6.978. LECTURE #8: OCTOBER 5

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

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

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

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

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

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

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

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

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

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

  WHICH CAN BE REPEATED TO BUILD UP THE LAYOUT.

O.K. First lets review PLA design:

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

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

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

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

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

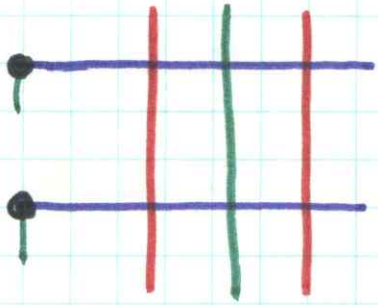
  Rather than just attacking the whole thing at once!

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

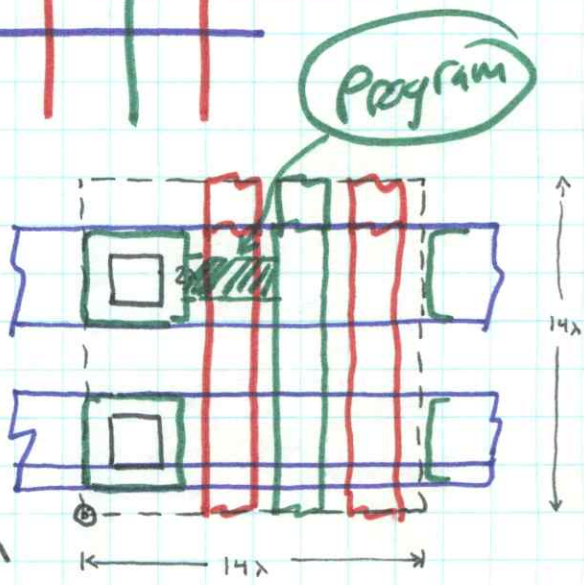
[THE PLANES WILL OCCUPY MOST OF THE AREA OF A LARGE PLA, EVEN IF THE PULLUPS, DRIVERS, etc., are not minimized.]

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

(TILT FOR "OR")



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

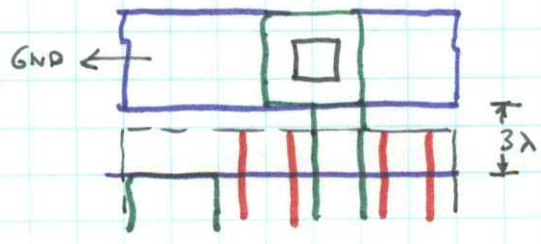


- PLACE DIFF FLASHES, DETERMINES PITCH OF CELL.
- So weve layed out a "PAIR of PLA Cells" For the AND/OR Planes

ITS SQUARE!

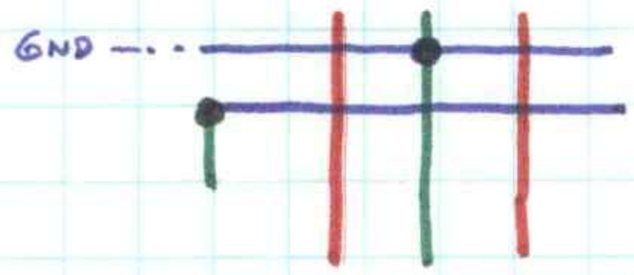
- WHAT ABOUT GND Return at edges of PLANES:

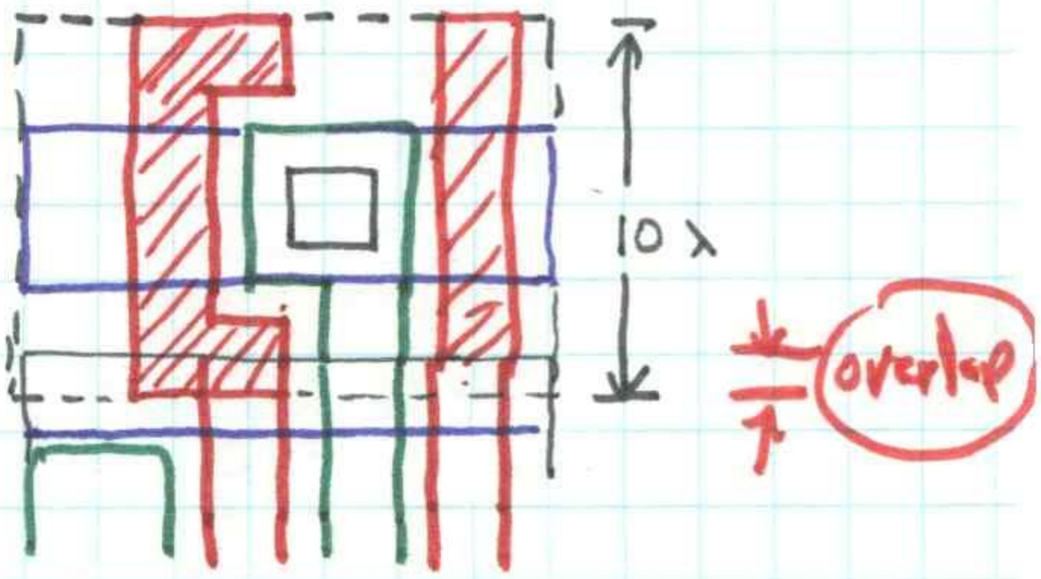
Could just use:



