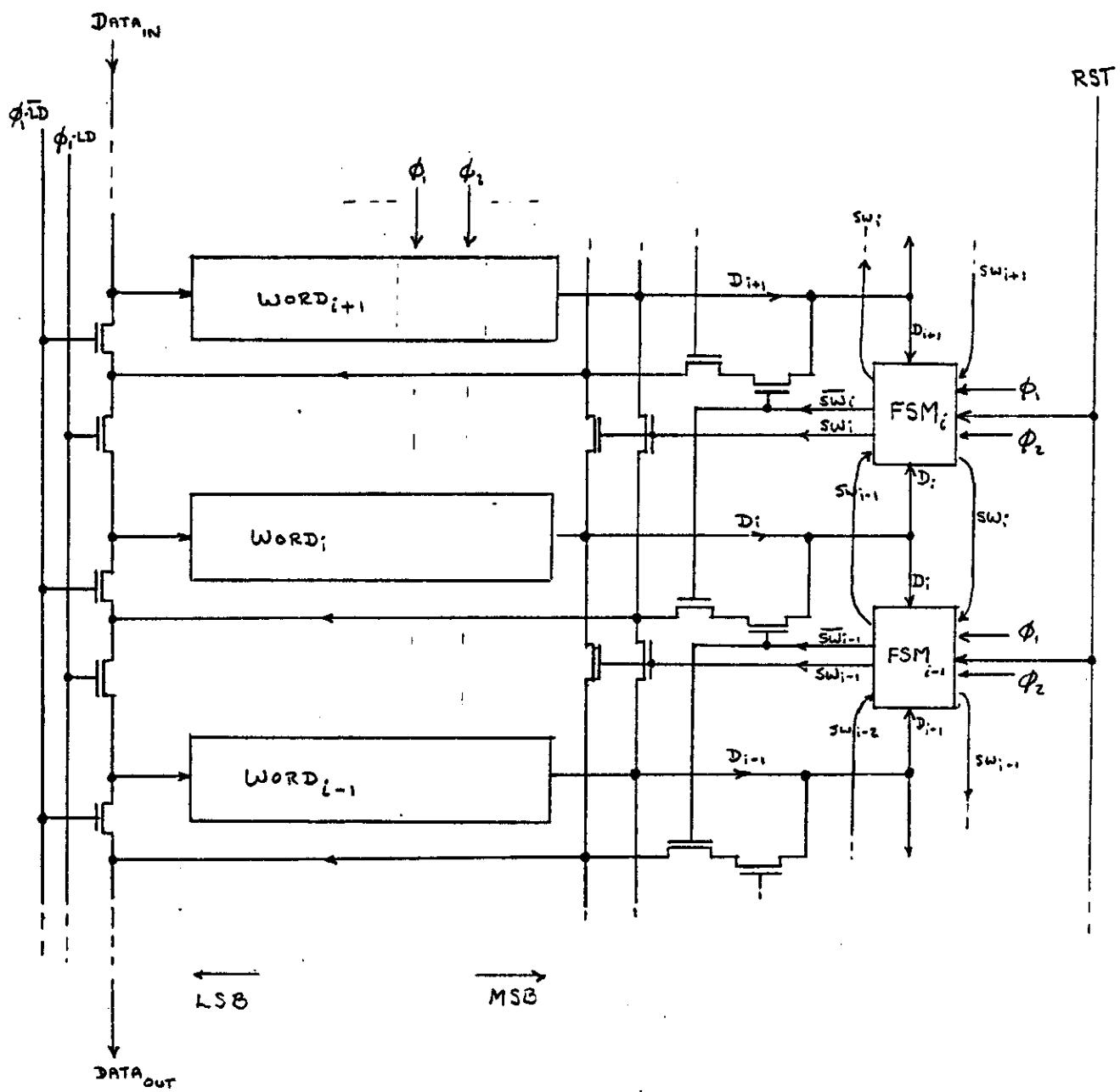
The sorter memory array contains a processing element between each adjacent pair of  $m$  bit words. If properly organized, the loaded sorter can perform a complete bubble sort in  $(N) \times (m)$  phase-1/phase-2 cycles. It can thus sort  $N$  words, using a memory space of  $N$  words, in time of order  $N$ .

During each set of  $m$  cycles, adjacent pairs of words are either recirculated in place or swapped, depending on which is larger. All these adjacent pair swapings can take place concurrently. Note that no two words may request swapping with a word in between them, if the decision to swap is based on the first encountered bit position where a lower word is = 1, and the word above is = 0.



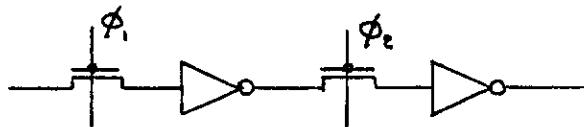
## 6.978. Homework #5. Problem 14 (cont.).

A possible organization for the rows of words and processing elements is given in the diagram below. Each row is a simple  $m$  bit shift register. Whether the shift registers recirculate or swap with a neighbor depends on the state of a finite state machine in between each pair of rows. The state sequencing of each finite state machine depends on the data shifted out of the adjacent word rows, according to the state diagram below. A RST control signal is provided after every  $m$  bits of shifting, to reset the state machines for the next set of swappings.

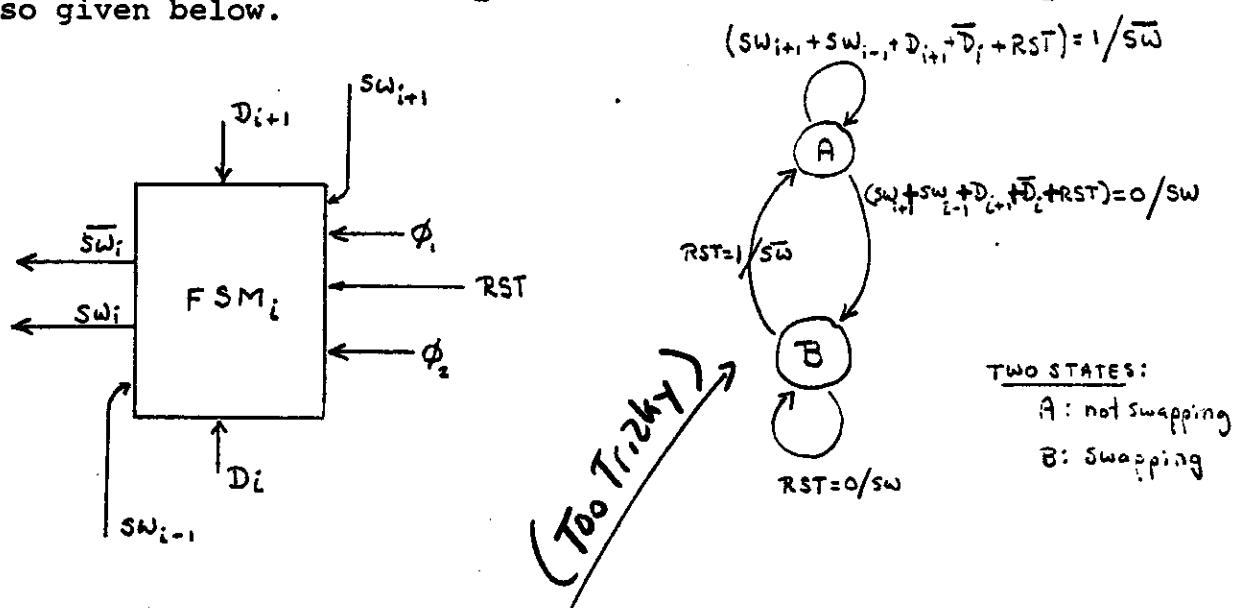


## 6.978. Homework #5. Problem 14 (cont.).

Note that each bit position within each word row is simply a one bit shift register clocked on phase-1/phase-2 as indicated. At the start of recirculation/swapping the most significant and least significant bits of each word are in the positions indicated on the previous page.



