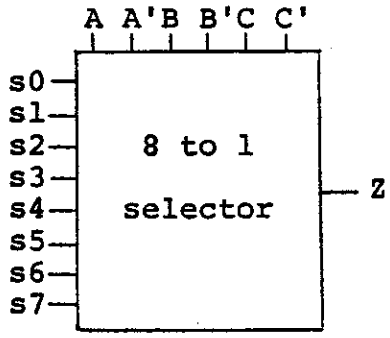
IV. Homework and Project Assignments

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:

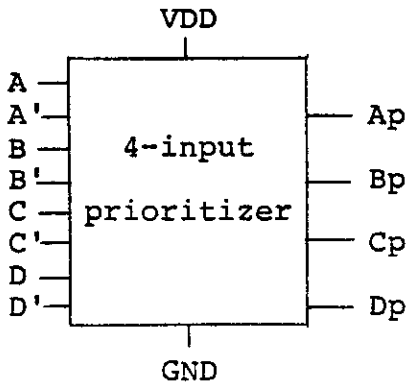


truth table:

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

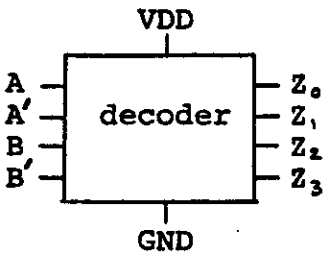
Chap3: Introduction,
Notation,
Combinational Logic

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_0	Z_1	Z_2	Z_3
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)

A	B	Z_0	Z_1	Z_2	Z_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

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:

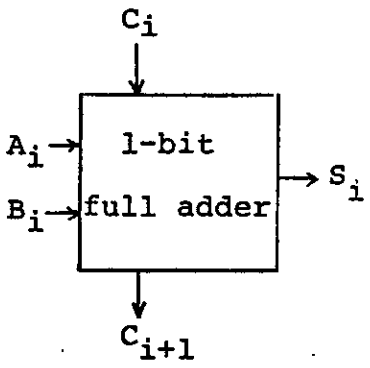
Chap3: Two Phase Clocks, The Shift Register, Relating Different Levels of Abstraction, Implementing Dynamic Registers, Designing a Subsystem.

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

6.978. Homework #2 (continued):

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

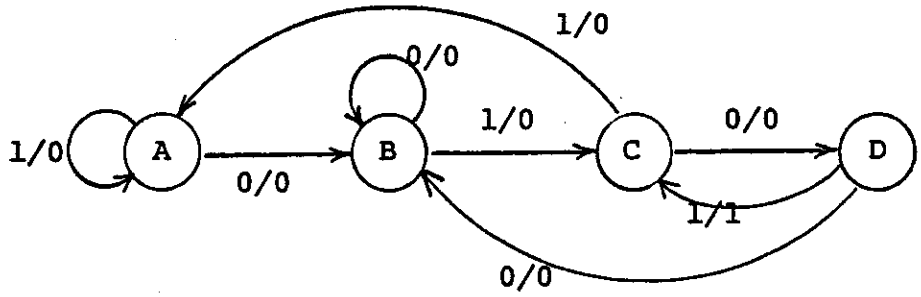
7. Construct the stick diagram for an nMOS PLA which implements a one-bit stage of a full adder. Include the phase-1 and phase-2 clocked registers which precede and follow the PLA logic (O.K. to use logic symbols for the inverters). Clearly indicate the topology of the carry-in signal path and the carry-out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



C_i	A_i	B_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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	✗C	1	success,output 1, and go to ✗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 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 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 lambda^2 = 3591μm^2. ('78 lambda = 3μm).

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 Zpu:Zpd = 8:1.

Create a more compact layout which still satifies 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 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 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

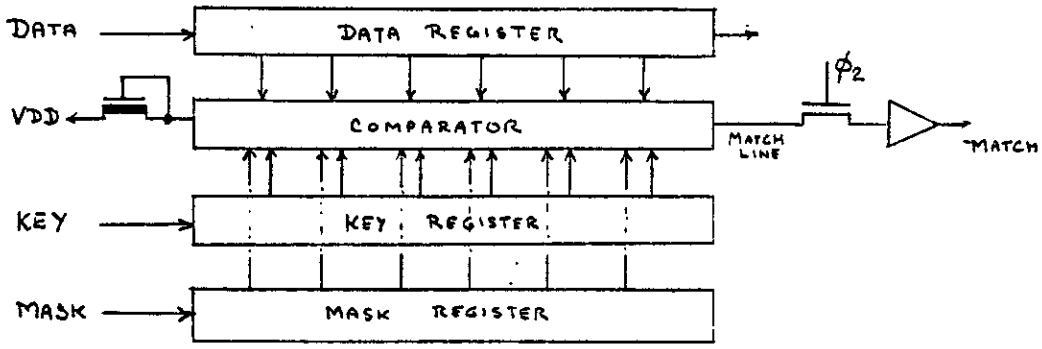


Figure 1. A Serial Bit String Comparator Subsystem. (Block Diagram)

