



University of California
College of Engineering
Department of Electrical Engineering
and Computer Sciences

E. Alon

Thursday, May 11th, 2017

3:00-6:00pm

EECS 151/251A: SPRING 2017—FINAL

NAME	Last <i>Solutions</i> First
-------------	-----------------------------

GRAD/UNDERGRAD	
-----------------------	--

Problem 1: ____ / 18

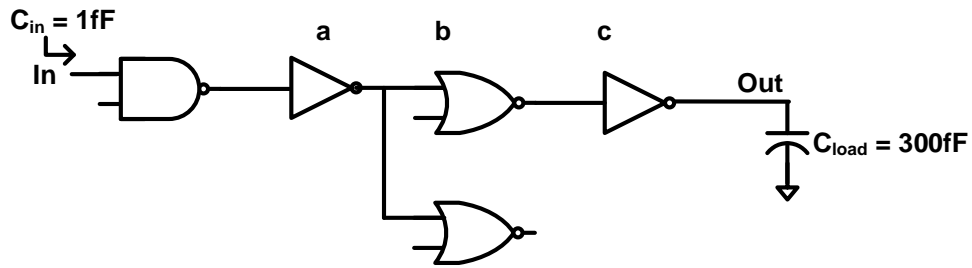
Problem 2: ____ / 16

Problem 3: ____ / 26

Problem 4: ____ / 28

Total: ____ / 88

PROBLEM 1. Logical Effort and Gate Sizing (18 points + BONUS 6 pts)



a) (4 pts) What is the path effort from In to Out?

$$PE = \pi LE \cdot \pi B \cdot F$$

$$= (5/3 \cdot 4/3) \cdot 2 \cdot \frac{300fF}{1fF} = \boxed{1333.33}$$

b) (2 pts) What EF/stage minimizes the delay of this chain of gates?

$$EF_{opt} = (PE)^{1/4}$$

$$= \boxed{6.04}$$

c) (6 pts) Size the gates to minimize the delay from In to Out.

$$\frac{300fF}{c} = 6.04$$

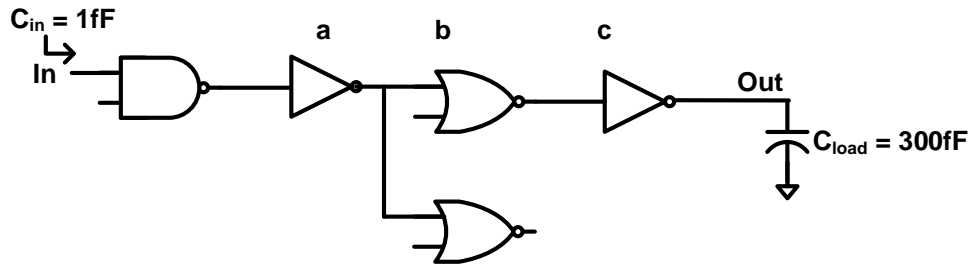
$$\frac{c}{b} \cdot \frac{5}{3} = 6.04$$

$$\frac{2 \cdot b}{a} = 6.04$$

Size	Value (fF)
a	4.53
b	13.69
c	49.65

- d) (6 pts) While maintaining the same logical functionality and without changing C_{in} , improve the delay of this chain of gates (repeated below) by adding or removing inverters; note that you don't need to worry about maintaining the logical polarity. Please draw a schematic of the improved chain; note that you don't need to provide gate sizes.