The finite state machines are the same for all row pairs. A block diagram for  $FSM_i$  is given below, indicating its inputs and outputs. Inputs are the data bits from the  $(i)$ th and  $(i+1)$ st rows, the RST line, the "swap" lines output from the adjacent state machines, and the clocks. Outputs are the true and complement values of "swapi". The state diagram for  $FSM_i$  is also given below.



**The Problem:** Design an nMOS circuit implementing  $FSM_i$ . Carry the design only through to a circuit diagram. Hint: You probably don't want to use a PLA for this one. Also, be sure to do the right things on the right clock phases. Think through the functioning of the overall subsystem. Does it really seem to work?

**Reading Assignment:** Read the following sections in the text:

Chapter 4: Hand Layout and Digitization Using a Symbolic Layout Language; The Caltech Intermediate Form.

Most people didn't take the hint and figure out that 3 states are required. I'd replace this diagram with the one from problem 15. Then have problem 15 be just to stick diagram the structure.

M.I.T., Department of Electrical Engineering & Computer Science  
6.978. Project Assignment #1. Due Thursday, October 26, 1978.

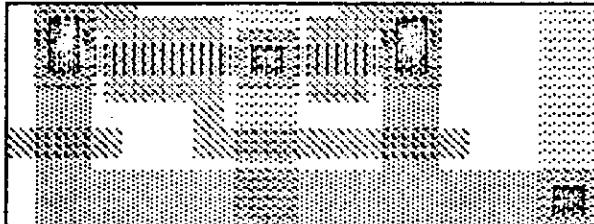
The project lab is located in room 36-561. For information on lab scheduling, call Joy Thompson (Sec'y to Prof. Jonathan Allen) at 253-7309.

1. Learn and practice: Login; creation, editing, and filing of text files; listing of text files on the line printer; plotting of CIF-code layout files on the HP color plotters.
2. Key-in and plot your CIF-code for Homework problem #13.
3. Project Selection: Select and briefly describe your proposed integrated system project.
  - (a) Provide a short description of the function of the system/subsystem.
  - (b) Construct a block diagram of a possible organization of the project, identifying the key components, inputs, outputs and controls.
  - (c) Indicate that portion of the system/subsystem that you plan to complete through layout by late November for possible inclusion on the M.I.T. '78 multi-project chip set.
  - (d) Do you plan to collaborate with others? If yes, please list their names in your proposal, and indicate the nature of the collaboration, i.e., on project design, or on design checking.
  - (e) If you would like to collaborate with others, but haven't yet made arrangements to do so, indicate this in your proposal, and whether you seek others to participate in design, or design checking.

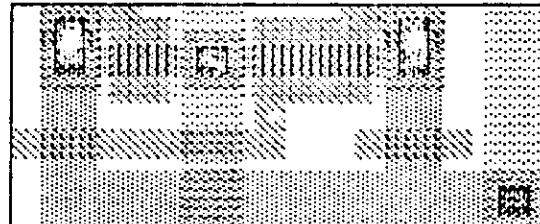
Documentation for Inverters:  
**InverterPair** .. **BackwardInverterPair**  
**InverterQuad** .. **BackwardInverterQuad**

Date	July 25, 1978	Status	used in summer 1978 MPC
Designer	Bob Baldwin	Address/Phone	PARC SSL
Info File	Inverters.LibDoc	CIF File	Inverters.cif
Design Rules	Mead/Conway	Scale	$\lambda = 3 \mu\text{m}$
<b>Dimensions and Replication</b>			
	InverterPair    X: 41 $\lambda$	Y: 16 $\lambda$	DX: no    DY: no
	Back...Pair    X: 37 $\lambda$	Y: 16 $\lambda$	DX: no    DY: no
	InverterQuad    X: 41 $\lambda$	Y: 28 $\lambda$	DX: no    DY: 30 $\lambda$
	Back...Quad    X: 37 $\lambda$	Y: 28 $\lambda$	DX: no    DY: 30 $\lambda$
Further Info	See section 7.3 of <i>A Guide to LSI Implementation</i> .		
Function/Use	Produces the inverted and non-inverted value of a signal that has been routed through pass transistors (i.e. the first inverter has $Z_{pu}/Z_{pd} = 8$ ). For InverterPair, the input comes from the left, and both the true and complement output are available in red and green on the right. BackwardInverterPair has the input on the right, and output on the left.		
	...Quads are pairs of ...Pairs, sharing a ground line between them.		
Connections	Vertical Vdd, Ground, and clocks in metal 4 $\lambda$ wide. Input is on the left (right) side in red 2 $\lambda$ wide. Both the true and complement output are available in red and green on the right (left) side.		
Included Cells	None.		
Loadings	Input load: $8\lambda^2$ gate = .03 pf (fanout of 1) The complement output drives the second inverter, so it already has a load of 1 fanout.		
Performance	True output time constant: $\sim(20k\Omega/2)\cdot C_{load}$ ~10ns/pf. Internal delay: output rises $\sim((f_{comp}+1)+4f_{true})\tau$ , falls $\sim(8(f_{comp}+1)+f_{true})\tau$ . Complement output time constant: $\sim(20k\Omega/4)(C_{load}+.03\text{pf})$ ~5ns/pf. Internal delay: output rises $\sim 8(f_{comp}+1)\tau$ , output falls $\sim(f_{comp}+1)\tau$		

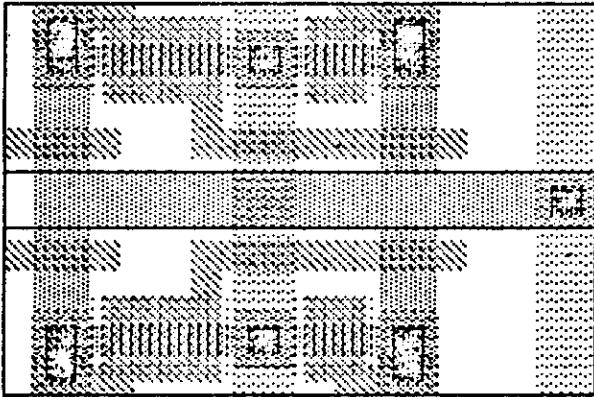
InverterPair



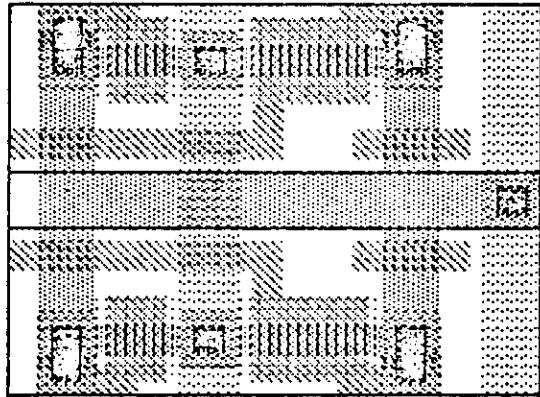
BackwardInverterPair



InverterQuad



BackwardInverterQuad



[ Note: These stipple patterns are different from those used in Chapter 4 in the textbook. ]

file: Inverters.cif

( Created by Sif from Inverters.ic );