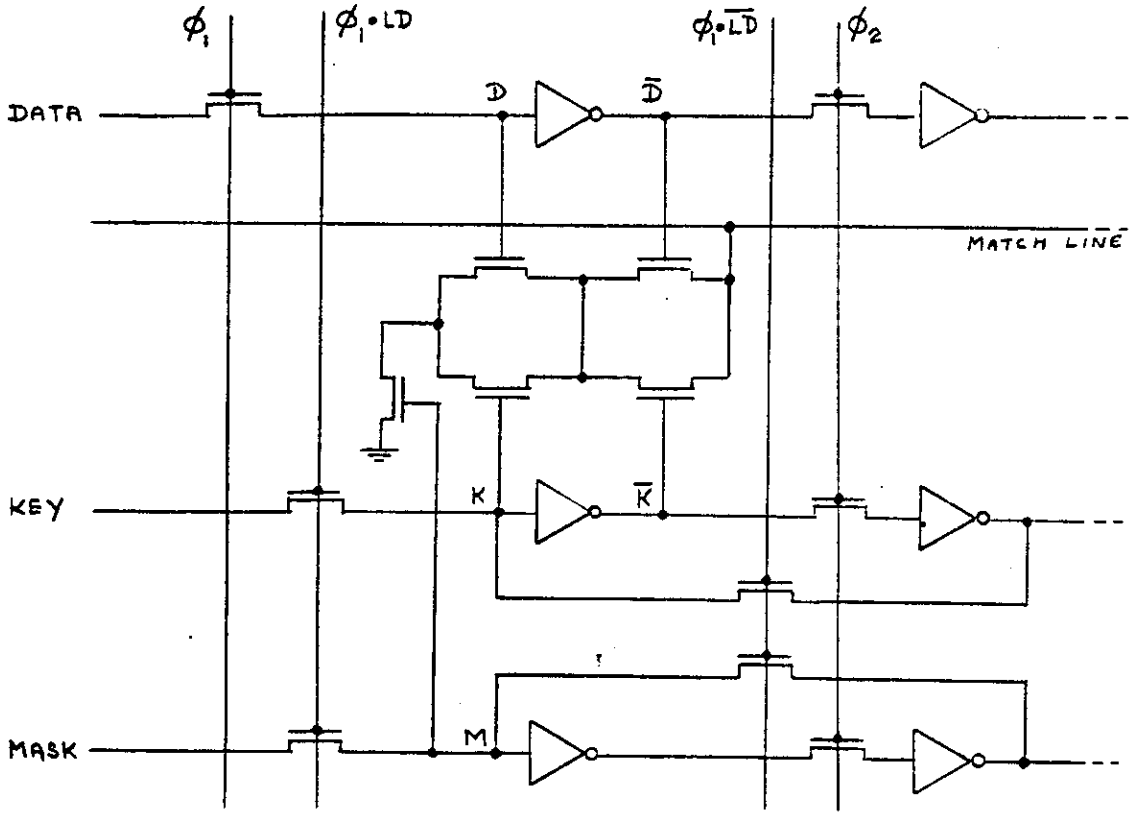


Figure 2. A One-Bit Vertical Slice thru One Possible nMOS Circuit/Logic design of the Serial Bit-String Comparator.

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.

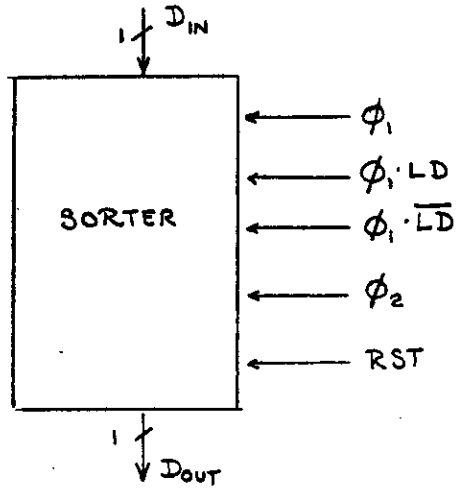
13. Construct a CIF-code description of the layout of the two dual-port register cells pictured in Figure 22a in chapter 5 of the text. However, modify the layout as needed to eliminate the use of boxes at other than direction (1 0).

