Problem 1: Inside the SRAM

The figure below shows a somewhat idealized version of what's inside your SRAM. An address decoder turns your N input lines into 2N row select lines, thereby selecting one row out of the storage array (1 bit in this case, 8 bits in your project SRAM). A row that is selected is either gated onto the data lines (when you are reading) or is forced into the state specified by the data lines (when you are writing). Physically, this selection is accomplished by a pair of tri-state buffers. Each of the blocks in the figure has some timing issues that need to be respected for the RAM to opperate correctly. These include:

$\mathbf{t}_{DEC}$ - new address stable to row select line stable

$\mathbf{t}_{ACCESS}$ - row select lines stable until internal data line(s) stable (during read)

$\mathbf{t}_{WRITE}$ - minimum time that internal data line(s) need to be stable to correctly write into the bit array (during a write, obviously).

$\mathbf{t}_{AND}$ - AND gate delay (for CS, and WE/OE products)

$\mathbf{t}_{BUF}$ - delay through an enabled tri-state buffer

$\mathbf{t}_{EN}$ - enable time for a tri-state buffer

Using these internal times, calculate some of the key timing consirations that you as the RAM user need to consider:

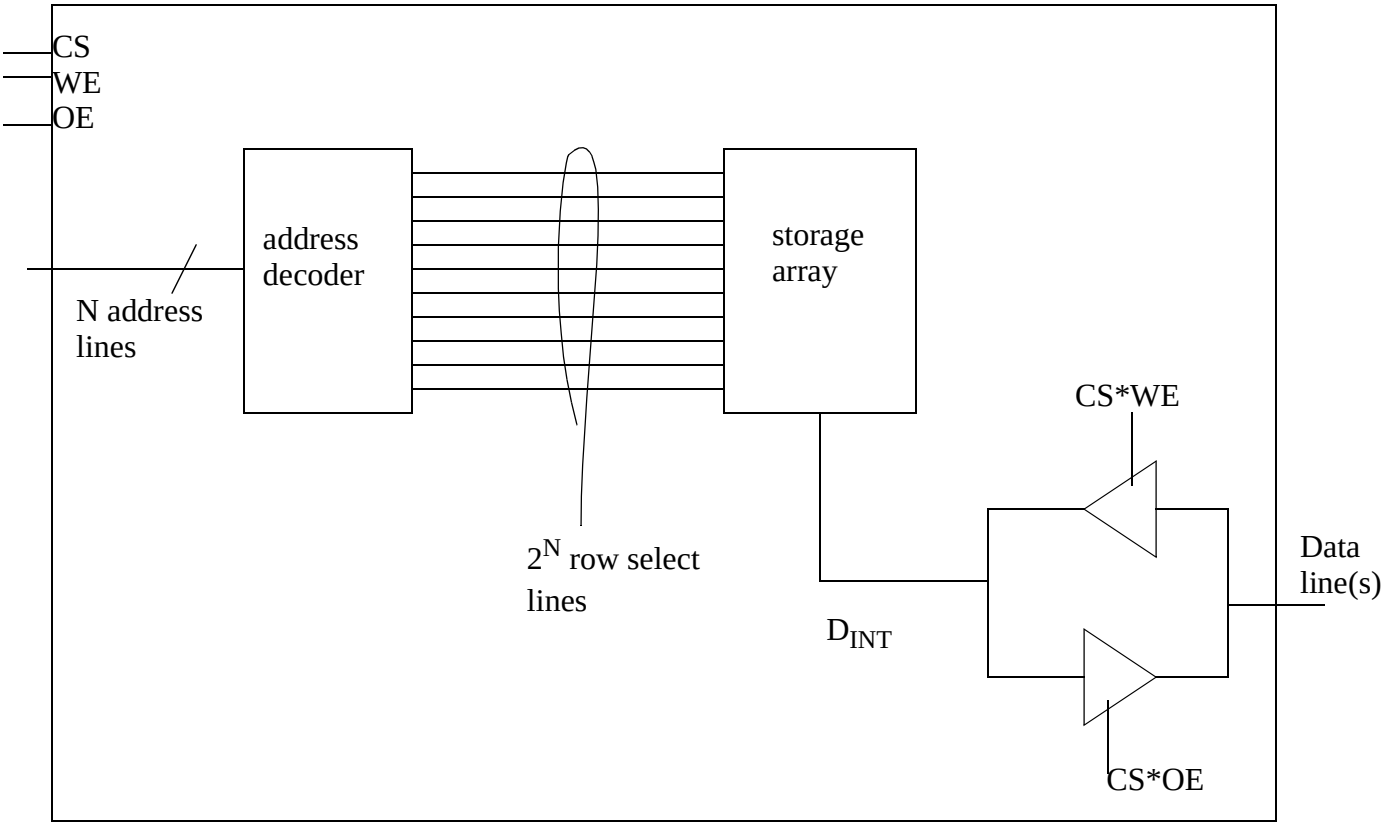
$\mathbf{t}_{A}$ - address stable until data stable