DS 1: ( Name: InverterPair );

( 27 Items. );

```
Layer NPol: Box Len 2400 Wid 600 Center 1200,-3000 ;
Layer NPol: Box Len 1200 Wid 900 Center 1200,-450 ;
Layer NDif: Box Len 1200 Wid 3000 Center 1200,-2100 ;
Layer NDif: Box Len 11700 Wid 1200 Center 6450,-4200 ;
Layer NMet: Box Len 1200 Wid 1800 Center 1200,-900 ;
Layer NCut: Box Len 600 Wid 600 Center 1200,-600 ;
Layer NCut: Box Len 600 Wid 600 Center 1200,-1200 ;
Layer NPol: Box Len 1500 Wid 600 Center 1950,-300 ;
Layer NDif: Box Len 3600 Wid 600 Center 3300,-1200 ;
Layer NImp: Box Len 3000 Wid 1800 Center 3300,-1200 ;
Layer NPol: Box Len 2400 Wid 1800 Center 3300,-1200 ;
Layer NPol: Box Len 600 Wid 1200 Center 4200,-2100 ;
Layer NPol: Box Len 5700 Wid 600 Center 6750,-3000 ;
Layer NMet: Box Len 1200 Wid 4800 Center 5400,-2400 ;
Layer NDif: Box Len 1200 Wid 1200 Center 5400,-1200 ;
Layer NCut: Box Len 600 Wid 600 Center 5400,-1200 ;
Layer NDif: Box Len 2700 Wid 600 Center 6750,-1200 ;
Layer NImp: Box Len 1800 Wid 1800 Center 6900,-1200 ;
Layer NPol: Box Len 1200 Wid 1800 Center 6900,-1200 ;
Layer NPol: Box Len 1200 Wid 600 Center 7500,-300 ;
Layer NDif: Box Len 1200 Wid 3000 Center 8400,-2100 ;
Layer NPol: Box Len 1200 Wid 900 Center 8400,-450 ;
Layer NMet: Box Len 1200 Wid 1800 Center 8400,-900 ;
Layer NCut: Box Len 600 Wid 600 Center 8400,-600 ;
Layer NCut: Box Len 600 Wid 600 Center 8400,-1200 ;
Layer NMet: Box Len 1200 Wid 4800 Center 11700,-2400 ;
Layer NCut: Box Len 600 Wid 600 Center 11700,-4200 ;
DF;
```

DS 2: ( Name: InverterQuad );

( 4 Items. );

Call 1 Trans 0.0;

Call 1 Mir Y Trans 0,-8400;

```
Layer NMet: Box Len 1200 Wid 3600 Center 5400,-6600 ;
Layer NMet: Box Len 1200 Wid 3600 Center 11700,-6600 ;
DF;
```

DS 3: ( Name: BackwardInverterPair );

( 27 Items. );

```
Layer NPol: Box Len 5700 Wid 600 Center 2850,-3000 ;
Layer NDif: Box Len 1200 Wid 3000 Center 1200,-2100 ;
Layer NPol: Box Len 1200 Wid 900 Center 1200,-450 ;
Layer NMet: Box Len 1200 Wid 1800 Center 1200,-900 ;
Layer NDif: Box Len 10500 Wid 1200 Center 5850,-4200 ;
Layer NCut: Box Len 600 Wid 600 Center 1200,-600 ;
Layer NCut: Box Len 600 Wid 600 Center 1200,-1200 ;
Layer NDif: Box Len 2700 Wid 600 Center 2850,-1200 ;
Layer NPol: Box Len 1200 Wid 600 Center 2100,-300 ;
Layer NImp: Box Len 1800 Wid 1800 Center 2700,-1200 ;
Layer NPol: Box Len 1200 Wid 1800 Center 2700,-1200 ;
Layer NMet: Box Len 1200 Wid 4800 Center 4200,-2400 ;
Layer NDif: Box Len 1200 Wid 1200 Center 4200,-1200 ;
Layer NCut: Box Len 600 Wid 600 Center 4200,-1200 ;
Layer NDif: Box Len 3600 Wid 600 Center 6300,-1200 ;
Layer NImp: Box Len 3000 Wid 1800 Center 6300,-1200 ;
Layer NPol: Box Len 2400 Wid 1800 Center 6300,-1200 ;
Layer NPol: Box Len 600 Wid 1200 Center 5400,-2100 ;
Layer NPol: Box Len 1500 Wid 600 Center 7650,-300 ;
Layer NPol: Box Len 2400 Wid 600 Center 8400,-3000 ;
Layer NPol: Box Len 1200 Wid 900 Center 8400,-450 ;
Layer NDif: Box Len 1200 Wid 3000 Center 8400,-2100 ;
Layer NMet: Box Len 1200 Wid 1800 Center 8400,-900 ;
Layer NCut: Box Len 600 Wid 600 Center 8400,-600 ;
Layer NCut: Box Len 600 Wid 600 Center 8400,-1200 ;
Layer NMet: Box Len 1200 Wid 4800 Center 10500,-2400 ;
Layer NCut: Box Len 600 Wid 600 Center 10500,-4200 ;
DF;
```

DS 4: ( Name: BackwardInverterQuad );

( 2 Items. );

Call 3 Trans 0.0;

Call 3 Mir Y Trans 0,-8400;

DF;

End

<- TYPE 114  
(MSG. # 114, 1330 CHARS)  
DATE: 2 Oct 1978 1:40 PM (MONDAY)  
FROM: LYON AT PARC-MANO  
SUBJECT: CIF 2.0 bugs  
To: CONWAY, SPROULL, HONGMEUR, FAIRBAIRN, WILNER, TRIM  
cc: LYON

...AND COPIES TO MEAD, GRAY, SEQUIN, AYRES, HENKE, AND MORE.

THE CIF 2.0 DESCRIPTION IN MEAD&CONWAY'S "INTRODUCTION TO VLSI SYSTEMS" HAS SEVERAL BUGS DUE TO EDITTING ERRORS.

THE WORST ERROR IS AN AMBIGUITY IN THE ORDER OF THE LENGTH AND WIDTH PARAMETERS TO THE Box COMMAND. THE ORDER APPEARS DIFFERENTLY ON PAGES 19 AND 21 OF THE CIF SECTION, UNDER THE "COMMANDS" COLUMN AND AS THE EXAMPLE OF A Box COMMAND. SPROULL AND I HAVE DECIDED THAT THE OFFICIALLY SUPPORTED VERSION SHOULD BE LENGTH, THEN WIDTH (FOR A DEFAULT DIRECTION, THIS IS X SIZE, THEN Y SIZE).