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.

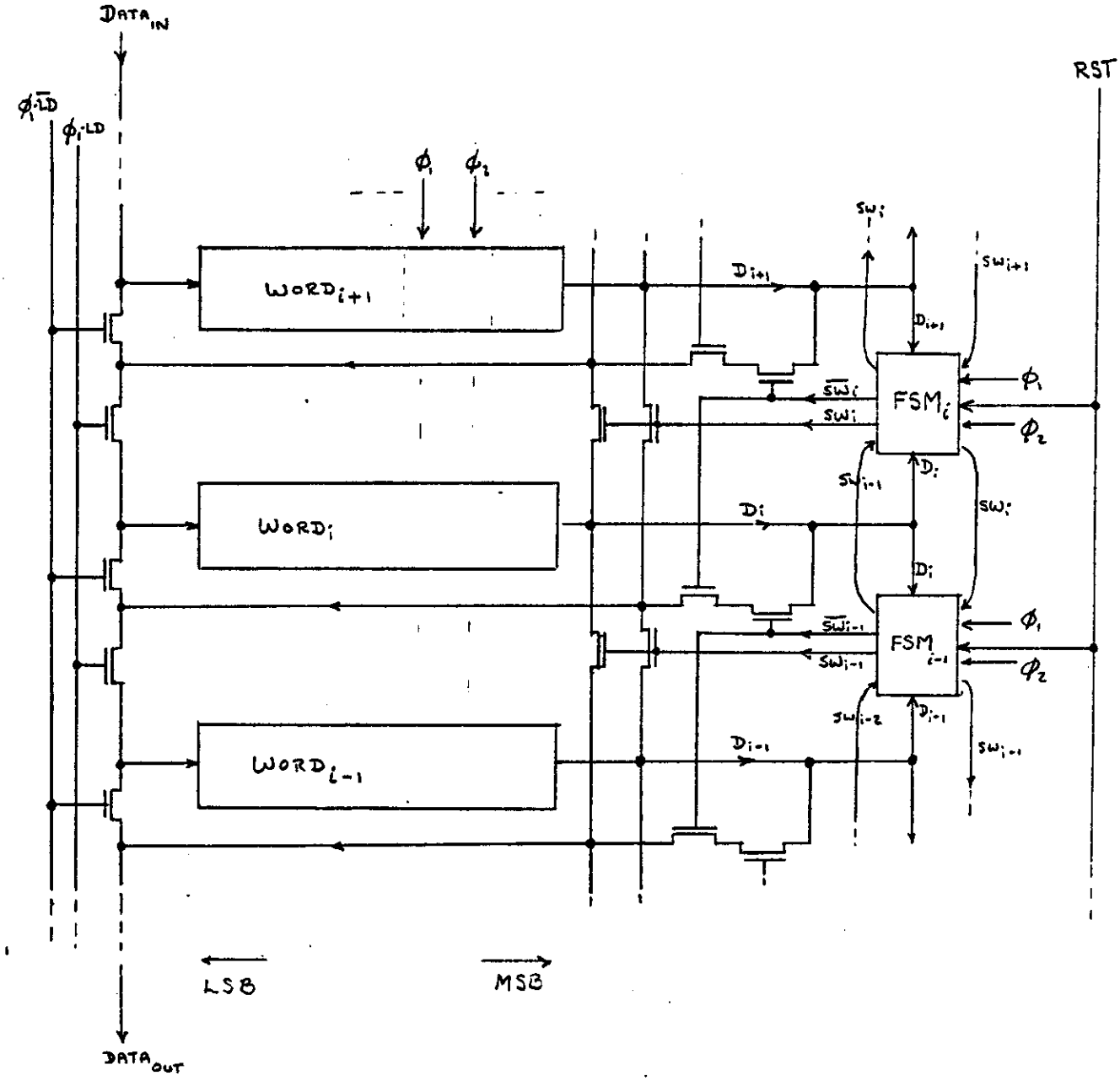
The sorter memory array contains a processing element between each adjacent pair of m bit words. If properly organized, the loaded sorter can perform a complete bubble sort in (N)x(m) phase-1/phase-2 cycles. It can thus sort N words, using a memory space of N words, in time of order N.