$\mathbf{t}_{WP}$- minimum write pulse

$\mathbf{t}_{EO}$ - chip enable to output valid

$\mathbf{t}_{DS}$- minimum time for data to be stable before the end of the write pulse

$\mathbf{t}_{DH}$- minimum time for data to remain stable after the end of the write pulse



Problem 2: Midterm, take 2

For the datapath and control signal implementation given below, trace the signals on the address lines, data lines, reg-A output, and ALU output as a function of time. Label the signals so that they make sense (e.g. A_n, RAM(A_n), RAM(A_{n+1}), RAM(A_n) + B, etc).

2A) Assume that register B is constant, and that the clock period T is long compared to all of the other delays in the circuit ($T >> t_*$).

2B) Use the following delays:

$\mathbf{t}_{INC}$	5ns	time from rising edge until counter outputs stable
$\mathbf{t}_{ALU}$	20 ns	ALU inputs stable to outputs stable
$\mathbf{t}_{EN}$	5 ns	rising edge of EN to data valid on tri-state buffer
$\mathbf{t}_{LATCH}$	10 ns	rising edge to output valid for register A
$\mathbf{t}_{logic}$	5ns	time for random logic (forming control signals)
$\mathbf{t}_{A}$	15 ns	address lines stable to data stable in RAM read
$\mathbf{t}_{DS}$	15 ns	data stable before end of write pulse during RAM write
$\mathbf{t}_{DH}$	0	data hold time after write pulse during RAM write
$\mathbf{t}_{WP}$	15ns	minimum write pulse
T	50ns	Master clock period

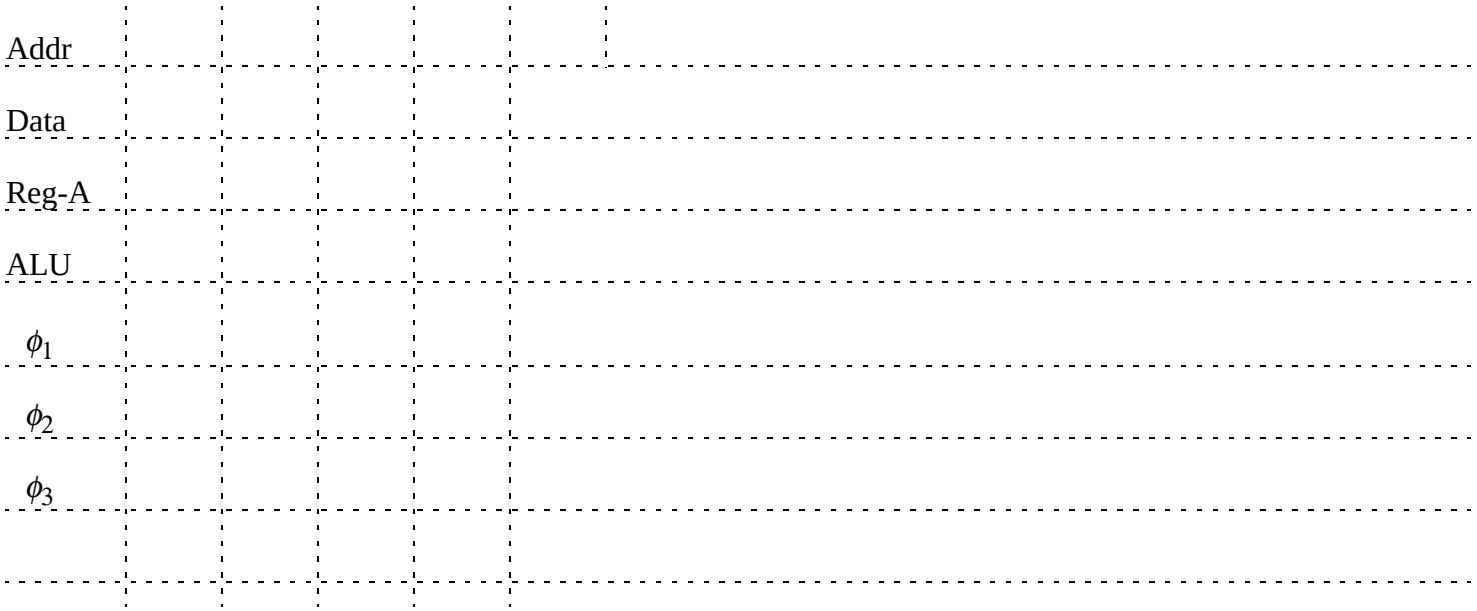
Now redesign the control signals to perform the same function in 4 states.

2C) Same as 2A, but use your new control signal timing scheme.

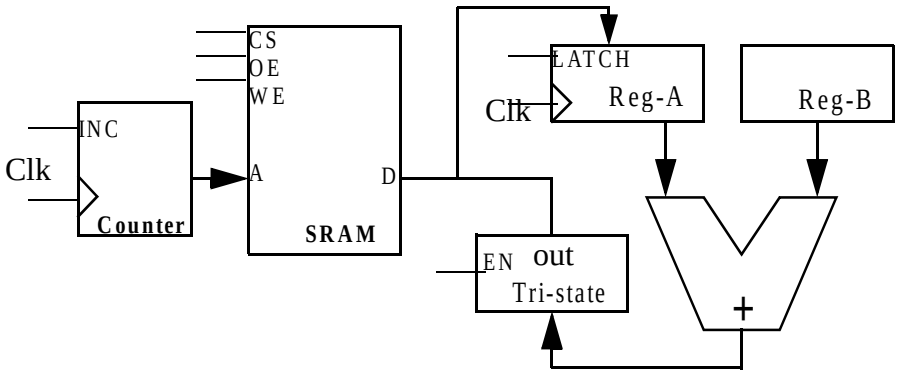
2D) Same as 2B, but use your new control signal timing scheme.

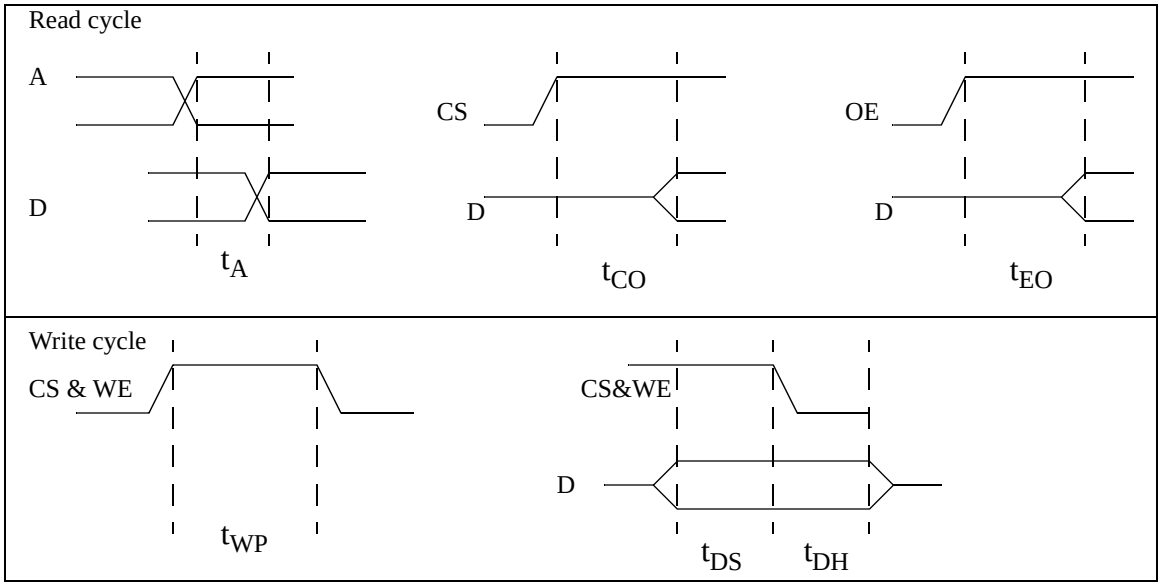
2E) Does the system work in all cases?