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

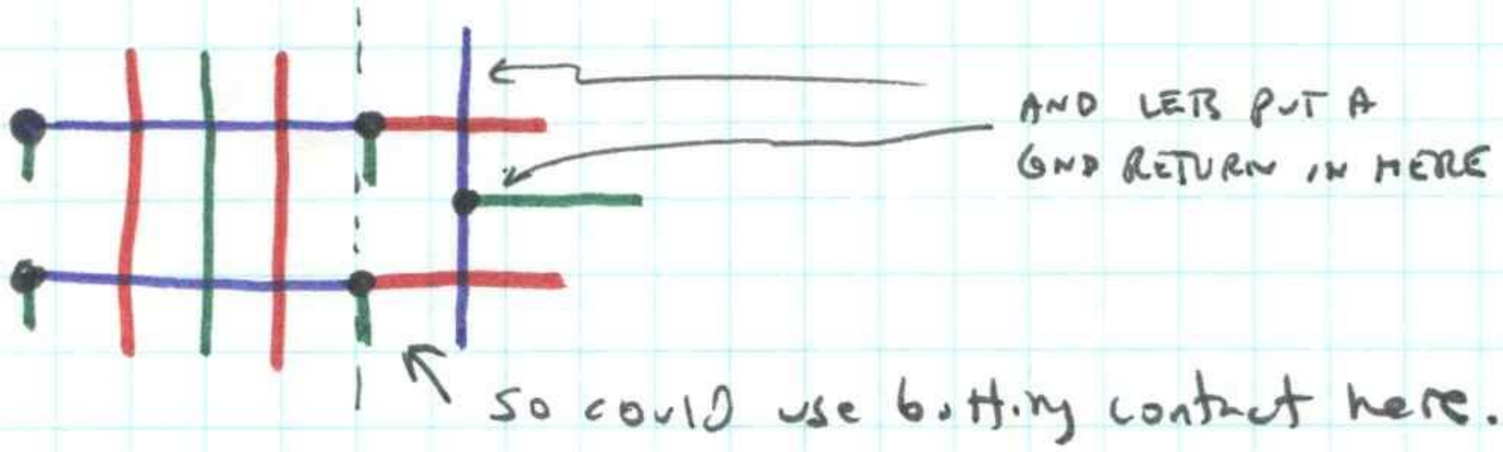
LETS MAKE A CELL THAT WE COULD USE AT INTERVELS, INSERTED AS ROWS, IN "AND" PLANE (Cols in "OR" plane).



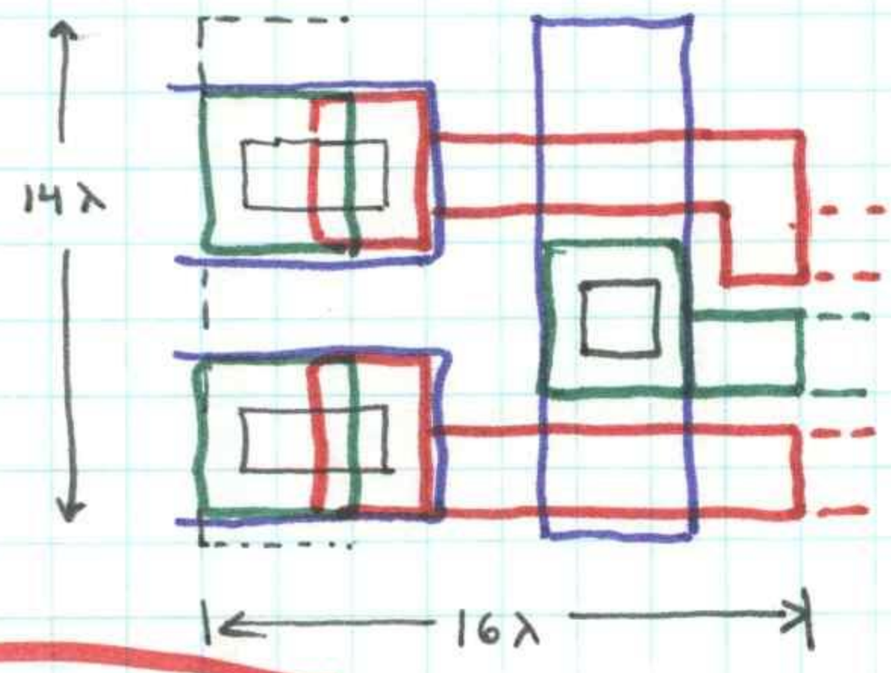


- LETS DESIGN CELLS TO CONNECT THE PLANES:

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

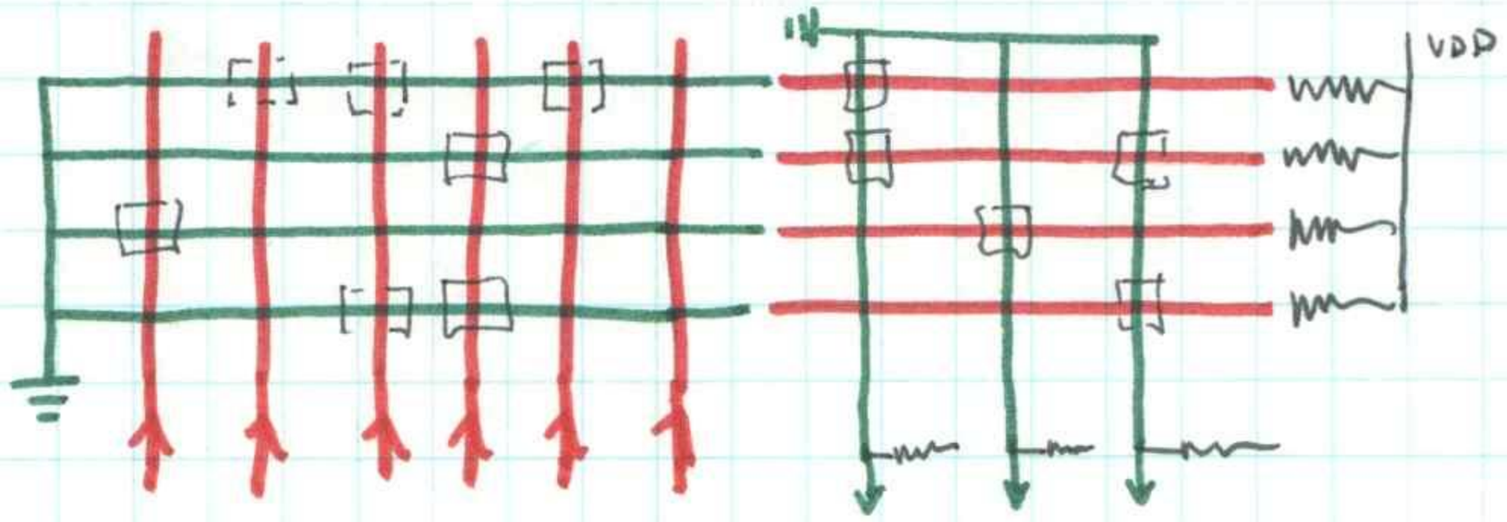


WHILE SEEM SIMPLE, NOTE, THESE ARE RATHER TRICKY, MINIMUM LAYOUTS

SHOW SLIDES CELLS, OVERALL, MANIPULATE CELLS

(COVER IF TIME)

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



- What would be advantages of this type of PLA?

  (much smaller area)
  except watch out for pullups!

- what would be disadvantages of NAND-NAND PLA?

  > much slower

  > speed is a fcn of size

  > WORST: PULLUP SIZE IS a function of SIZE!
  (so can't use same cell for all such PLAs)

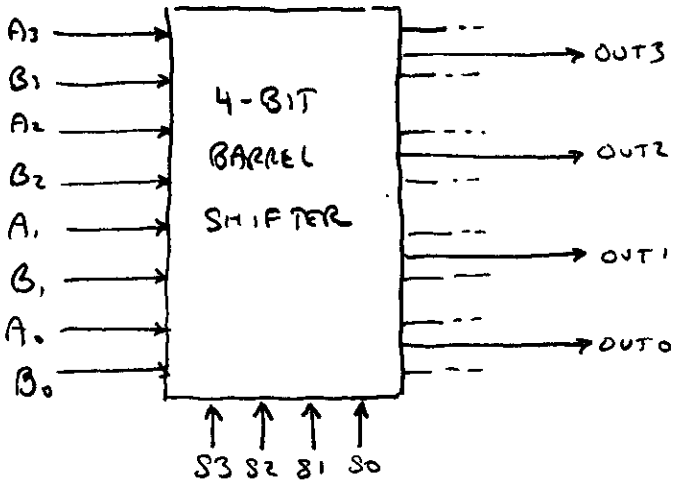
MORAL: DON'T USE NAND-NAND PLA UNLESS IT IS SMALL, YOU ARE TIGHT FOR SPACE, AND YOU CHECK YOUR DELAYS & PULLUP/PULLDOWN RATIOS CAREFULLY.