A RESULT OF THIS INTERPRETATION IS THAT THE CIF GENERATED AT CMU LAST YEAR IS NOT CORRECT, AND PROGRAMS THAT HANDLE IT AT CMU AND CALTECH ARE OBSOLETE. CONVERSION TO THE CORRECT FORM SHOULD BE TRIVIAL, HOWEVER. THE CIF SECTIONS IN "A GUIDE TO LSI IMPLEMENTATION" ARE BELIEVED TO BE CORRECT.

OTHER PROBLEMS THAT APPEAR IN M&C ARE:

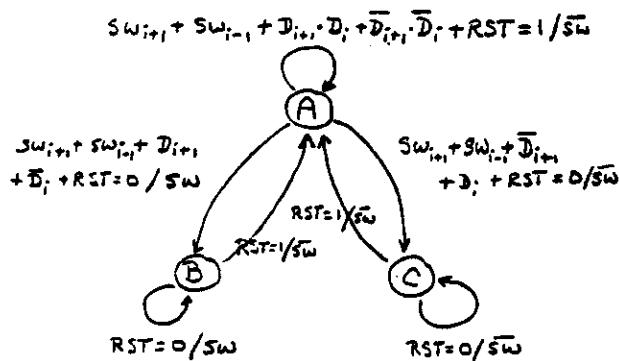
1. P.25 HAS AN EXAMPLE WITH S COMMANDS. SUBSTITUTE R COMMANDS.
2. P.26 HAS COMMENTS WITH THE OLD "/" DELIMITER. SUBSTITUTE "(....)".
3. PP.18&30 HAVE RON AYRES'S NAME MISSPELLED.

CREDIT GOES TO BILL HENKE OF MIT FOR REPORTING MOST OF THESE BUGS.

DICK

6.978. Homework #6. Due: Tuesday, October 24, '78.

15. As we've seen, the FSM's controlling the sorter described in problem 14 must each have at least three states. one possible state diagram for  $\text{FSM}_i$  is as follows:

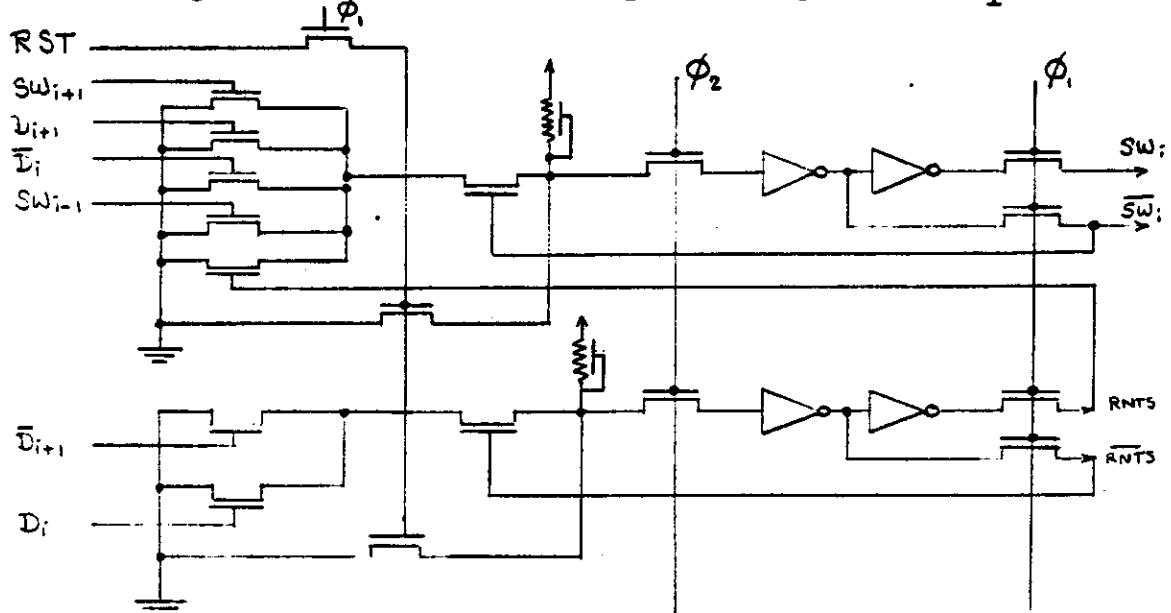


STATES:  
 A : NOT YET SWAPPING/RESET  
 B : SWAPPING  
 C : REMEMBER NOT TO SWAP

SW	RNTS
0	0
1	X
0	1

IF  $SW_{i+1} + SW_{i-1} + RST = 1$ , STAY IN ④  
IF  $SW_{i+1} + SW_{i-1} + RST = 0$ ,  
THEN IF  $D_{i+1} = D_i$ , STAY IN ④  
IF  $D_{i+1} < D_i$ , GO TO ⑤  
IF  $D_{i+1} > D_i$ , GO TO ⑥

Now, one possible MOS circuit implementing this  $\text{FSM}_i$  is:



Starting with the above circuit, and with the subsystem diagram given in HW #5, page 2, construct a stick diagram of  $\text{FSM}_i$  and the first/last bit positions of the two adjacent word registers, including the load control switches.

Reading Assignment: Read & Study:

Ch1: Delays in Another Form of Logic Circuitry, Transit Times And Clock Periods.

Ch2: Yield Statistics.

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Project Assignment Schedule:

Assignment 1: Due Thursday October 26: Proposed Project Selection.

Assignment 2: Due Thursday November 9: Detailed Project Description.

In a format of your choosing, provide detailed block diagrams and circuit/logic diagrams of your project design. Include a short written description of the project function, and a detailed description of the algorithms involved. The idea here is to produce a description that others could review to help you uncover problems/errors at the top level before you get too far into detailed design. If possible, provide stick diagrams of key cells, and an estimate of the area your project will require on the chip.

Assignment 3: Due Tuesday November 21: Preliminary Layout.

Update your Assignment 2 description to include any changes made during detailed design and layout. Append copies of your stick diagram design notes and copies of checkplots of your project cells. Append any written comments which might be helpful in clarifying your overall layout. You should have completed at least a preliminary version of your entire project layout by this date.

Assignment 4: Due Thursday December 7: Final Project Report.

In a format of your choosing, drawing on all the preceding materials and your final design results, describe your project at the various levels from subsystem overview down to final layout.

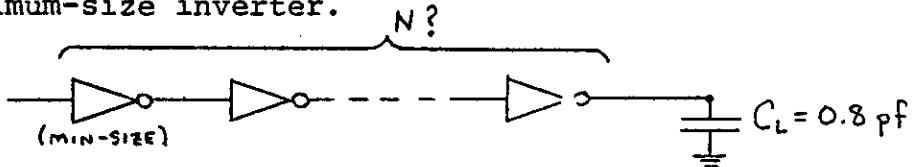
Note: Projects will be selected during the period 21 to 28 Nov., for inclusion in the M.I.T. multi-project chip set. Final design files will be merged into the chip file on 5 Dec.