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 swappings 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.



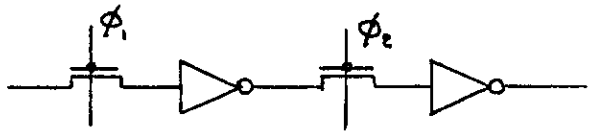
6.978. Homework #5. Problem 14 (cont.).

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 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.

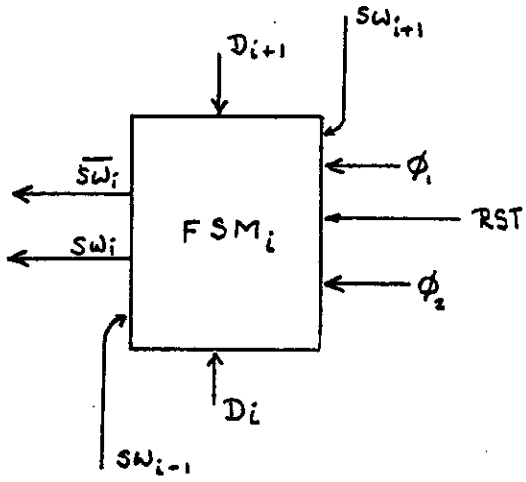


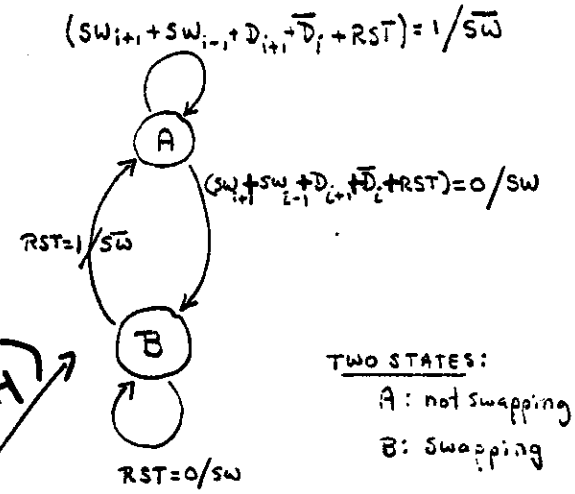
6.978. Homework #5. Problem 14 (cont.).

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 "$swap_i$". The state diagram for FSM_i is also given below.





(Too T;2ky)

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 state 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	$\lambda = 3\ \mu m$

Dimensions and Replication

InverterPair	X: 41λ	Y: 16λ	DX: no	DY: no
Back...Pair	X: 37λ	Y: 16λ	DX: no	DY: no
InverterQuad	X: 41λ	Y: 28λ	DX: no	DY: 30λ
Back...Quad	X: 37λ	Y: 28λ	DX: no	DY: 30λ

Further Info See section 7.3 of *A Guide to LSI Implementation.*

Function/Use Produces the inverted and non-inverted value of a signal that has been routed though 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 avialible 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.

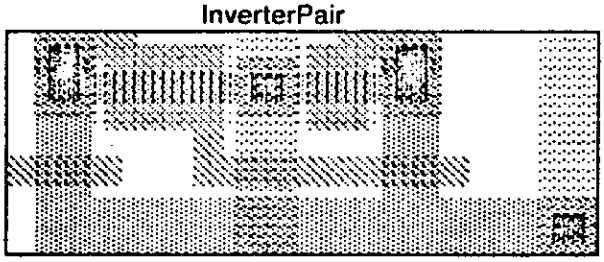
Connections Vertical Vdd, Ground, and clocks in metal 4λ wide. Input is on the left (right) side in red 2λ wide. Both the true and complement output are avialible in red and green on the right (left) side.

Included Cells None.

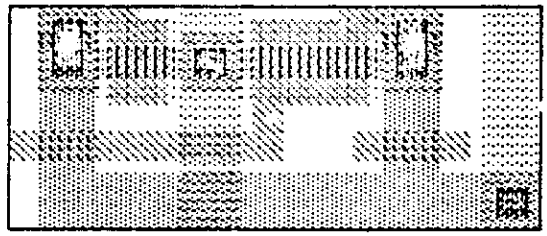
Loadings Input load: $8\lambda^2$ gate = .03 pf (fanout of 1)
The complement output drives the second inverter, so it already has a load of 1 fanout.