THE BARREL SHIFTER

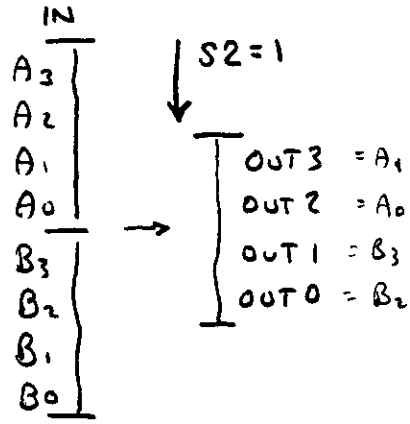
- ONLY C/L, in fact can be done with a switch array, but is so useful and so clever, that I consider it an important "SUBSYSTEM".
- This maps nicely into silicon, and is kludgey using purchased parts.

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

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

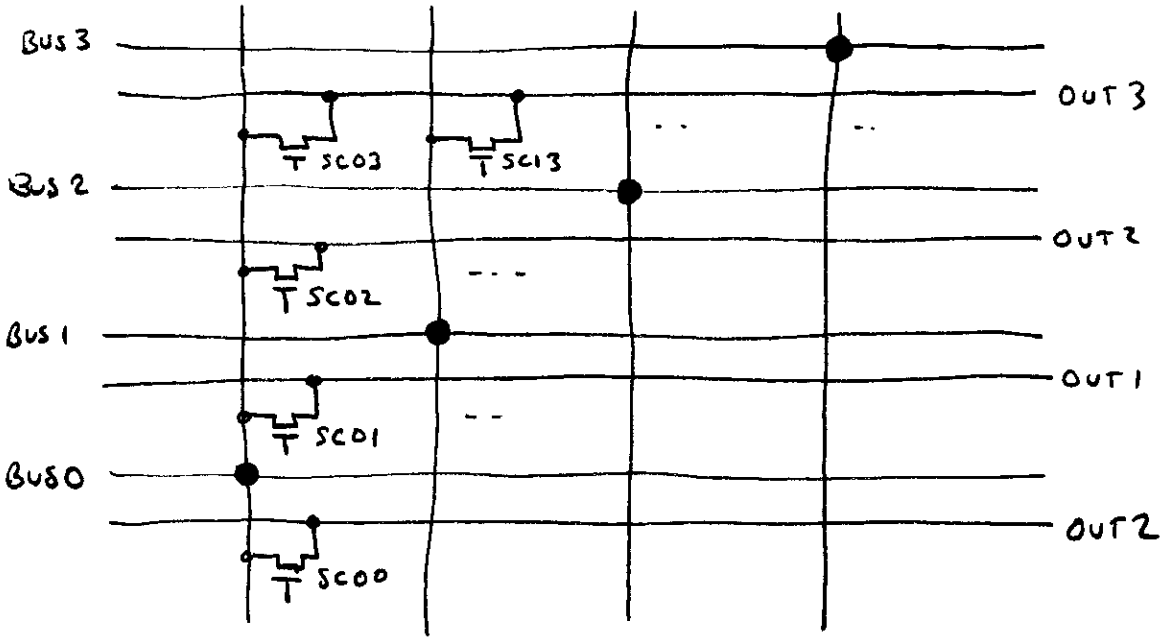


CONCEPTUAL PICTURE OF FUNCTION:



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

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



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

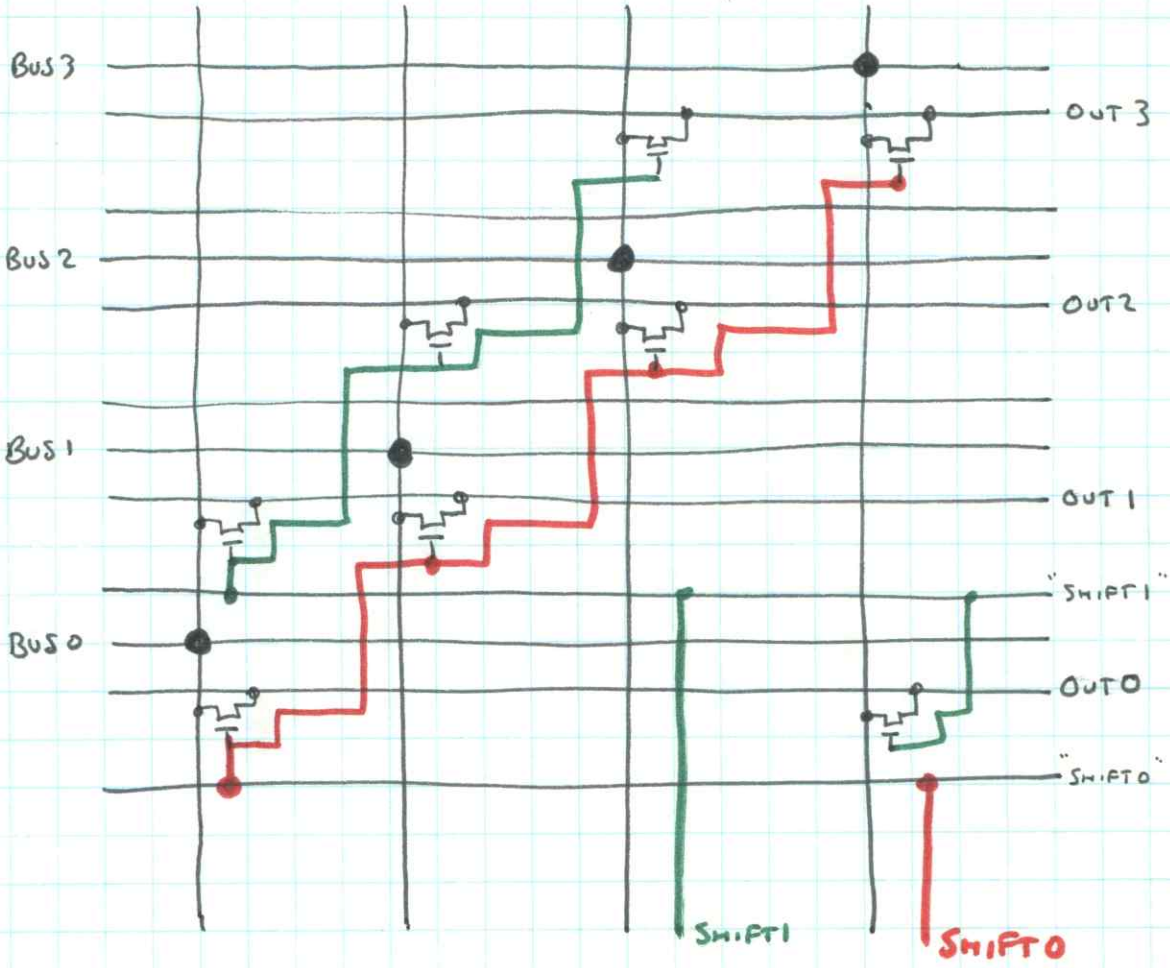
- WE COULD DO ALL SORTS OF SHIFTING, INTERCHANGING WITH THIS STRUCTURE.