M.I.T. Department of Electrical Engineering & Computer Science

6.978. Midterm Exam. November 2, 1978.

Problem 1. (25%) (use elect. parameters tabulated on p.15, Chap.2)

(a) You are designing a control driver which must drive a 0.8 pf capacitive load. Minimum delay is essential in this case, and you are not concerned with area. The control signal originates from a minimum-size inverter.



How many stages of inverters (or super-buffer sections) would you use, and what would be the ratio of their successive sizes?

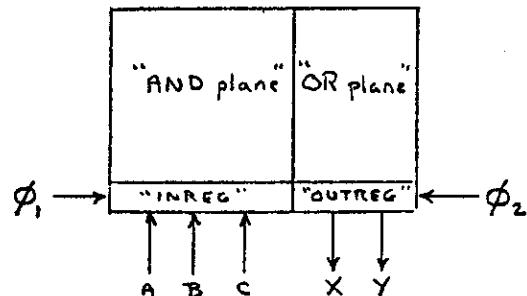
(b) How wide (in microns) should the VDD and GND wires be which supply power to the OM-2 Register Array Subsystem pictured in the frontispiece in the text? The detailed layout of each pair of cells is shown in Fig. 22a, Chap. 5. Assume that metal wires are 1 micron in height, and use a maximum current density of 1 ma per micron<sup>2</sup>. Clearly indicate your assumptions and each step in your calculation.

Problem 2. (25%)

Using a mixed notation as in Fig.13c, Chap.3, give a stick diagram for a NOR-NOR type nMOS PLA (including its input/output registers) which implements the following functions. Use a minimum number of product terms.

$$X = (A + \bar{B})(\bar{A} + B + C)(B + \bar{C})$$

$$Y = (A + C)(\bar{A} + B + C)$$



## 6.978. Midterm Exam (cont.)

Problem 3. (25%)

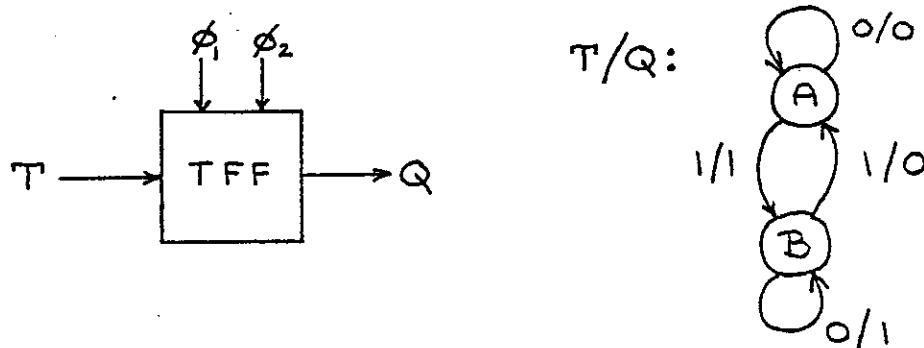
(a) Assume that all lines in Fig. 4b, Chap. 5, are of minimum width. What is the design rule violated in this figure?

(b) What is the major error in the design in Fig. 17, Chap. 4.

(c) You've become tired of the tedious job of digitizing layouts directly into CIF code. So, you've decided to define a Symbolic Layout Language, and construct a translator to CIF code. What are three capabilities, beyond or in place of those of CIF, that you would include in your language to simplify the task of describing layouts?

Problem 4. (25%)

Design and give a circuit diagram for an MOS circuit implementing a Toggle Flip-Flop (TFF). The TFF changes state (after a full  $\phi_1$ - $\phi_2$  clock cycle) if its input T is high during  $\phi_1$ . If T is low during  $\phi_1$ , the TFF remains in the same state. The block diagram and state diagram for the TFF are as follows:



Briefly but carefully describe the functioning of your circuit (to help the grader believe that it really works!).

## **V. Other Course Handouts**

**6.978: Introduction to VLSI Systems:**

*Background Questionnaire: The following questions are not meant to establish requirements, but are an information gathering tool to help in course and course project planning.*

Name: \_\_\_\_\_ Course: \_\_\_\_\_ Year: \_\_\_\_\_

Local Address/phones: \_\_\_\_\_

Major Technical Interests: \_\_\_\_\_

M.I.T. Courses taken: 6.011  6.012  6.031  6.032  6.082  6.112

Other equivalent or relevant courses taken: \_\_\_\_\_

Could you sketch the block diagram of a stored program computer? \_\_\_\_\_

Have you ever designed a major digital subsystem such as a CPU, an I/O controller, etc? \_\_\_\_\_

Could you sketch the block diagram of an assembler? \_\_\_\_\_

Have you ever designed and implemented a major software subsystem such as an assembler or compiler? \_\_\_\_\_

Describe your most ambitious software or hardware project? \_\_\_\_\_

What programming languages do you know? \_\_\_\_\_

Had any experience with graphic design languages? \_\_\_\_\_ As a user of computer aided design systems? \_\_\_\_\_

Have you any experience in the design of high-level languages? \_\_\_\_\_ Of data structures? \_\_\_\_\_

Have you ever interfaced and programmed microprocessors? \_\_\_\_\_

Had any experience in micro-programming? \_\_\_\_\_ For emulation? \_\_\_\_\_ For I/O control? \_\_\_\_\_

Could you draw the logic diagram of a 4-bit, 16-instruction ALU to the gate level? \_\_\_\_\_

Could you design, and implement in TTL, a finite state machine controller such as a traffic light controller? \_\_\_\_\_

Could you draw the current vs voltage characteristics of an MOS transistor? \_\_\_\_\_

Ever laid out an integrated circuit? \_\_\_\_\_ In what technologies? \_\_\_\_\_

If yes, what was your most complex circuit? \_\_\_\_\_

And, if yes, did you make use of a circuit simulation program? \_\_\_\_\_ Which one? \_\_\_\_\_

Could you describe the wafer fabrication sequence for an MOS-IC? \_\_\_\_\_

What do you hope to learn from this course?

---

---

---

**6.978. Scheduling Questionnaire:** (return on Thurs., Sept. 28)

Your completion of the following questionnaire will help in planning the scheduling of lab sessions and possible seminars.

Place 1's in those boxes indicating times you would prefer to attend lab sessions or seminars. Place at least six 1's in the table.

Place 0's in those boxes indicating times during which you could not attend, or it would be difficult for you to attend lab sessions or seminars.

Name: \_\_\_\_\_

PM		MON	TUE	WED	THU	FRI
	1-2		0		0	
	2-3		0		0	
	3-4					
	4-5					
	5-6					
	6-7					
	7-8					
	8-9					
	9-10					

CIF Tran  
users  
Manual

CIFTran Users' Manual

Version 1.0

23 October 78

Bill Henke

Research Laboratory of Electronics  
Rm 36-525  
Massachusetts Institute of Technology  
Cambridge, Mass. 02139

Contents

Preface	2
Language Exceptions From Standard	2
Usage	3
Acknowledgment	4

## Preface

---

CIF is a textual language for describing graphic items (mask features) of interest to LSI circuit and system designers. The defining document for CIF is a section entitled "The Caltech Intermediate Form for LSI Layout Description" in the text "Introduction to VLSI Systems" by Mead and Conway (to be published by Addison-Wesley).