Original chain:

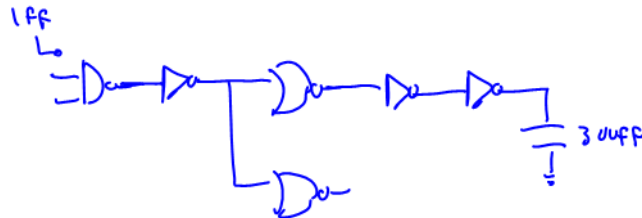


Improved chain:

To find the optimal number of stages, take $\log_4(PE)$ (since we want EP/stage to be 4):

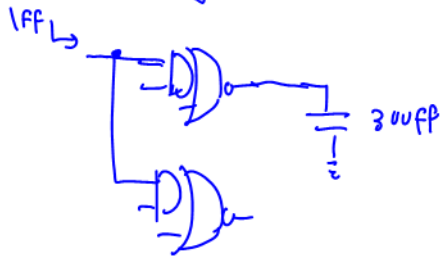
$$N_{\text{stages, opt}} = \log_4(PE) = 5.19$$

Rounding to 5 stages, we want one more inverter in the chain (at the end):



- e) **(BONUS: 6 pts)** Now let's imagine that you are working in a new technology where the γ ($= C_D/C_G$) of the transistors is 1000. Would this alter your answer to part d)? If so, explain (qualitatively, but as specifically as possible) how your answer to part d) would be different. If not, explain why changing the γ of the transistors shouldn't affect your answer to part d).

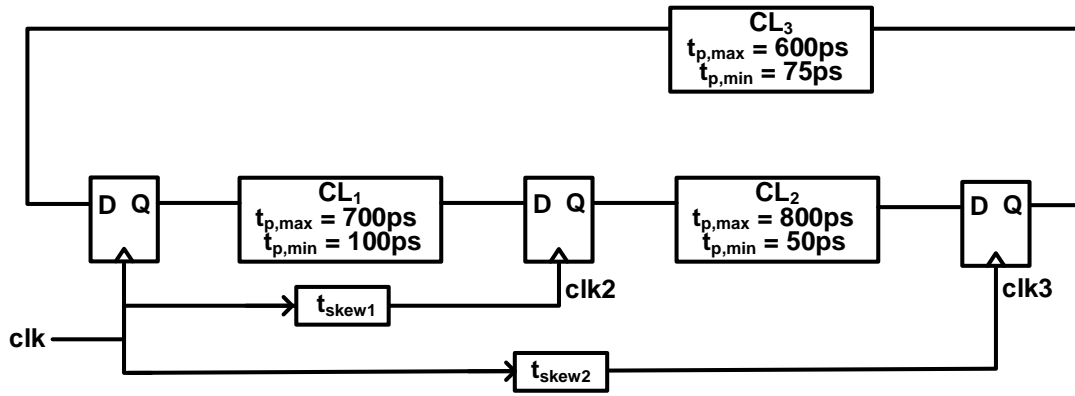
Recall that optimal f (and hence EP) was set by γ ; if we make γ very large, that implies parasitic delay will be very large, and hence want small number of stages. In this case we've made γ so large that we probably just want a single stage, so the "chain" would look something like this:



In other words, we want to use as small of a chain as possible, and hence would implement the logic using a single complex gate.

PROBLEM 2: Timing (16 points)

In this problem we will be examining the pipeline shown below. The minimum and maximum delays through the logic are annotated on the figure, and the flip-flops have the following properties: $t_{\text{clk-q}} = 50\text{ps}$, $t_{\text{setup}} = 25\text{ps}$, and $t_{\text{hold}} = 40\text{ps}$. You can assume that the clock has no jitter, but t_{skew1} and t_{skew2} can be either positive or negative.