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

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

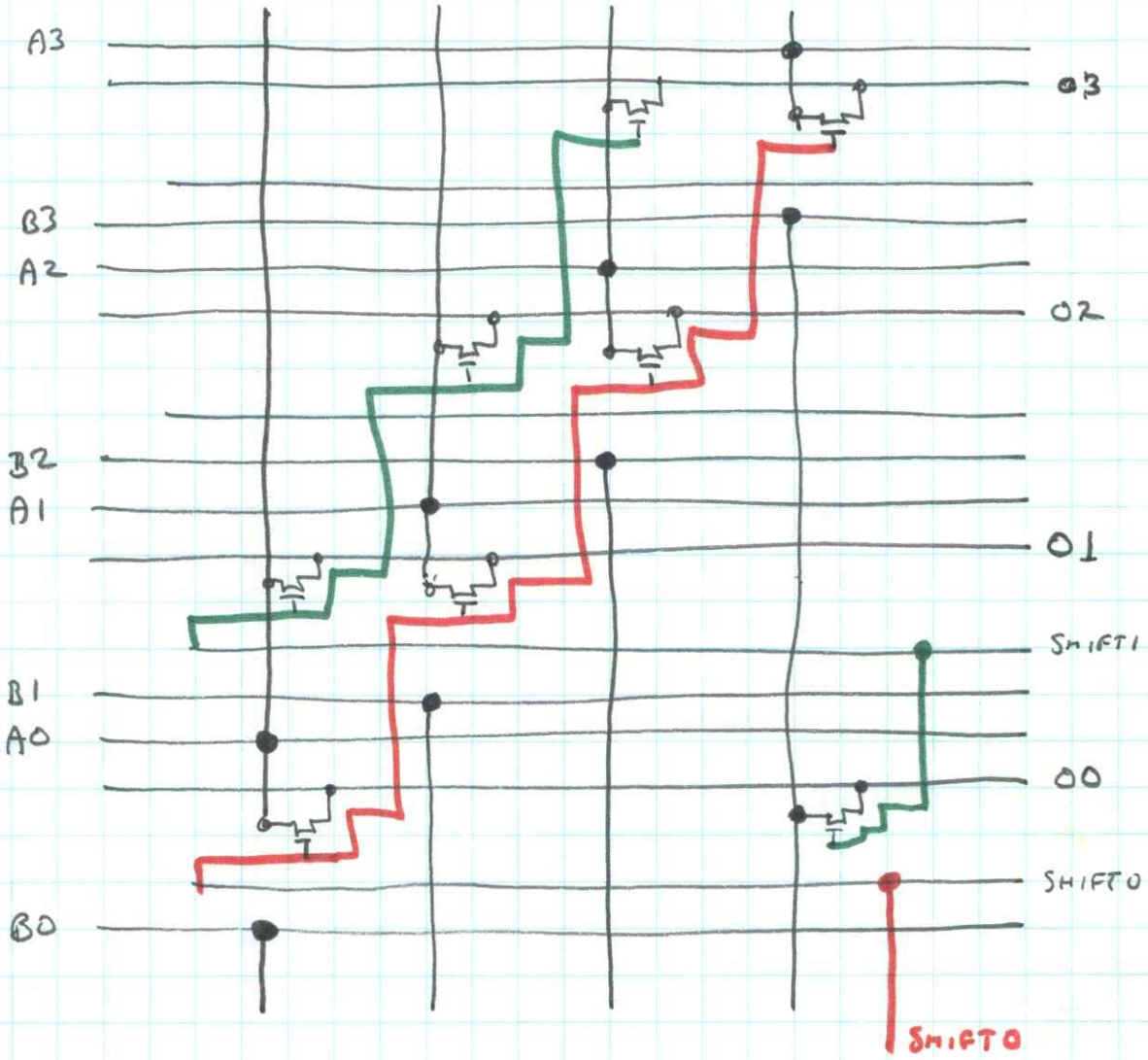
(cont.)

DRAW SAME DIAGRAM: WITH A FEW MORE LINES:



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

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



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

, etc.

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

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

  ? ~~Hmm looks Strange~~ ! OK!

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

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

LAYOUT NOT MIN-ON PITCH OF ALU

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

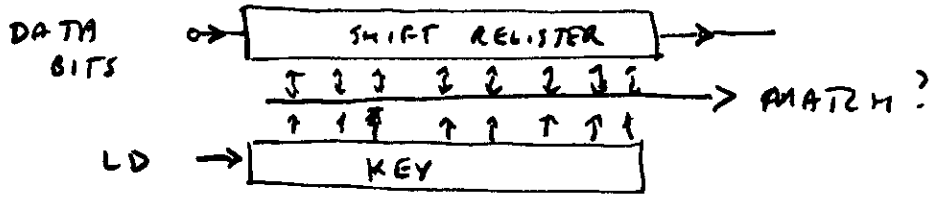
(A COUPLE OF MONTHS AGO I TALKED WITH BILL LATTIN WHO HEADS UP INTELS NEW ARCHITECTURE GROUP IN OREGON)--

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

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

A SERIAL BIT STREAM COMPARATOR:

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

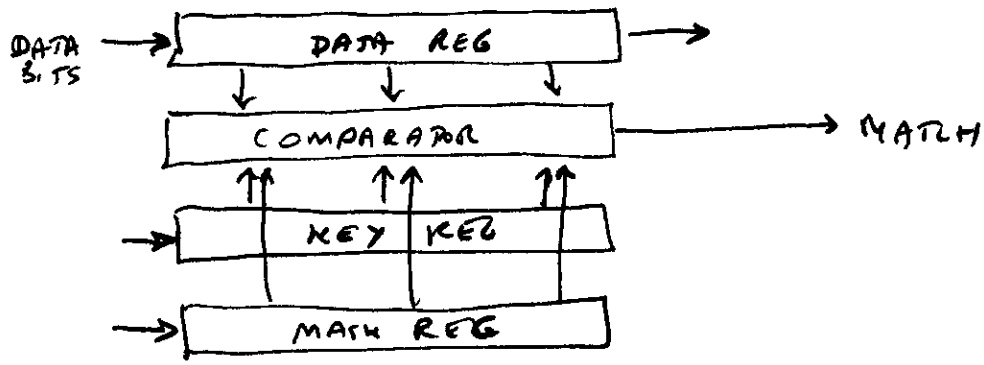


- MIGHT THINK OF LOADING A KEY STRING INTO A REGISTER, AND RUNNING DATA THRU A REG. NEXT TO IT AND SOMEHOW COMPARING ALL THE BITS, WHEN THEY ALL MATCH → GET A TRUE MATCH OUTPUT.