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: $\sim(20k\Omega/4)\cdot(C_{load}+.03pf)$ ~5ns/pf.
Internal delay: output rises $\sim 8(f_{comp}+1)\tau$, output falls $\sim(f_{comp}+1)\tau$

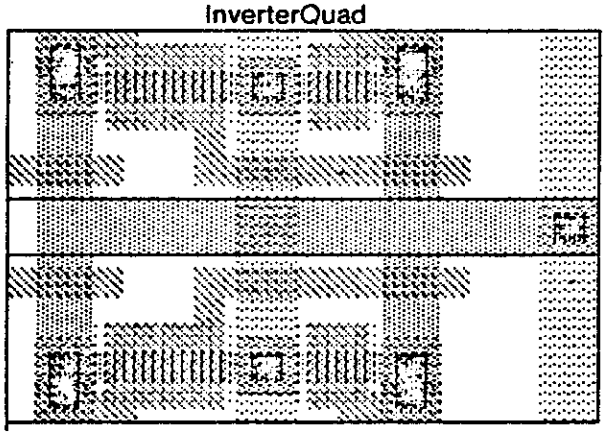
InverterPair



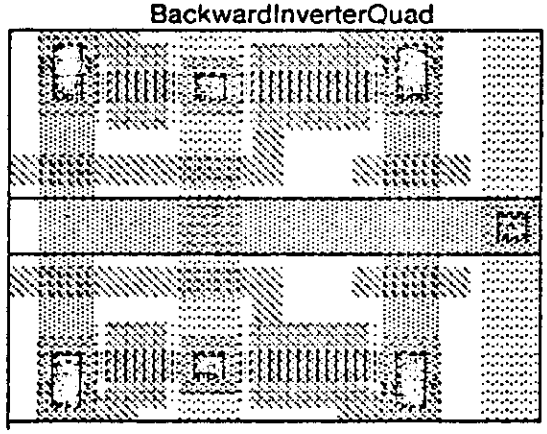
BackwardInverterPair



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 1700  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 1700  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-MAXC
Subject: CIF 2.0 bugs
To: Conway, Sproull, Hon@cmub, 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.

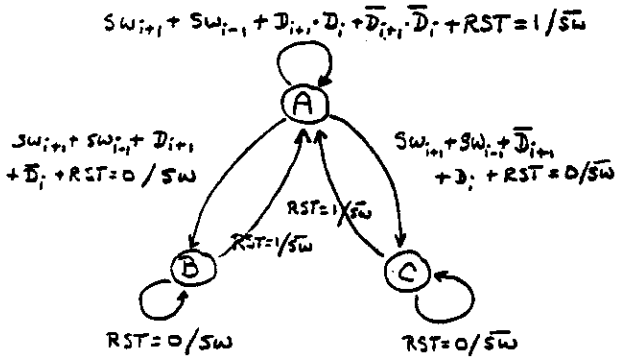
Credit goes to Bill Henke of MIT for reporting most of these bugs.

Dick

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

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 FSM_i is as follows:



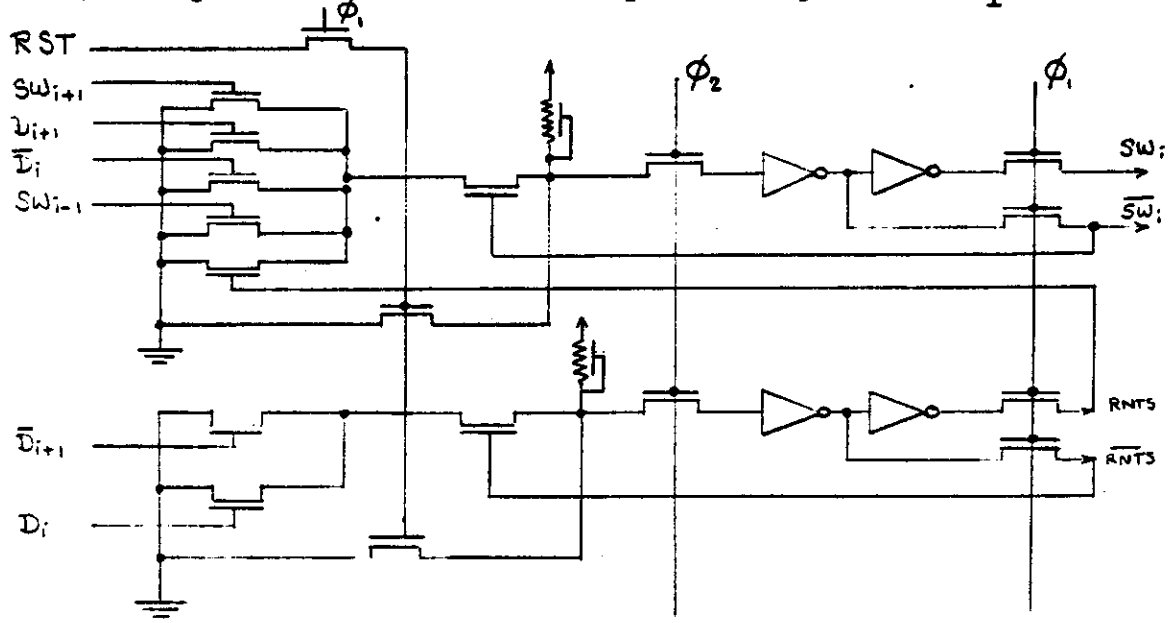
STATES: A : NOT YET SWAPPING/RESET
B : SWAPPING
C : REMEMBER NOT TO SWAP

SW	RNTS
0	0
1	X
0	1

IF $SW_{i+1} + SW_{i-1} + RST = 1$, STAY IN (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)

Now, one possible MOS circuit implementing this FSM_i is:



Starting with the above circuit, and with the subsystem diagram given in HW #5, page 2, construct a stick diagram of 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 chosing, 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 chosing, 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.

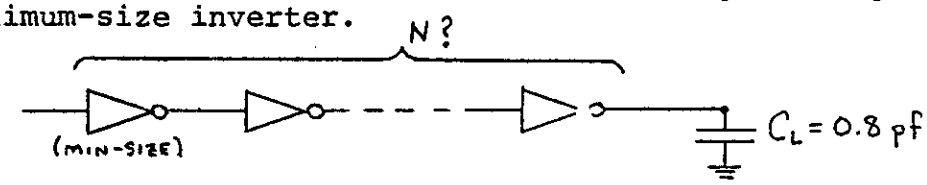
Note: Projects will be selected during the period 21 to 28 Nov., for inclusion in the M.I.T. multi-project chip set. Final design files will be merged into the chip file on 5 Dec.

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?

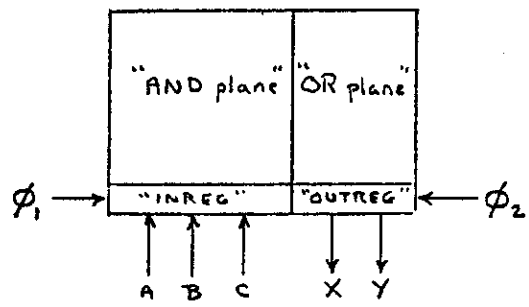
(b) How wide (in microns) should the VDD and GND wires be which supply power to the OM-2 Register Array Subsystem pictured in the frontispiece in the text? The detailed layout of each pair of cells is shown in Fig. 22a, Chap. 5. Assume that metal wires are 1 micron in height, and use a maximum current density of 1 ma per micron^2. 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.

$$X = (A + \overline{B})(\overline{A} + B + C)(B + \overline{C})$$

$$Y = (A + C)(\overline{A} + B + C)$$



6.978. Midterm Exam (cont.)

Problem 3. (25%)

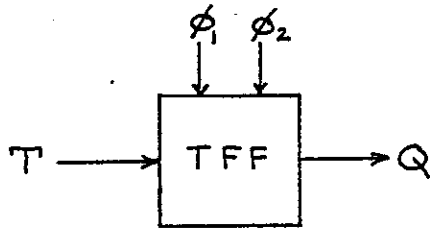
(a) Assume that all lines in Fig. 4b, Chap. 5, are of minimum width. What is the design rule violated in this figure?

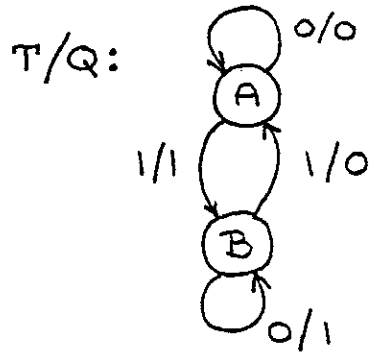
(b) What is the major error in the design in Fig. 17, Chap. 4.

(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 translater 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 ϕ_1-ϕ_2 clock cycle) if its input T is high during ϕ_1. If T is low during ϕ_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!).