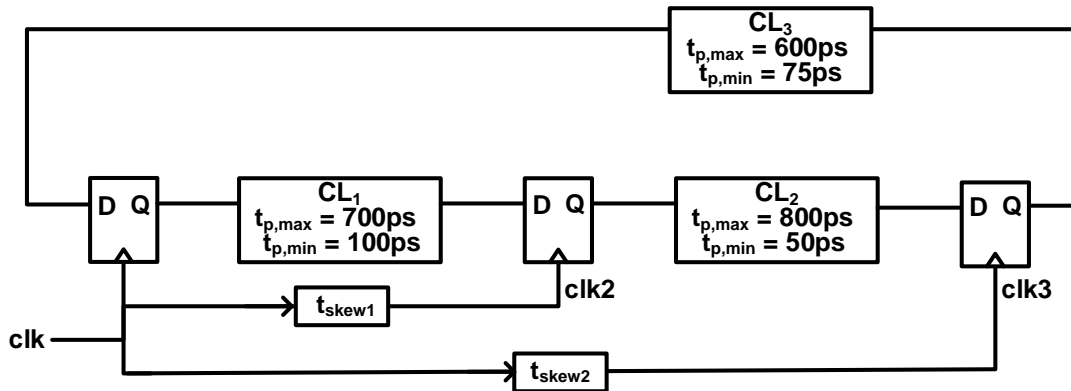
- a) (4 pts) What is the minimum clock cycle time if $t_{\text{skew1}} = 100\text{ps}$ and $t_{\text{skew2}} = 50\text{ps}$?

Notice that CL_2 has $>100\text{ps}$ more delay than the other two paths, and skew values here mean that CL_2 gets $100\text{ps} - 50\text{ps}$ less time to propagate than the clock cycle.

$$\text{So: } T_{\text{cycle, min}} - (t_{\text{skew1}} - t_{\text{skew2}}) = t_{\text{clk,q}} + t_{\text{p,max,CL2}} + t_{\text{setup}}$$

$$T_{\text{cycle, min}} - 50\text{ps} = 50\text{ps} + 800\text{ps} + 25\text{ps}$$

$$T_{\text{cycle, min}} = 925\text{ps}$$

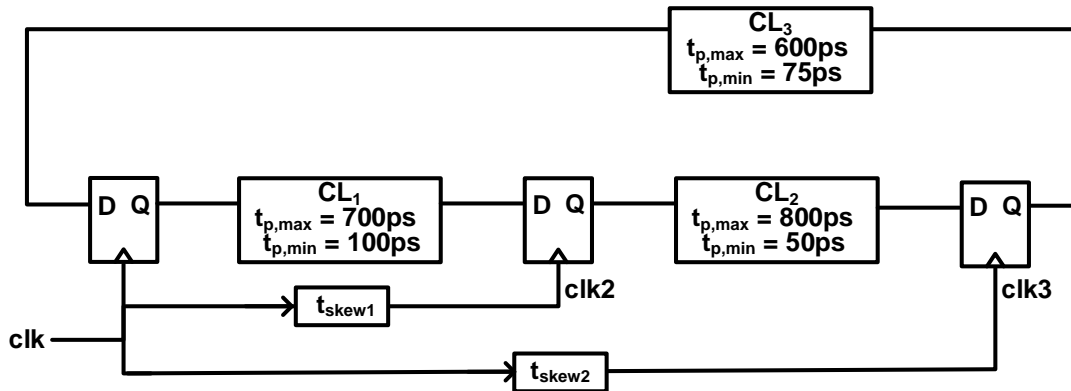


b) (4 pts) Assuming t_{skew1} is fixed at 100ps, how positive can t_{skew2} be before this pipeline (repeated above for your convenience) fails a hold-time constraint?

Check CL_1 : $t_{clk,q} + t_{p,min,cl_1} \geq t_{skew1} + t_{hold}$
 $50ps + 100ps \geq 100ps + 40ps$
 $50ps \geq 40ps \rightarrow$ no hold time (doesn't depend on t_{skew1} anymore)

Check CL_2 : $t_{skew1} + t_{clk,q} + t_{p,min,cl_2} \geq t_{skew2} + t_{hold}$
 $100ps + 50ps + 50ps \geq t_{skew2} + 40ps$
 $160ps \geq t_{skew2}$

Check CL_3 : $t_{skew2} + t_{clk,q} + t_{p,min,cl_3} \geq t_{hold}$
 $t_{skew2} + 50ps + 75ps \geq 40ps \leftarrow$ no hold time for any positive value of t_{skew2}



- c) (8 pts) If you could intentionally set the values of t_{skew1} and t_{skew2} , what values would you choose in order to minimize the cycle time of this pipeline (again repeated above)? What would be the cycle time in this case?