This CIFTran Users' Manual documents the use of a particular implementation of a translator for that language. This translator reads and interprets files of CIF code, and generates several forms of output.

Independent of the type(s) of output selected, a translation pass always reads and completely interprets the CIF file in the same way, and generates syntax error diagnostic reports.

Currently implemented forms of output included commands to drive a HP 7221A four color graphics plotter, graphic commands for a Tektronics DVST graphics terminal, and an alphanumeric representation of "pen type" commands such as move, draw, and select layer.

## Language Exceptions From Standard

---

The CIF language supported by this translator is that of the "standard" (as amended in a note dated 2-Oct-78 by Lyon) with the following exceptions. All transformations and the nesting of symbol calls (instantiations) are supported. The "P", "R", "W", and "DD" commands are currently unsupported. Unimplemented CIF commands will be reported either as unimplemented or as "unknown commands".

The following text string is an example of a complete acceptable CIF file.

```
(EX1.CIF Example of CIF code );
( Symbol definitions );
( A single box, conceived of with a lambda of 3 );
Define Symbol 1 Scale 3/1;
Layer NDif;
Box Length 4 Width 2 Center 2,1;
DF;
(
A cell consisting of two calls on the single box.
( This is a nested comment );
);
```

```
DS 2 4 1;
Call 1 Rotate to 0,1 Translate to 10,0;
Call 1 Translate to 0,-2;
DF;

(Main program);
Call 2;
Layer NPoly; Box L 30 W 10 40,40 Dir 1,1;
E
```

## Usage

---

The CIF translator is started by executing the program currently named "CIFTRN". Upon initiation, the program will read in various options and parameters from a file named "CIFTRN.PAR". Such a file must exist somewhere in the current search path. It is suggested that each user make a copy in his own directory, since each application will entail different parameter values. (The designation [name] of the file to be copied is <LSI.UTILITIES>CIFTRN.PAR.) Parameters are set by using a text editor to edit this file. The format of the file consists of a header and trailer which should not be modified, and a listing of option situations expressed in the following form:

```
<option name> = <option value>
```

The various option names, their function, and acceptable values are as follows:

ISPLTO - Is Plotter Output desired on this pass. Acceptable values are T (true or yes) or F (false or no). If plotter output is requested, a particular plotter must first be allocated/assigned/connected. The policy and procedure for doing this will be selected by the teaching assistant on duty. (A "local logical name" of "HP:" must be defined to be one of the plotter connections before running CIFTRN.)

ISALFO - Is Alphanumeric Output desired on this pass, acceptable values are T or F. If alphanumeric output is desired, a destination designator will be solicited by CIFTRN. The default is TTY:

ISTEKO - Is Tektronics DVST graphics terminal output requested (T or F).

ISTRAC - Is trace output requested. Tracing will cause the entry to each symbol instantiation and the beginning of each box command to be listed. Such a mode is sometimes useful for debugging. Trace output goes to the same destination as does alphanumeric output, and the destination designator is specified as documented above under ISALFO.

A CIF file specifies the layout of objects in a Cartesian coordinate system of essentially unlimited size, with the unit of measure being taken as a centimicron. Some finite region of this layout space must be selected for mapping onto display devices, and this region is called the "window". The window position and size is set with the following parameters:

WINXL - Window X of left edge, units of centimicrons.

WINYB - Window Y of bottom edge, units of centimicrons.

WINDX - Delta X (width) of window. The height of the window is determined by the aspect ratio of the display device, a value of 0.7 being fairly typical.

All files used with the CIFTran system should be "unsequenced", i.e., line numbers should not be included in the files. The default for the "standard" editor EDIT is to include line numbers unless told otherwise by either the "EU" command or the initial value command EDIT/UNSEQUENCE in a file called SWITCH.INI. Most of the other editors on the system do not include line numbers and so they should work satisfactorily.

Since the plotters are fairly slow devices it is suggested that plotter output not be requested until syntax errors have been removed. Error diagnostics are reported even if no output forms are requested, and so a pass should be made over CIF files with no output requested to scan for syntax errors. Sometimes the alphanumeric output generated by either or both the ISTRAC and the ISALFO switches is helpful in localizing errors.

Following syntax error reports is a display of the "current line". Since line ends are not syntactically significant (except as token delimiters) the current line often appears to be the line following the offending line. For example, in the CIF code segment

```
(This is a comment)
B 1 2 3 4;
```

an error will be reported for the second line since at the beginning of the second line a ';' is needed to terminate the comment command.

#### Acknowledgment

---

Thanks go to Glen Miranker for writing the software which converts graphics commands into commands for the HP plotters.

## A Quick Guide to CIFTRN

The program CIFTRN is an enhanced version of CIFTRN. It has three additional features:

1. True windowing - with clipping
2. Selective plotting of layers
3. Plotting sorted by layer or color.

To use CIFTRN add the line

ADDPAR = 1

to your file CIFTRN.PAR  
between the lines WINDX = ... and the closing \$

That is:

. . . . .  
.

becomes

WINDX = ...	WINDX = ....
	ADDPAR = T
\$	\$

### 1. WINDOWING

With CIFTRN the origin (WINYB, WINXL) may be set to any point, and the window width (WINDX) may be set to any value. All features of your CIF file that fall in the rectangle defined by these three parameters will be plotted - filling the plotter platen.

2. When CIFTRN is run it will prompt you for additional parameters.  
Appropriate responses are:

(see over)

RED.	-	only red features will be plotted
BLUE.	-	" blue "
BLACK.	-	" black "
GREEN.	-	" green "
NIML.	-	" implant boxes will be plotted
NMET.	-	" metal "
NPOL.	-	" poly "
NDIF.	-	" diffusion "
NCUT.	-	" contact cuts will be plotted
NGLS.	-	" overglass cuts will be plotted
ALLS1.	-	all layers, sorted by color, will be plotted

When the layer (color) you specify has been plotted, you will again be prompted for additional parameters. If you are done type <CTRL-C>. If not, enter the next layer you want plotted.

Attached are two sample runs. Both plots of an output pad. The first run plotted only the poly and diffusion layers. The second run plotted all layers - sorted by color.

@CIFT11

CIFTRN 1.1

Enter CIF source designator: OUTPUTPADDEM.CIF

Enter additional parameters source designator.

Unit=21 DSK:LAYERS/ACCESS=SEQIN/MODE=ASCII

Enter new file specs. End with an \$(ALT)

\*NPOL,\$

\$DSPPAR

WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,

PLTLAY=T, T, 5\*X, \$

Overall bounding rectangle is:

XMin= 2100., XMax= 35700., YMin= 2100., YMax= 24600.

WAITING FOR PLOTTER TO FINISH . . .

Pass complete.

Enter additional parameters source designator.

Unit=21 DSK:LAYERS/ACCESS=SEQIN/MODE=ASCII

Enter new file specs. End with an \$(ALT)

\*NDIF,\$

\$DSPPAR

WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,

PLTLAY=T, 6\*X, \$

Overall bounding rectangle is:

XMin= 600., XMax= 35700., YMin= 300., YMax= 25500.