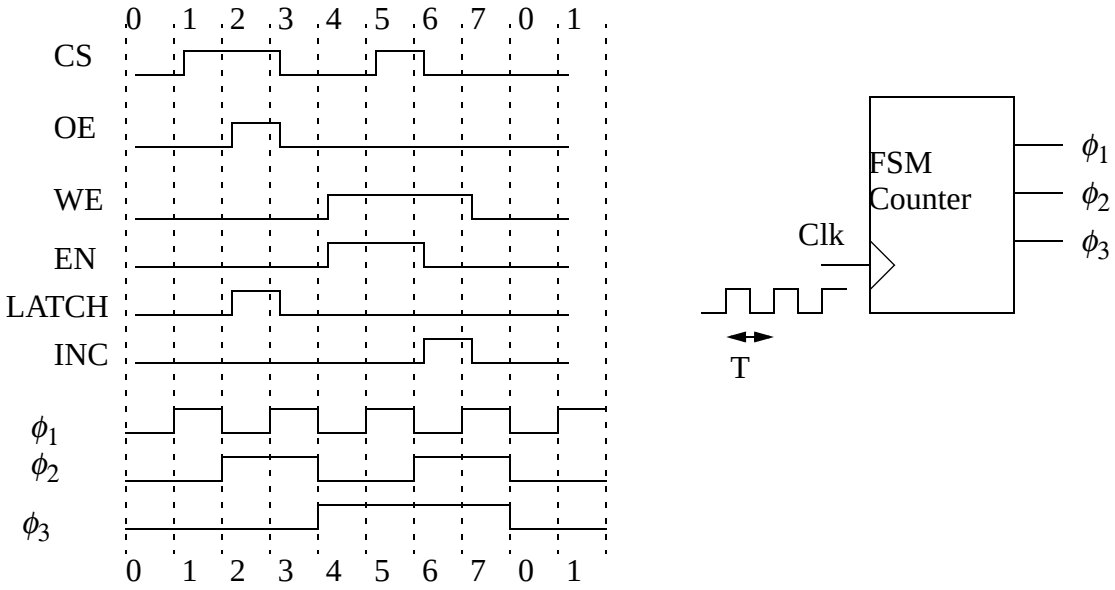
2F) How short can I make T (how fast can I run the master clock) in 2B and 2D? What if all of the RAM related timingb


Addr
Data
Reg-A
ALU
ϕ_1
ϕ_2
ϕ_3

Problem 2 Midterm, take 2







Problem 3: Thresholding.

Redraw the datapath from problem 2 so that there is an input I (for Image) and an input T (for threshold, assumed positive) going into the B input of the ALU. You can use a MUX or tri-state registers to select the input. Assume that the ALU has control inputs that allow you to do the following operations: A, B, A+B, A-B. The ALU also has outputs NEG and ZERO. Assume that your RAM is storing 7 bit intensity values.

Design a datapath controller which will sequence through the RAM, subtract the current value of I from the current RAM contents, and see if the difference is greater in magnitude than T. The controller should have two outputs, DIFF and DIR. DIFF should be asserted if the RAM location and input differ by more than T. If DIFF is asserted, then DIR should be a 1 if the new input is greater than the RAM value, and 0 if the new input is less than the RAM value.