CL_2 has the largest delay (by 100ps), so we want to give it more time by making $t_{skew2} > t_{skew1}$. However, when t_{skew1} was 100ps, we already saw in part b) that we couldn't make $t_{skew2} > 160ps$ - in other words, we can give CL_2 at most 60ps of extra time before we run in to a hold time violation. Since both CL_1 and CL_3 will still be less critical than CL_2 in this case, we can just choose $t_{skew1} = 0ps$ and $t_{skew2} = 60ps$, so that:

$$T_{cyc, min} + t_{skew2} = t_{clk-q} + t_{p, max, CL_2} + t_{setup}$$

$$T_{cyc, min} = 50ps + 800ps + 25ps - 60ps$$

$$= 815ps$$

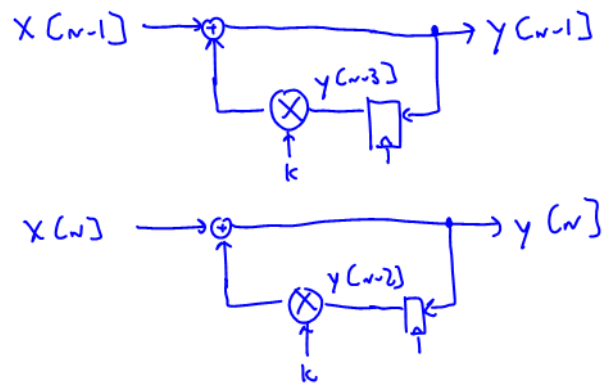
PROBLEM 3. (26 pts) Miscellaneous

- a) **(8 pts)** Given $x[n]$ and $x[n-1]$ as inputs and using only adders, multipliers, and registers, sketch a design that uses unrolling to produce new values of $y[n]$ and $y[n-1]$ on each clock cycle, where $y[n]$ is set by the following equation:

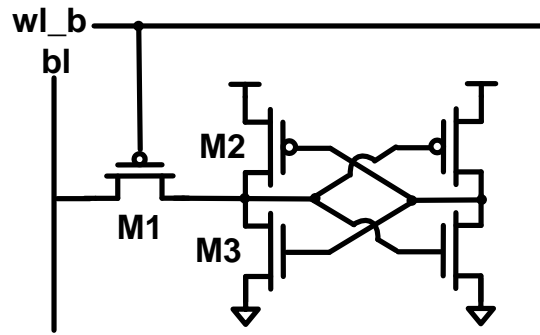
$$y[n] = x[n] - k*y[n-2]$$

where k is some fixed constant.

$$y[n-1] = x[n-1] + k y[n-3]$$



- b) (8 pts) One of your colleagues approaches you and suggests switching all of your memories from using the classic 6 transistor SRAM cell to the 5 transistor (5T) cell shown below. Assuming that you are not allowed to use sense-amps or mode-dependent (read vs. write) voltage levels to read out of the memory, explain why it is not possible to ensure both robust read and write operations from a memory built out of these 5T cells.



Consider the write case first: the value we want to store in the cell would be driven on to bl, and in order to ensure that the cell actually takes on that value, M1 would need to be sized such that it has significantly less resistance than both M2 and M3.

The problem then is that during a read, we don't want M1 to be able to overwrite the value stored in the cell, but we just argued above that it has to be able to do that in order to perform writes...