WAITING FOR PLOTTER TO FINISH . . .

Pass complete.

Enter additional parameters source designator.

Unit=21 DSK:LAYERS/ACCESS=SEQIN/MODE=ASCII

Enter new file specs. End with an \$(ALT)

\*^C

@CIFT11

CIFTRN 1.1

Enter CIF source designator: OUTPUTPADDEM.CIF

Enter additional parameters source designator.

Unit=21 DSK:LAYERS/ACCESS=SEQIN/MODE=ASCII

Enter new file specs. End with an \$(ALT)

\*ALLS1,\$

\$DSPPAR

WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,

PLTLAY=T, T, 5\*X, \$

Overall bounding rectangle is:

XMin= 2100., XMax= 35700., YMin= 2100., YMax= 24600.

WAITING FOR PLOTTER TO FINISH . . .

Pass complete.

\$DSPPAR

WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,

PLTLAY=T, 6\*X, \$

Overall bounding rectangle is:

XMin= 600., XMax= 35700., YMin= 300., YMax= 25500.

Pass complete.  
\$DSPPAR  
WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,  
PLTLAY=3\*F, T, 3\*F, \$  
Overall bounding rectangle is:  
XMin= 0., XMax= 37200., YMin= 0., YMax= 25800.  
WAITING FOR PLOTTER TO FINISH . . .  
Pass complete.  
\$DSPPAR  
WINXL= -4000.000 , WINYB= -3000.000 , WINDX= 50000.00  
ISTRAC=F, ISALFO=F, ISPLTO=T, ISTEKO=F, ADDPAR=T, ISLSEL=T,  
PLTLAY=2\*F, T, F, T, F, T, \$  
Overall bounding rectangle is:  
XMin= 900., XMax= 33900., YMin= 600., YMax= 25200.  
WAITING FOR PLOTTER TO FINISH . . .  
Pass complete.  
Enter additional parameters source designator.  
Unit=21 DSK:LAYERS/ACCESS=SEQIN/MODE=ASCII  
  
Enter new file specs. End with an \$(ALT)  
\*^C  
@^C  
@

M.I.T., Department of Electrical Engineering & Computer Science.

6.978. Questionnaire Regarding Project Testing:

Name: \_\_\_\_\_

Did your project get on the M.I.T. multi-project chip set? \_\_\_\_\_

If yes, are you interested in participating in project testing during IAP, assuming the wafers are fabricated in time? \_\_\_\_\_

Or are you planning to do your project testing on your own? \_\_\_\_\_

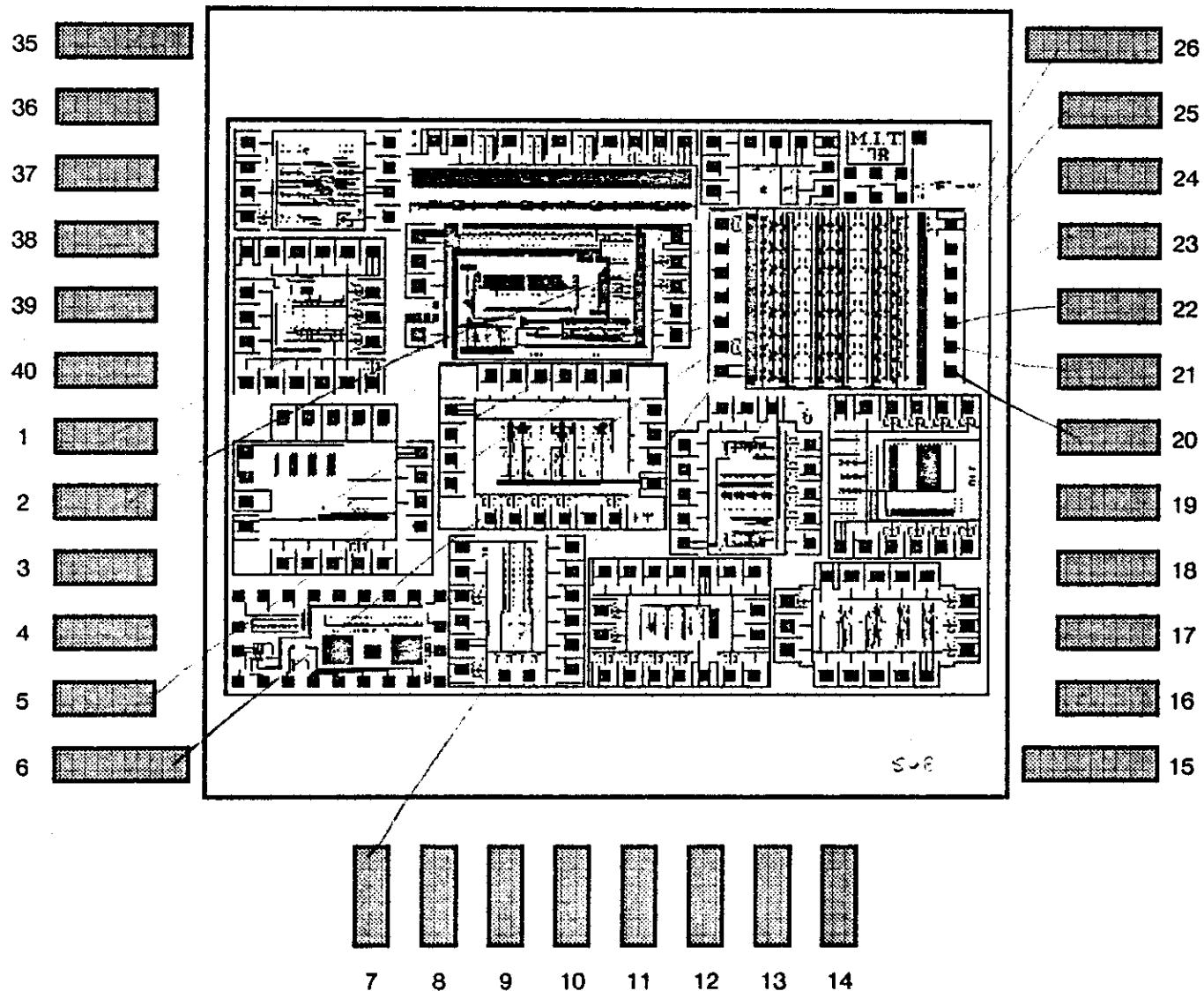
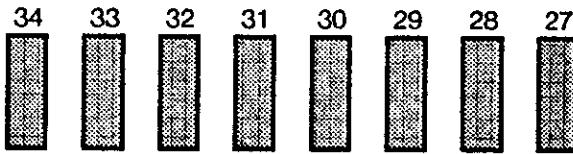
If you are interested in participating in project testing at M.I.T. during IAP, where can you be reached during January:

Address(es) :

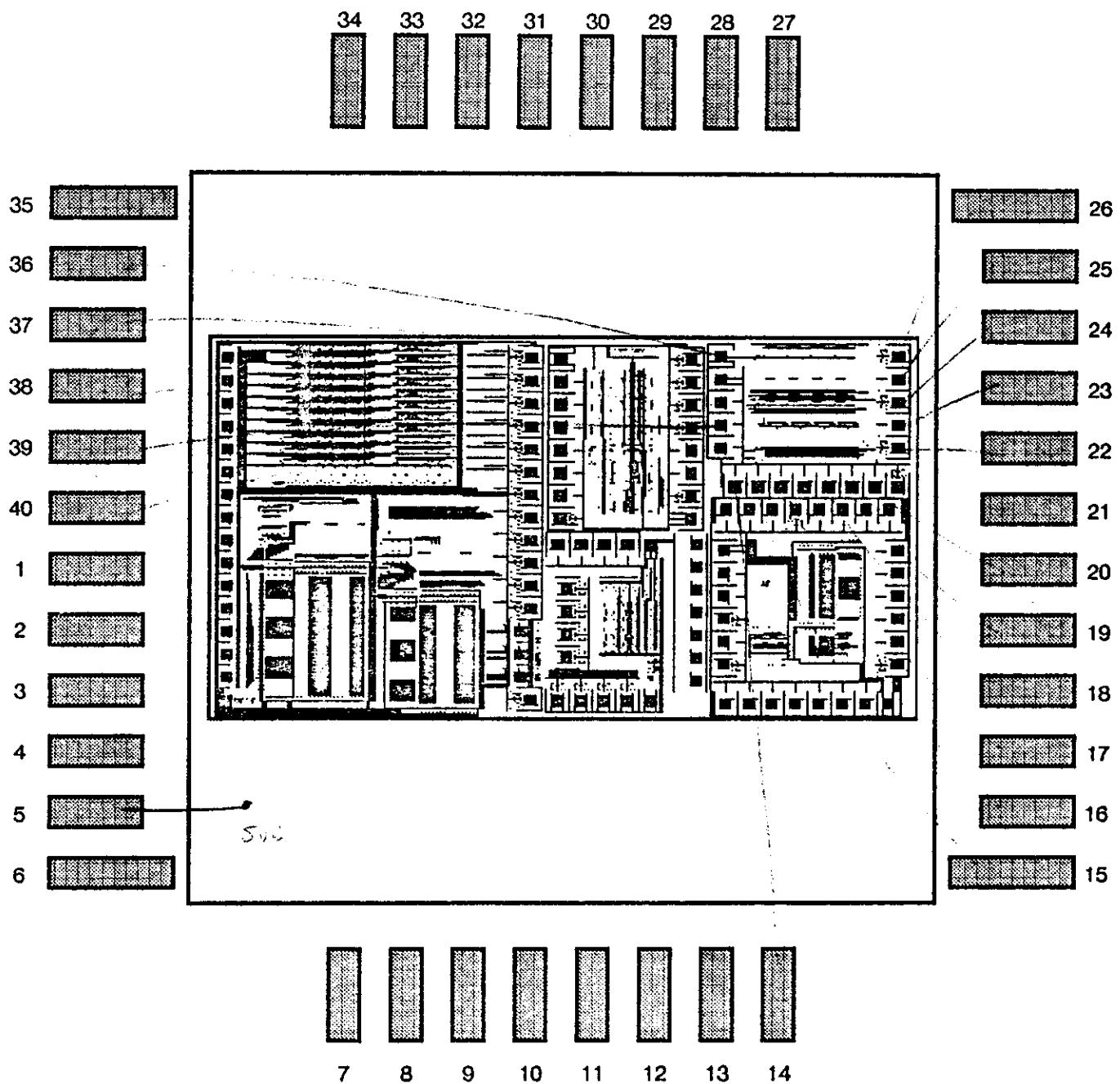
Phones:

If you are interested in receiving additional artifacts of the M.I.T. project set as they become available (project plots, both Versatec and solid color, unmounted chips, etc.), give your address for the period Feb. - May '79:

Sample Bonding Map  
(Cherry)



Sample Banding Map  
(Shaver)



\*\* COURSE SIX SUBJECT EVALUATION FALL 1978 \*\*

Instructions: Please answer questions with whole numbers between 1 and 5. If a question is not applicable, answer with a zero. These questionnaires will be reviewed by the faculty, and the results will be published next term.

Subject Number: \_\_\_\_\_

Rec. Instructor: \_\_\_\_\_

T.A.: \_\_\_\_\_

- 1) What is your overall rating of this subject?  
(5 = outstanding, ... , 1 = poor)

1) \_\_\_\_\_

How much did you learn from each of the following?

(5 = a great deal, ... , 1 = very little, 0 = not applicable)

- 2) Lectures 2) \_\_\_\_\_  
3) Recitations 3) \_\_\_\_\_  
4) Tutorials 4) \_\_\_\_\_  
5) Text books 5) \_\_\_\_\_  
6) Class notes 6) \_\_\_\_\_  
7) Problem Sets and Homework Assignments 7) \_\_\_\_\_  
8) Laboratory 8) \_\_\_\_\_  
9) Tests and Quizzes 9) \_\_\_\_\_

How would you rate the overall effectiveness of these faculty members?

(5 = outstanding, ... , 1 = poor)

- 10) Lecturer 10) \_\_\_\_\_  
11) Recitation Instructor 11) \_\_\_\_\_  
12) Teaching Assistant 12) \_\_\_\_\_  
13) How easy was it to get help from the appropriate faculty members? 13) \_\_\_\_\_  
(5 = very easy, ... , 1 = very difficult, 0 = not applicable)

- 14) In general, how hard was the course material for you? 14) \_\_\_\_\_  
(5 = very hard, ... , 1 = very easy)

- 15) How was the overall pace of the course? 15) \_\_\_\_\_  
(5 = too fast, ... , 1 = too slow)  
In general, homework assignments were:  
16) (5 = too long, ... , 1 = too short) 16) \_\_\_\_\_  
17) (5 = too many, ... , 1 = too few) 17) \_\_\_\_\_  
18) (5 = too hard, ... , 1 = too easy) 18) \_\_\_\_\_

In general, tests and quizzes were:

- 19) (5 = too long, ... , 1 = too short) 19) \_\_\_\_\_  
20) (5 = too hard, ... , 1 = too easy) 20) \_\_\_\_\_

In general, laboratory assignments were:

- 21) (5 = too long, ... , 1 = too short, 0 = not applicable) 21) \_\_\_\_\_  
22) (5 = too hard, ... , 1 = too easy, 0 = not applicable) 22) \_\_\_\_\_

Laboratory facilities were:

23) (5 = excellent, . . . , 1 = deplorable)

23) \_\_\_\_\_

24) On the average, how many total hours (both inside and outside the classroom) per week do you spend on this subject? (Round to the nearest hour, please.)

24) \_\_\_\_\_

Blank questions: Please complete this section only if your instructor provides you with specific questions to be answered.

25) \_\_\_\_\_

26) \_\_\_\_\_

27) \_\_\_\_\_

28) \_\_\_\_\_

Comments: Please use the space below to make additional comments about the course. You may wish to elaborate on your answers to the questions above. PLEASE remember that your individual comments are a vital part of the Subject Evaluation process.