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

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

So:

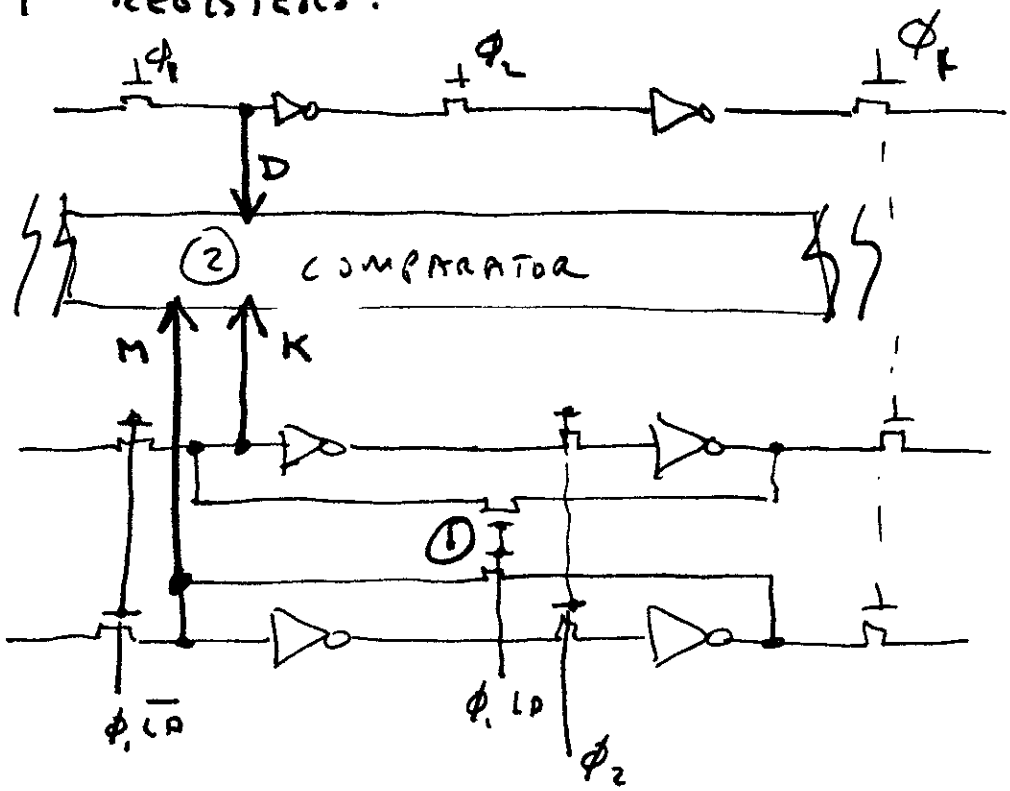


I.E., ONLY TRY TO MATCH THE SUBSET MARKED BY THE MASK BITS.

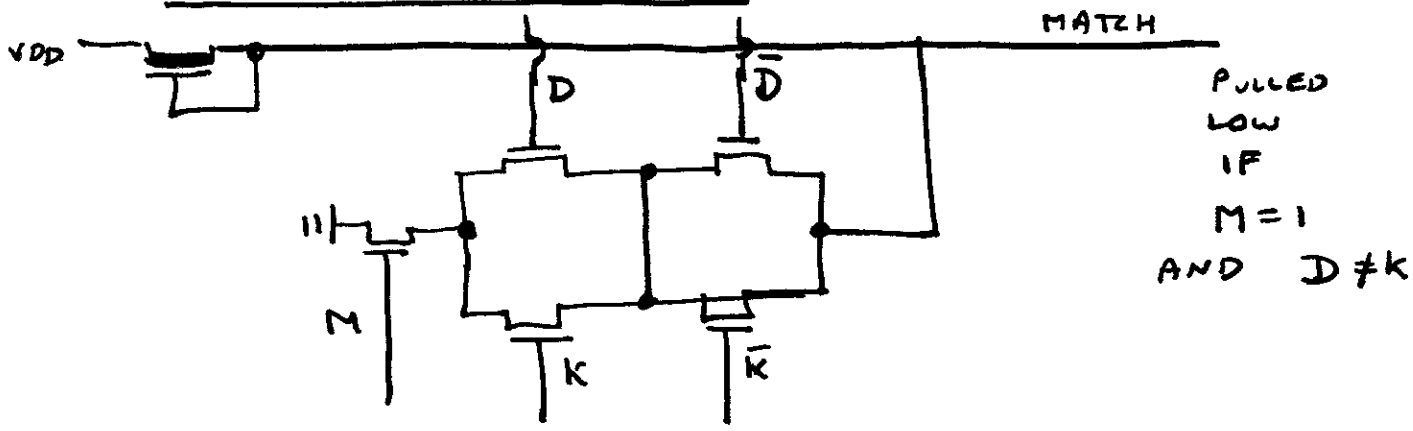
HOW WOULD WE DESIGN THIS:

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

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

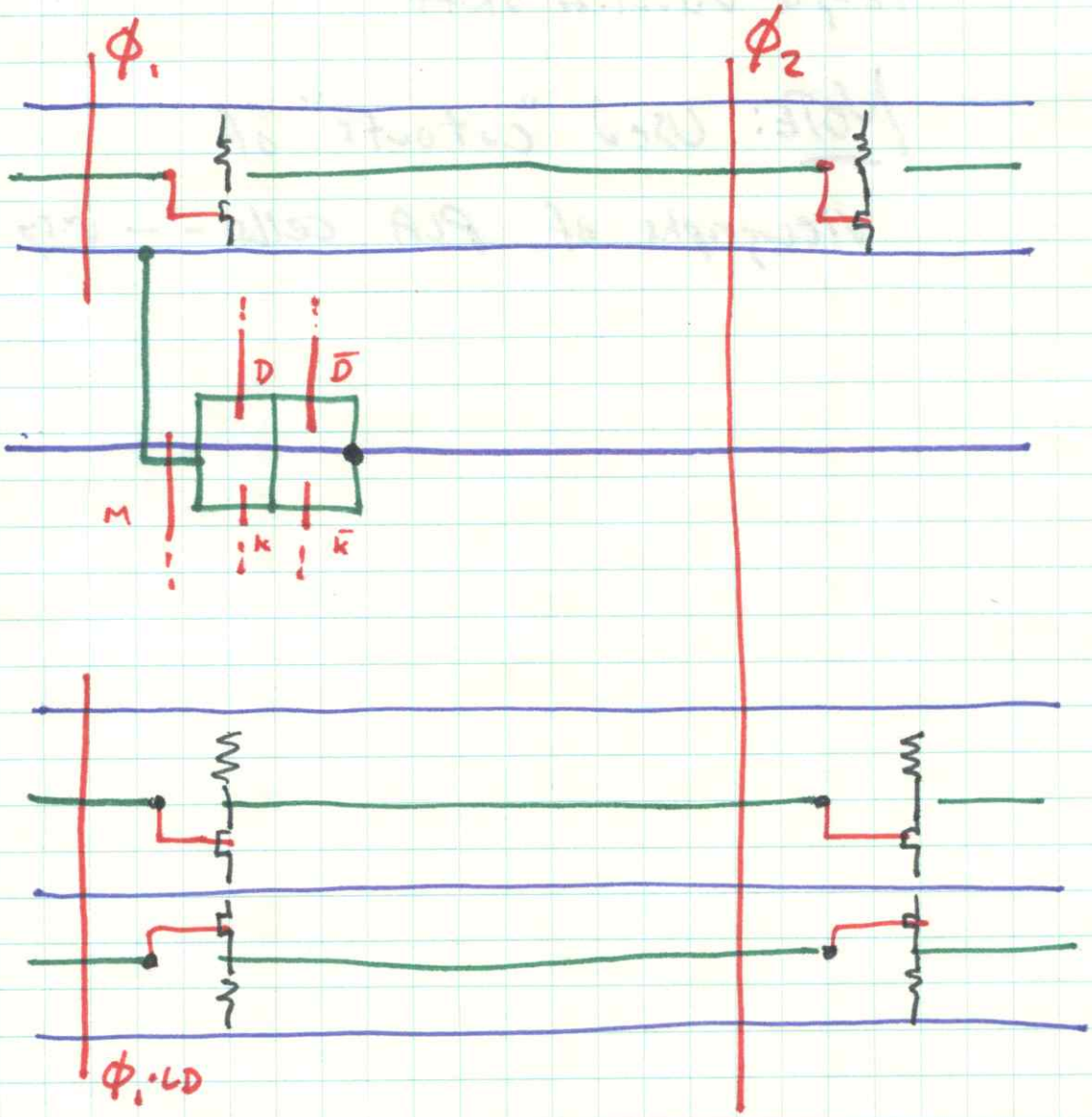


- MAKE LOWER TWO LOADABLE (1)
- BRING MASK, DATA, KEY TO COMPARATOR (2)
- NOW DESIGN COMPARATOR:



HAND OUT H.W. #4

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



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