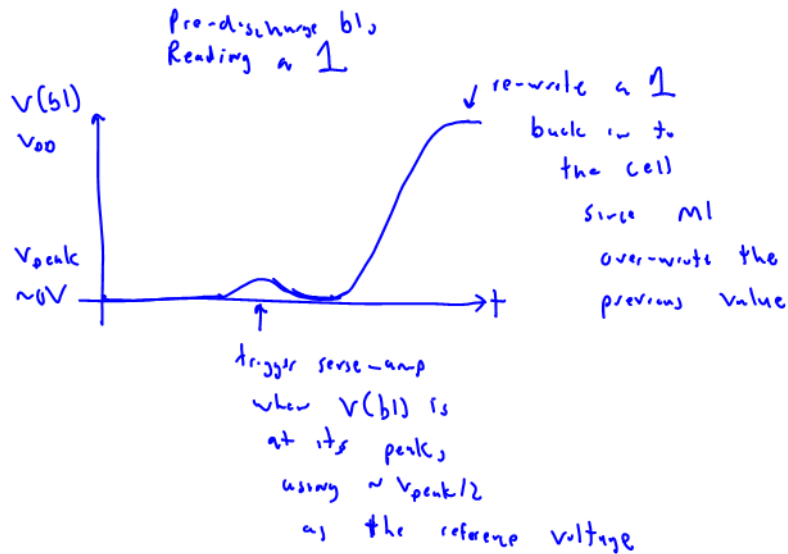
To put this more mathematically and assuming bl is pre-discharged for reads:

$$\text{For write: } R(M1) < R(M2) \quad (\text{writing a zero})$$

$$R(M1) < R(M3) \quad (\text{writing a one})$$

$$\text{For read: } R(M1) > R(M2) \quad (\text{reading a one})$$

- c) (10 pts) Now assuming that you can use sense-amps (but still no mode-dependent voltages) and that the PMOS transistor M1 is sized so that its resistance is much lower than that of both M2 and M3, explain how you would read out of a memory built out of such 5T cells. In particular, you should sketch a waveform of what should happen on the bitline during a read, indicate when the sense-amp should be triggered, and what should happen to the bitline after the sense-amp has been triggered.



(Can also pre-charge bl to $v_{00}/2$ or even v_{00} , key is to trigger the sense-amp at the right time and re-write the cell.)

PROBLEM 4. (28 points) Datapaths and Arithmetic

You recently joined a start-up named ML4eva that has decided to implement a custom processor to accelerate neural network computations, and you have been put in charge of developing the datapath for ML4eva's processor design. Fortunately, your datapath (for now) only ever needs to handle one specific instruction named MLinst.

Calling MLinst(\$c1, \$c2, \$c3, \$r1, \$r2, \$r3, \$r4) tells the processor to perform the following computation:

$$r4 = f(c1*r1 + c2*r2 + c3*r3)$$

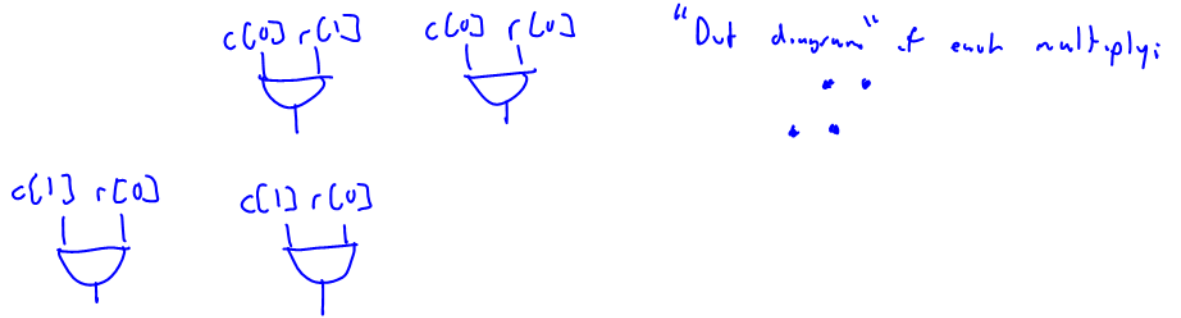
where $f()$ is some non-linear function, $c1$, $c2$, and $c3$ are the values stored in registers in a "coefficient file" (at locations indicated by \$c1, \$c2, and \$c3), and $r1$, $r2$, $r3$, and $r4$ are the values stored in registers in a register file (at locations indicated by \$r1, \$r2, \$r3, and \$r4). In other words, register \$r4 stores the result computed by applying $f()$ to the weighted sum (with weights set by $c1$, $c2$, and $c3$) of $r1$, $r2$, and $r3$.

- a) (4 pts) Assuming you wanted to implement a datapath that could complete an MLinst instruction in a single cycle, how many independent read ports would the coefficient file require? How many independent read and write ports would the register file require?

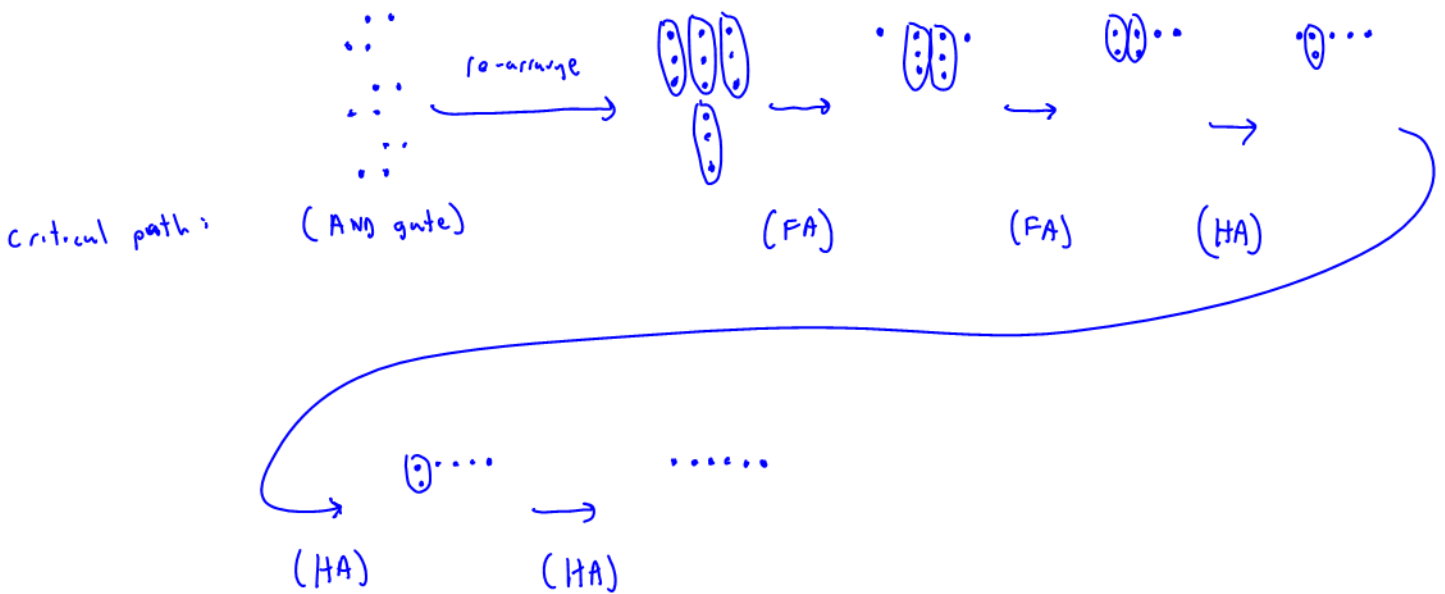
* Need to read 3 coefficients every cycle, so need 3 read ports for coefficient file.
* Need to read 3 registers and write 1 to 1 every cycle, so need 3 read ports and 1 write port for register file.

b) (12 pts) Assuming that both the coefficients and the registers each contain 2-bit unsigned values (e.g., the value stored in \$c1 is $c1[1:0]$, where $c1[1]$ is the MSB) and using only AND gates, full adders, and half-adders, sketch an implementation of a hardware block that would compute $c1*r1+c2*r2+c3*r3$. In order to receive full credit, the critical path of your design should be no more than 6 gates long (where you can assume all gates have equal delays; in this context you can treat both full adders and half adders as gates). Note that you will receive full credit for sketching your design using a "dot diagram" as long as you indicate how the initial partial products are formed.

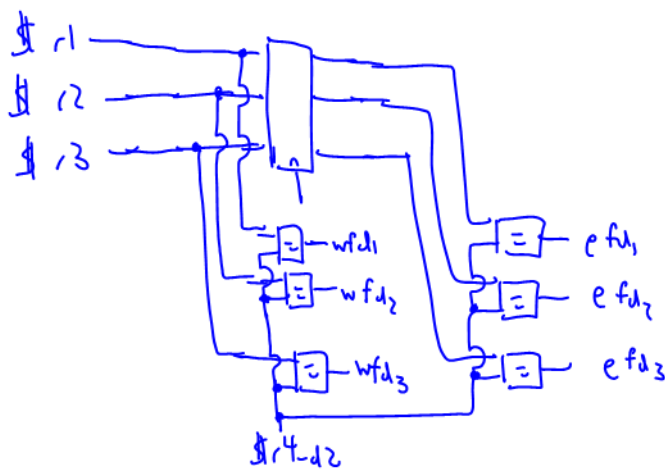
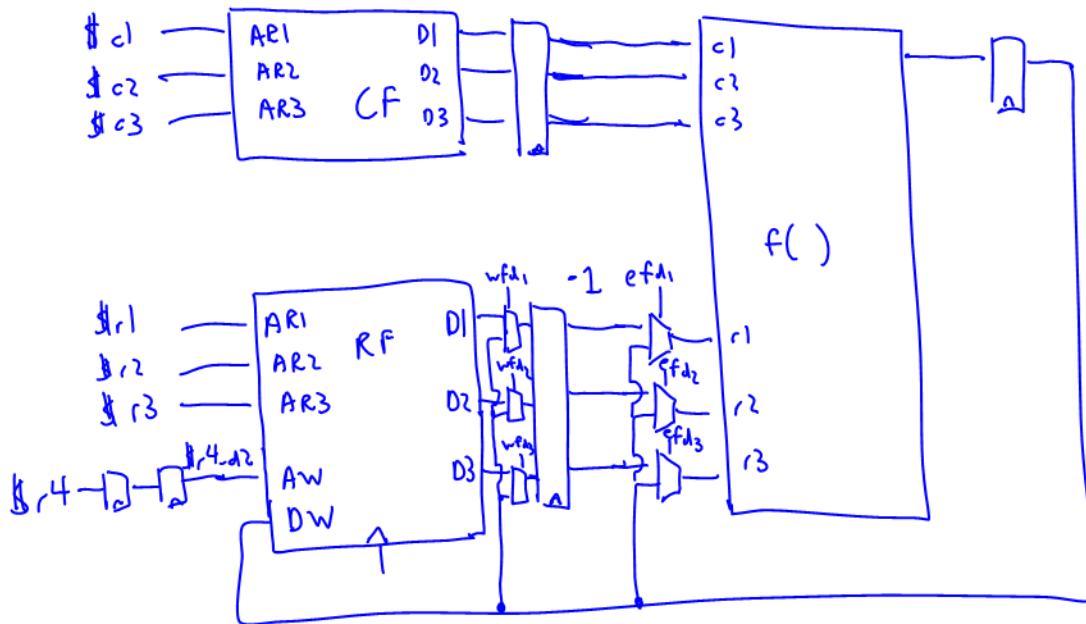
Each multiply's partial products are formed with AND gates:



Overall hardware needs to compress 6 partial products:



- c) (12 pts) Assuming you are given coefficient file and register file blocks as well as a block that produces as its output $f(c1*r1 + c2*r2 + c3*r3)$ (given $c1, c2, c3, r1, r2,$ and $r3$ as inputs), and ignoring the hardware associated with fetching the instruction, sketch a 3-stage pipelined implementation of this datapath that would consist of Register/Coefficient Fetch, Execute, and Write Back. You can assume that the register file has asynchronous reads but synchronous writes. For full credit, be sure to include any forwarding hardware necessary to avoid data hazards.



EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK

EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK