

CS 5 spring 2021, hw 5 guidelines...

[CS5home](#)

Black BLACK! (following gold this week...)

Gold GOLD! The hw we're grading for Gold and Black: [Link to gold HW: Hw 5. gold](#)

Gold and **Black**

This week's comment theme: **Wildly Incorrect Predictions** (see below for many others), e.g.,

This week we're including wildly incorrect predictions to help put into perspective the sometimes-surreal details of Logisim/Circuits/Hmmm/Assembly-language/Python loops...

For example:

*Stocks have reached what looks like a permanently high plateau.
... by I. Fisher, Yale Professor of Economics, 1929.*

+++ Good luck with Hmmmwork #6! +++

hw5pr0: *reading responses (to the* Water-based computer *article)*

- Read the excerpt and questions at <https://www.cs.hmc.edu/twiki/bin/view/CS5/Reading5>
- Week 5 had two prompts - they should have addressed one or the other (-1 or -2 if not)
 1. The article makes the claim that the water-droplet computer has a "universal nature." What circuit-level functionality did the researchers need to show in order to claim universality, i.e., that their computer could compute *anything* that an ordinary computer could?
 2. The article contrasts the water-droplet computer with our era's silicon devices by noting that the droplet computer works by distributing physical matter (well, water as opposed to electrons, at least). Insofar as biological organisms are "types of computer" (not an agreed-upon point, for sure), are they fundamentally computing by moving matter, by moving electrons, or some other physical interaction?

This is **not** a neuroscience question, or even a question with a right or wrong answer. Rather, it's one of perspective: what is the raw material, fundamentally, that biological computers are manipulating, in your opinion? **Incorrect, but admired, answer:** *Like the Starbucks injectible nanobot, which strategically releases caffeine throughout one's life, humans are computers that move around matter, i.e., themselves!*

- Here is a more general [page of scoring guidelines and possible comments](#)

Circuitverse!

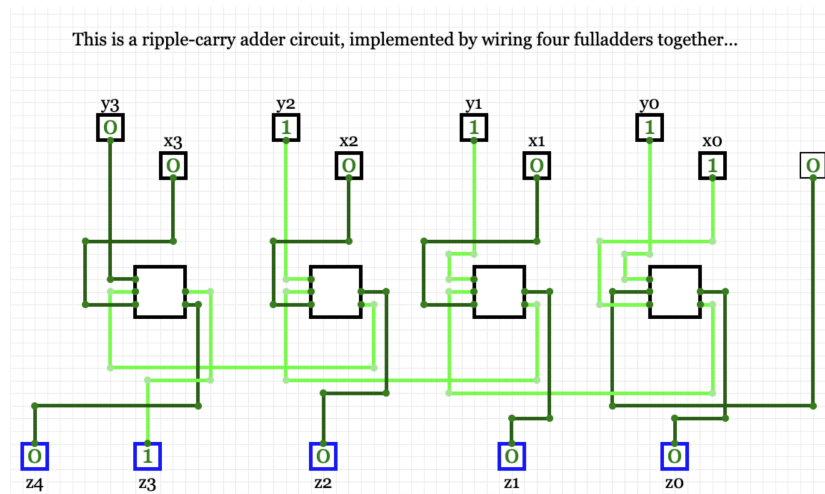
We are going to sight-grade our hw5 **Circuits**: *You'll soon be Circuitverse-fluent!*

hw5pr1: lab (ripple-carry adder) 30 pts total

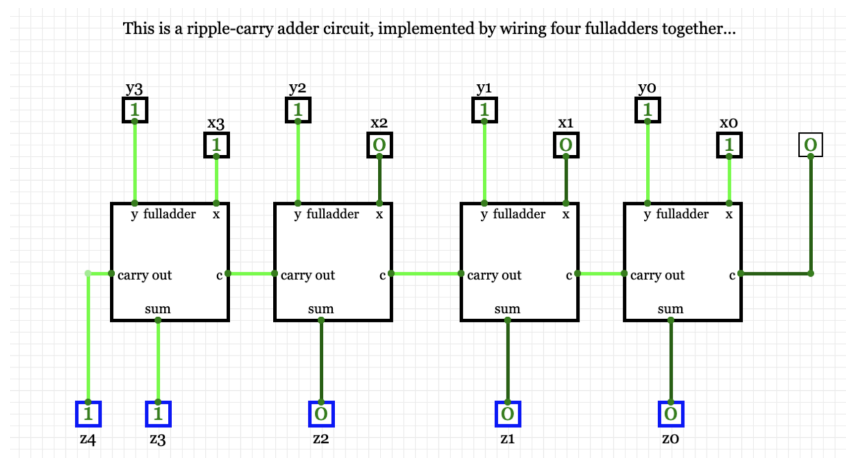
- + [[only look at the FA and XOR if there's no ripple-carry... ~ estimate the effort invested]]
- + are there **four** Full Adders? [8pts]
- + Do the carries "**ripple**" (from R to L) [8pts]
- + are the other **inputs and outputs** hooked up correctly? [8pts]
- + is the **CARRY OUT of the left** column hooked into z4? [3pts]
- + is the **CONSTANT 0** hooked into the "carry-in" bit of the right column? [3pts]
- + are there artistic flourishes?! [up to +2.42 EC pts]

Examples:

Regular!

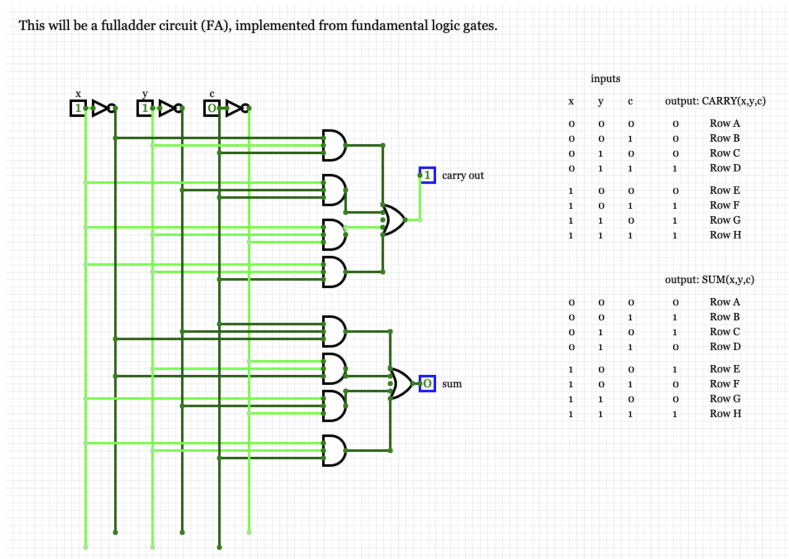


Especially-well-formatted design!!



Example FA, just in case:

This will be a fulladder circuit (FA), implemented from fundamental logic gates.

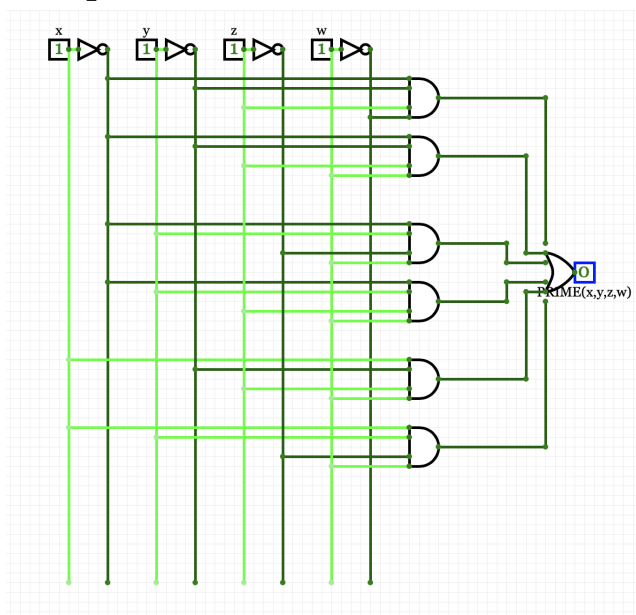


The above circuit is the **Full Adder** (It is not the prime-checking circuit: see below)

hw5pr2: A Prime-checking circuit (minterm) 25 pts total

- + Are there **six** AND gates? [10pts]
- + Does each **AND** connect to all four input bits (inverse or uninverted)? [5pts]
- + Does an **OR** gate (or a few) collect all of the ANDs' outputs? [5pts]
- + Was there a good-faith effort? [5pts]
- + Are there artistic flourishes?! [up to +2.42 EC pts]

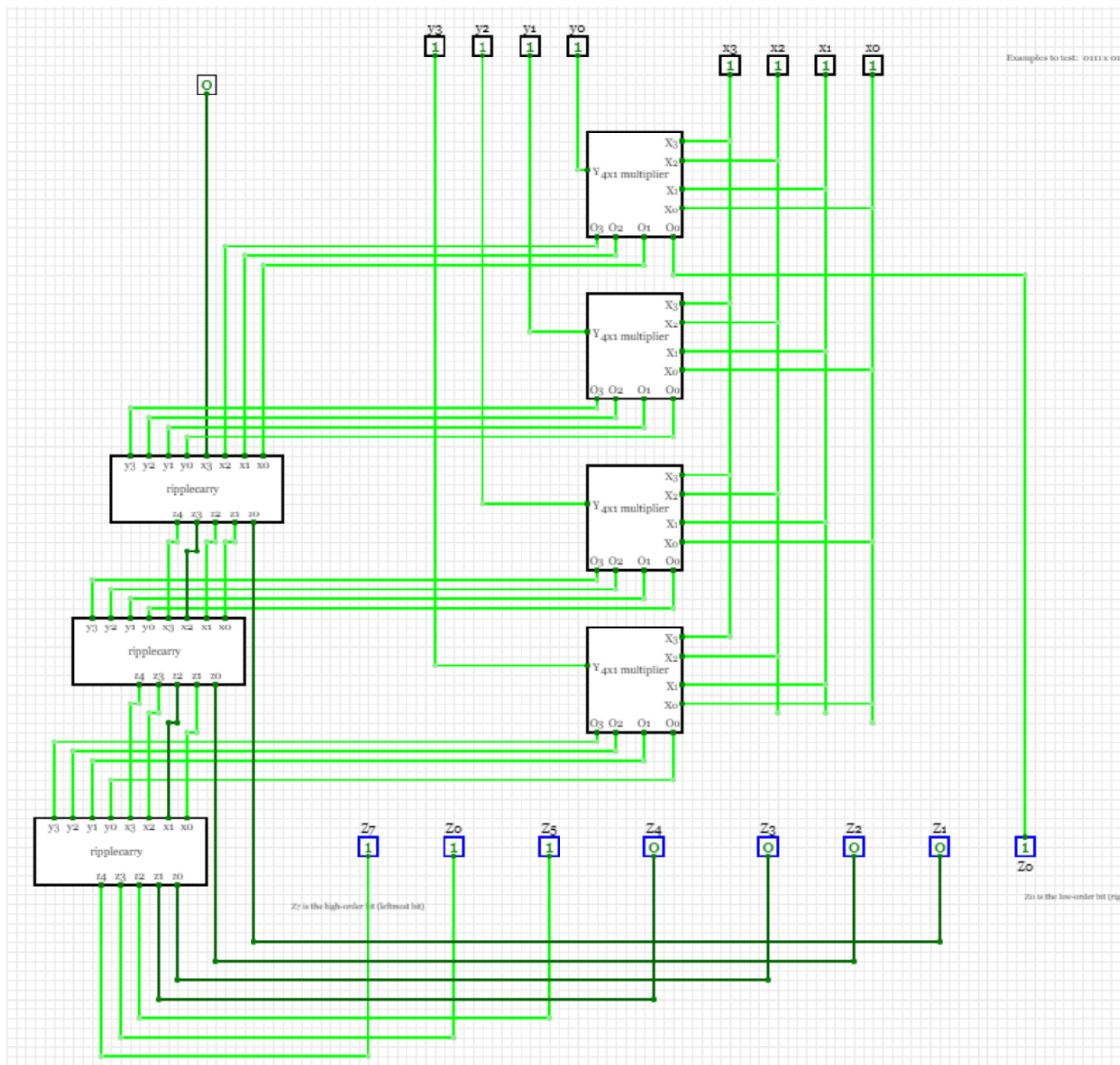
Example:



hw5pr3: Four-bit Multiplier circuit 40 pts total

- + Are there **three** ripple-carry adders (RCA)? [5pts]
- + Are there **four** four-by-1-bit multipliers? [5pts]
- + Does the **low** bit of the first 4x1's go **directly to Z0**? [5pts]
- + Is there a **constant 0** at the **high** bit of the first RCA? [5pts]
- + Do all outputs but one of the **first** RCA go into the **second** RCA? [5pts]
- + Do all outputs but one of the **second** RCA go into the **third** RCA? [5pts]
- + Do those second- and third-RCA's also get inputs from a 4x1 multiplier? [5pts]
- + Was there a good-faith effort?! [5pts]
- + Is it really organized? [up to +1.42 EC pts]
- + Are there other artistic flourishes?! [up to +4.42 EC pts]

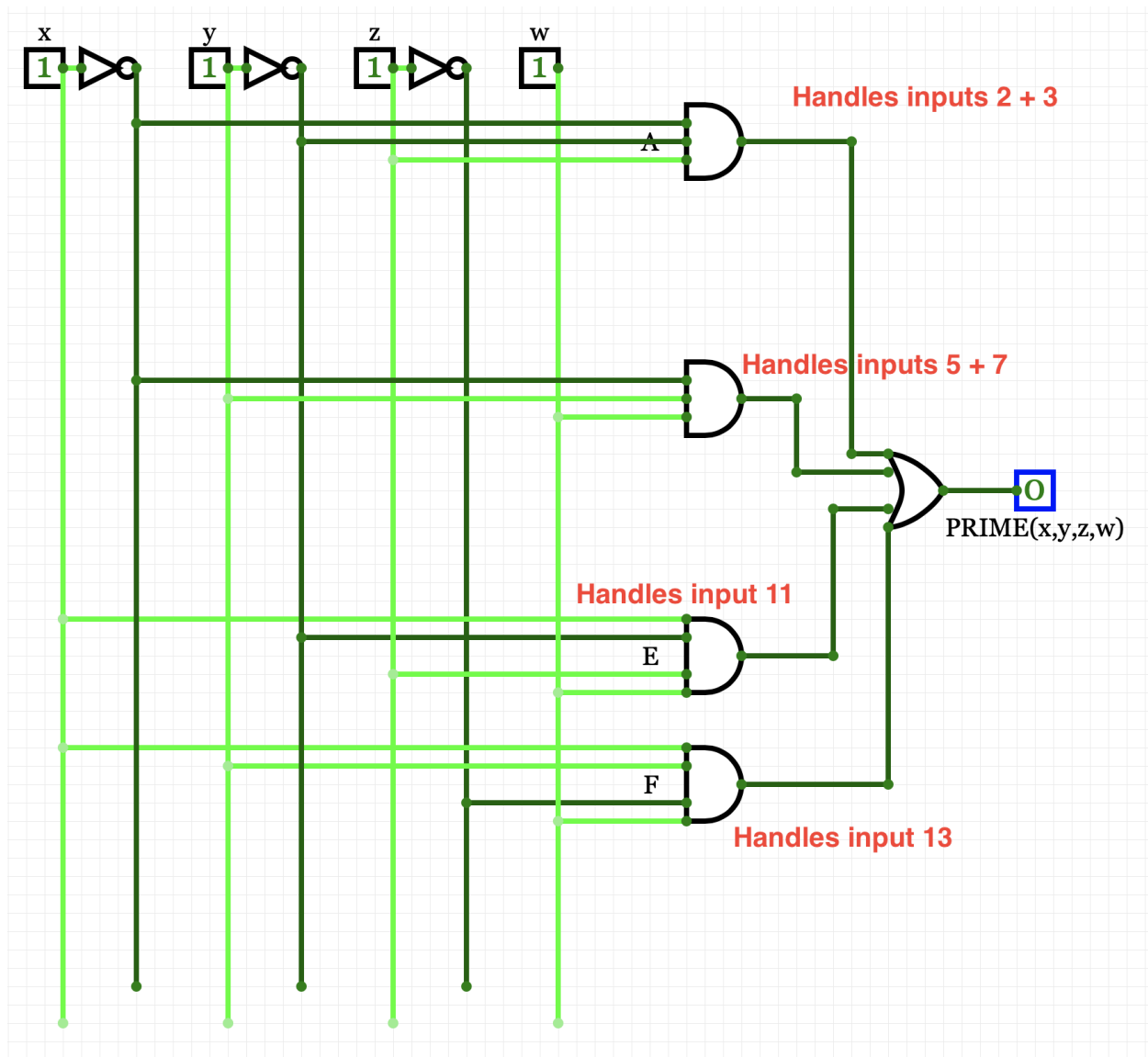
Example: (reasonably organized, but not "extra-credit" level...)



hw5pr4 Optimized Prime! up to +5 pts

- + Are there **five or fewer** AND+OR gates? [+5pts]
- + More than five? [use your judgment, perhaps +2-3]
- + Four or fewer—we want to know! (check to see if it looks like it works...)
- + The circuit should output 1 for inputs of 2, 3, 5, 7, 11, 13
- + And should output 0 for all other inputs: 0, 1, 4, 6, 8, 9, 10, 12, 14, 15
- + If you spot a couple of primes (or composites) it is wrong about, -1

Example: (typical, but not the only approach!)



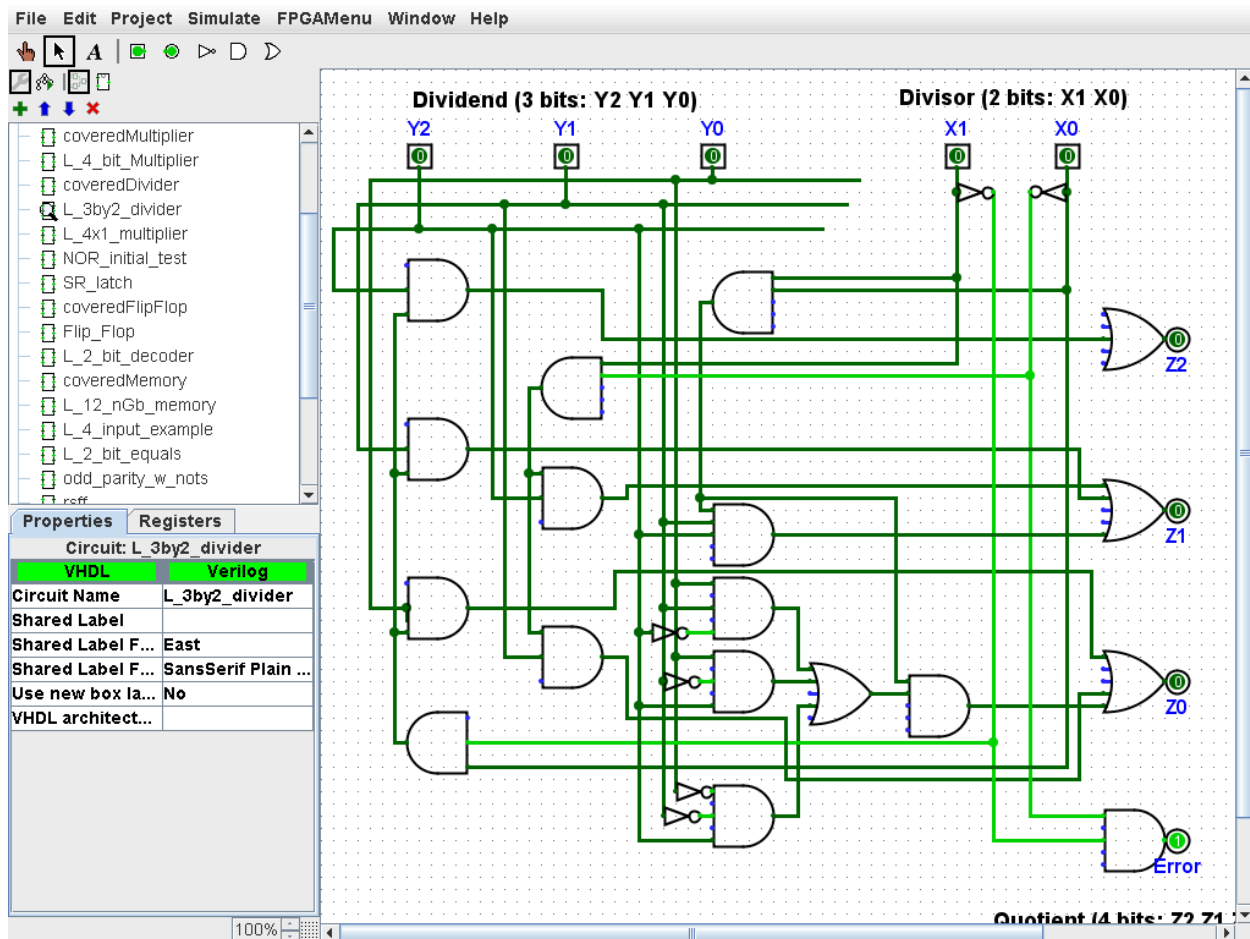
- *Cool circuit artwork?* +2-5pts (or more) of Ex. Credit is OK...

hw5pr5 Divider e.c. up to +5 pts

This one is tough because it can be a bit of a wiring mess.

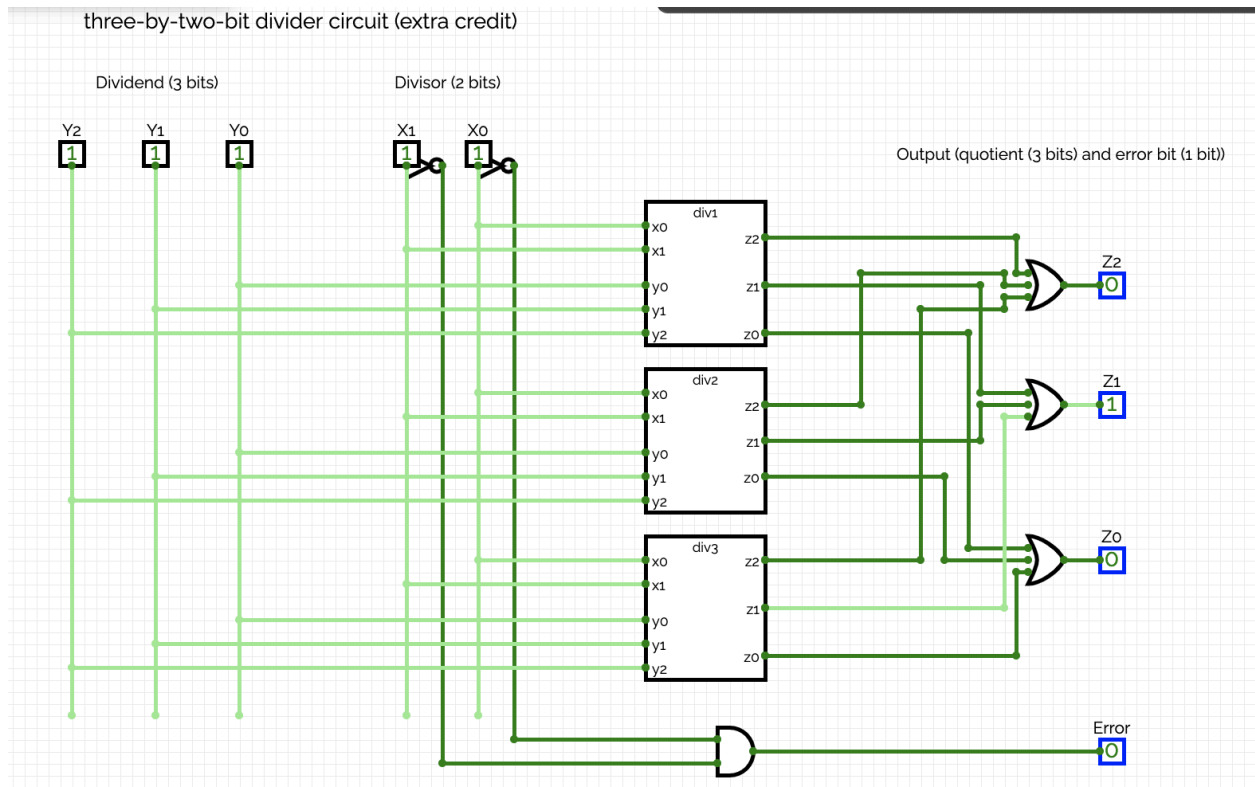
- + Are there **at least two OR** gates? [+5pts]
- + Is there a single AND gate for the 00 case?
- + Is the 00 case wired correctly to produce an error? [-1]
- + If you can spot that the divide-by-1 case gives no change, good [-1]
- + Likewise, if divide-by-2 gives a shift right, that's good [-1]
- + For divide-by-3, we'll just trust them!
- +

Example: (pretty bad, but not the only approach!)

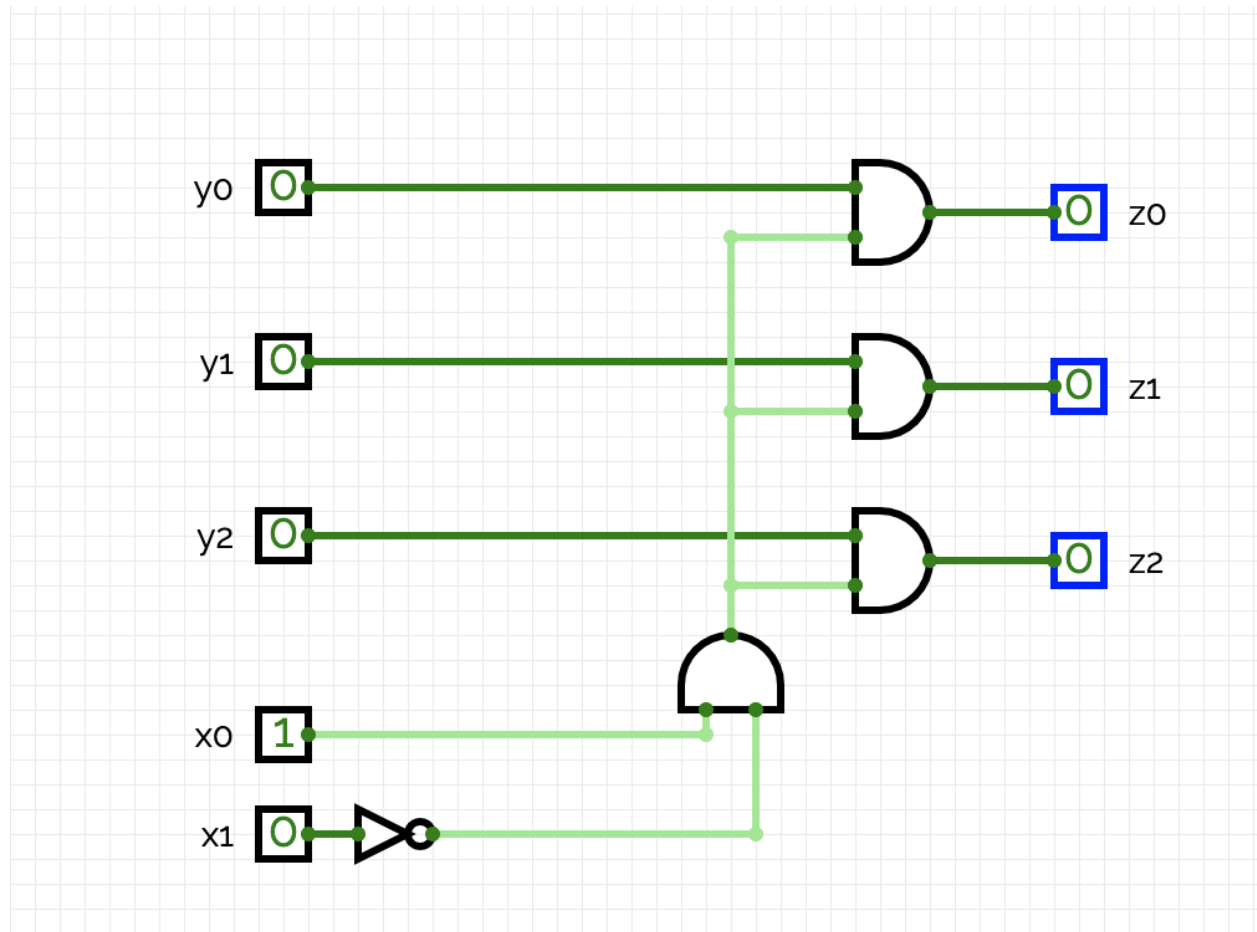


Better example: doing it piece-by-piece:

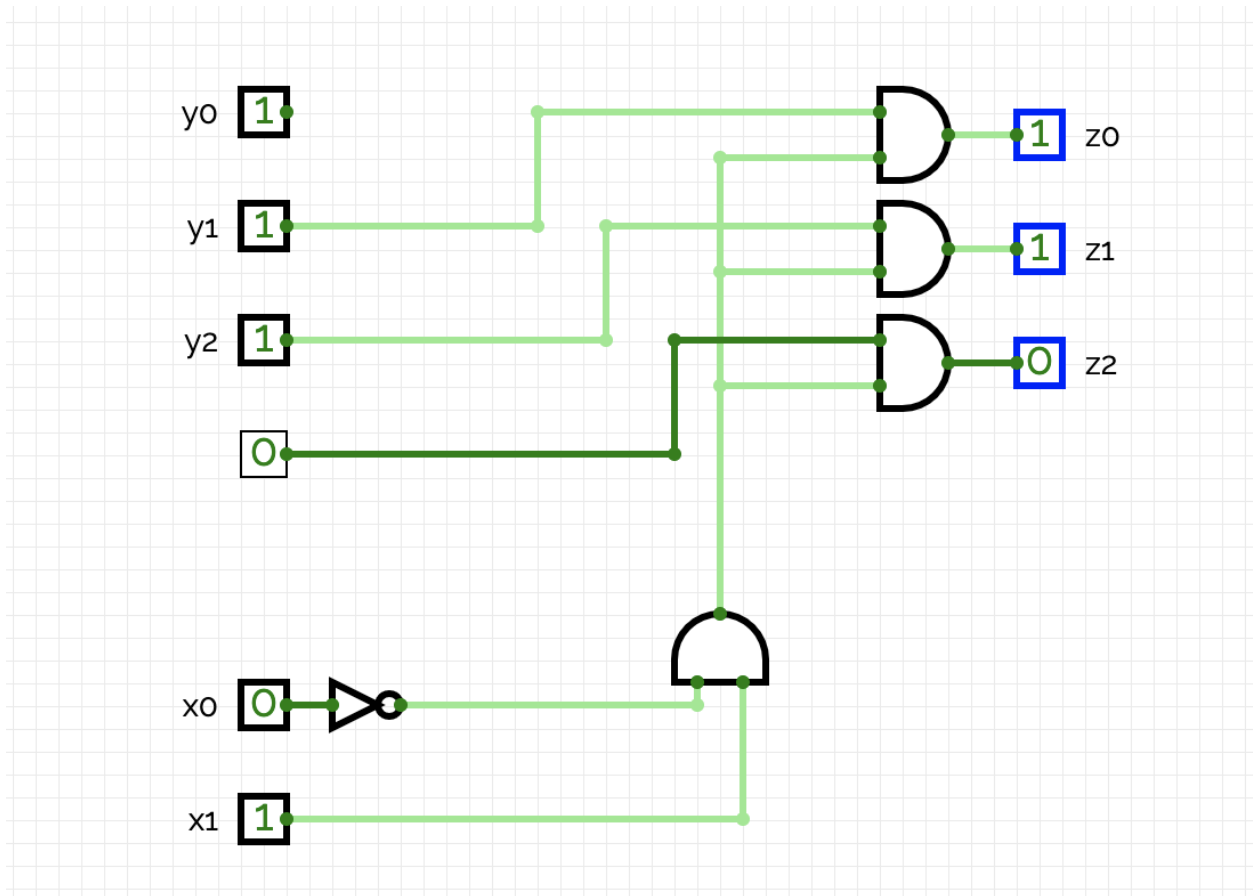
Main divider (with built-in error checking):



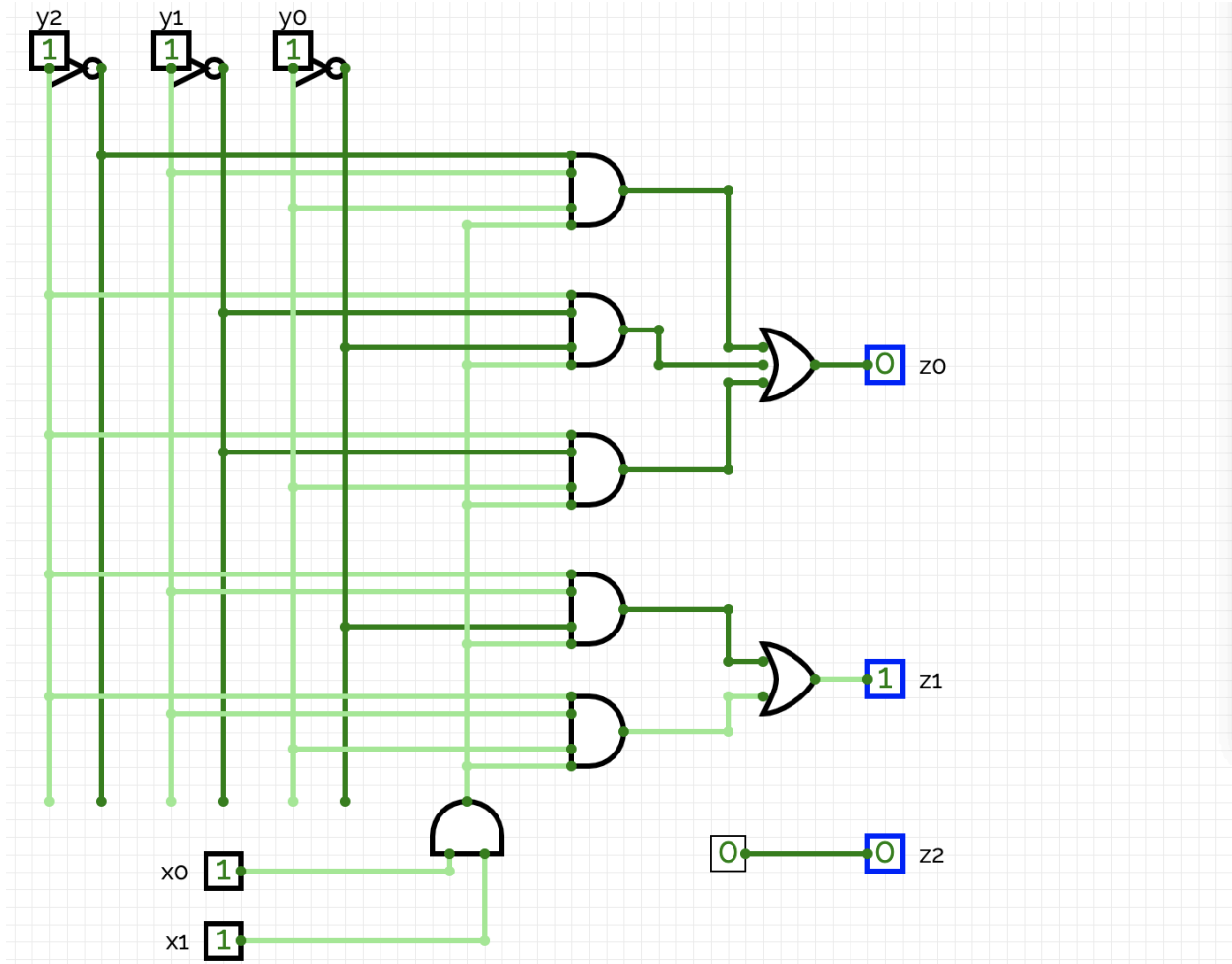
Divide-by-1:



Divide-by-2:



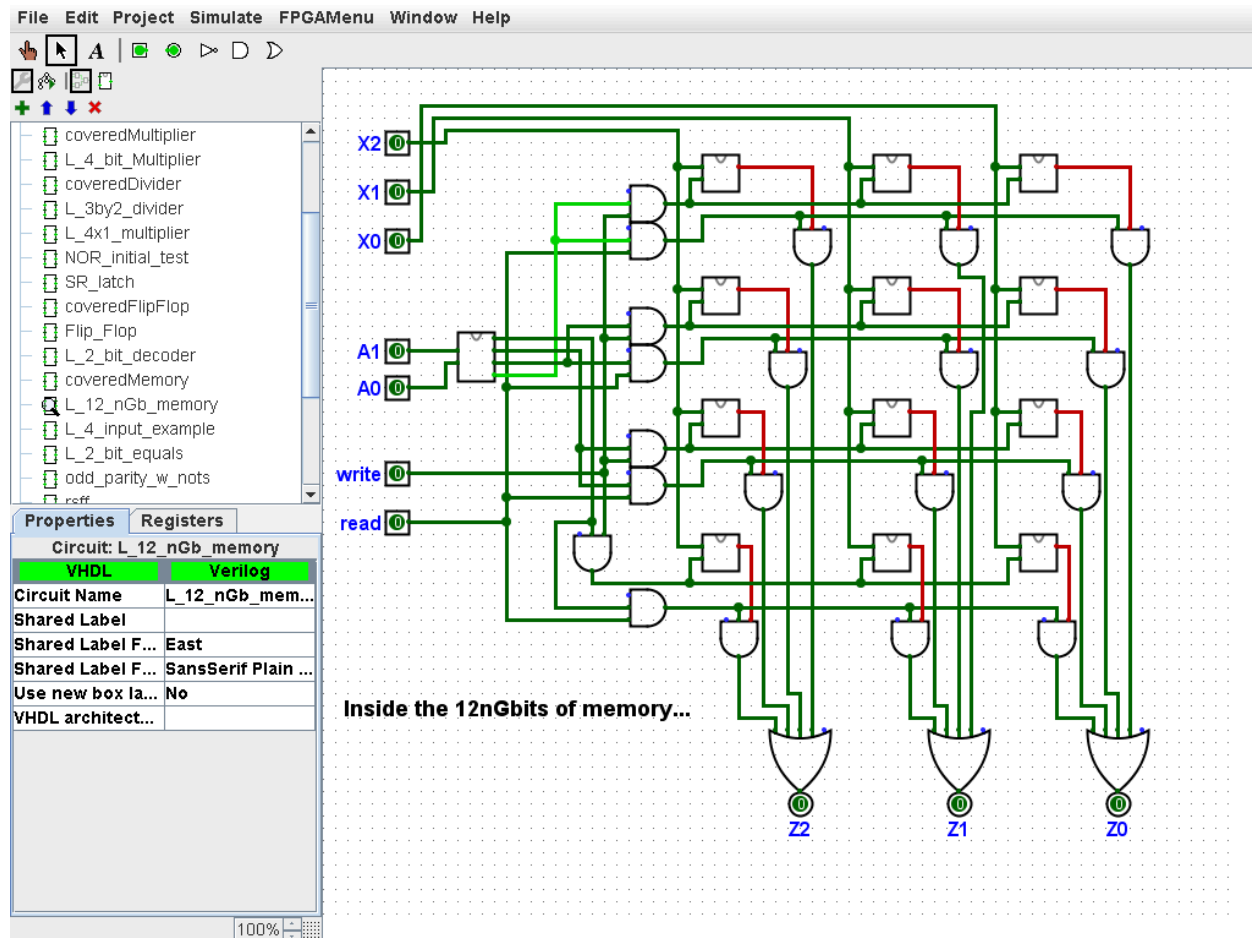
Divide-by-3:



hw5pr6 Memory e.c. up to +12 pts

- + Is there a 1-of-n decoder? [+2pts]
- + Is the decoder wired to AND gates that connect to the WRITE input and the clock inputs of the flip-flops? [+4pts]
- + Does the READ input connect to AND gates that select bits for output? [+2pts]
- + Does each output have a 4-input OR gate? [+2pts]
- + Are the three inputs wired to the three columns of flip-flops? [+2pts]
- + It's OK not to have a flip-flop screenshot

Example: (typical, but not the only approach!)



- **Add a comment** ... and, perhaps a failed prediction that you like
- **Submit** everything with the 'z' key or "Next Ungraded" ...

See our "good" bad predictions, here, if you'd like to add those to the grading comments...

"Good" prediction fails to use... [mostly from http://rinkworks.com/said/predictions.shtml](http://rinkworks.com/said/predictions.shtml)

- The coming of the wireless era will make war impossible, because it will make war ridiculous — G. Marconi, inventor of the radio, 1912
- 640K ought to be enough for anybody. — Bill Gates, 1981
- Fooling around with alternating current is just a waste of time. Nobody will use it, ever — Thomas Edison, 1889
- Where the ENIAC is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and weigh only 1.5 tons. — *Popular Mechanics*, 1949
- *It will be years — not in my time — before a woman will become Prime Minister.* — Margaret Thatcher, 1974.
- *Stocks have reached what looks like a permanently high plateau.* — Irving Fisher, Professor of Economics, Yale University, 1929.
- The cinema is little more than a fad. It's canned drama. What audiences really want to see is flesh and blood on the stage — Charlie Chaplin, 1916

- Who the hell wants to hear actors talk? — *H. M. Warner, Warner Brothers, 1927.*
- The wireless music box [radio] has no imaginable commercial value. Who would pay for a message sent to no one in particular? — David Sarnoff, inventor of the television, 1921
- While theoretically and technically television may be feasible, commercially and financially it is an impossibility. — *Lee DeForest, inventor.*
- X-rays will prove to be a hoax. — Lord Kelvin, 1833
- There is no reason anyone would want a computer in their home.
— Ken Olson, DEC chairman, 1977
- There will never be a bigger plane built — said by a Boeing engineer, right after he got off from his first flight of the 247, a double-engine plane that could accommodate up to ten passengers.
- "The concept is interesting and well-formed, but in order to earn better than a 'C', the idea must be feasible.
— *A Yale management professor, in response to Fred Smith's paper proposing overnight delivery service. (Smith went on to found the FedEx Corporation...)*

Thank you, everyone, for grading CS5!

Worth keeping on all grading pages: XKCD's 10,000 strip: <http://xkcd.com/1053/>

How to get to this page:

The CS5 site should be top link when Googling

These notes are linked at the bottom of CS5's page as

Also, the submissions site is at

hmc cs5

for grutors - guidelines

cs.hmc.edu/submit

Prior years' notes:

Green

GREEN!

[editable guidelines](#)

[read-only full-course guidelines](#)

Link to [CS5 BLACK grading page](#)

- Download and look over their Logisim file briefly...

- **Style**

- *NO style points for circuits*

EXTRA CREDIT is available for cool/artistic flourishes - and for MEMORY (see below)!

- **Correctness -- these do need to be hand-tested...**

- download the file you'll test...

- **First, test the MULTIPLIER**

- test $6*7 == 42$ (also test the reverse: $7*6 == 42$)
- test $15*15 == 225$ (225 is 11100001 in binary)
- if these all work, the FullAdder and RippleCarryAdder must work, too (total: **+70 points**)
- if one or both of those fail, subtract **-10pts** for each of the two tests that doesn't work, then...
- check if the Ripple-Carry Adder is correctly implemented (**-15pts** if not, partial credit available)
- check if the Full Adder is correctly implemented (minterm) (**-15pts** if not, again, w/partial cred.)
- as usual, if a single, small error cascaded into many, you can adjust for that...

- **Then, test the DIVIDER**

- test $7/0 == \text{Error}$ (the error bit must be on; anything is OK in the other output bits)
- test $7/1 == 7$ (the error bit should be off)
- test $7/2 == 3$ (the error bit should be off)
- test $7/3 == 2$ (the error bit should be off)
- test $5/3 == 1$ (the error bit should be off)
- test $5/2 == 2$ (the error bit should be off)
- you're welcome to test a couple of others of your choice, but not really needed...
- **-5pts** for the first error, **-2pts** for additional errors,
- as usual, if a single, small error cascaded into many, you can adjust for that...

- **Then, test the Extra Credit**

- test the 2-bit binary-to-single-wire **decoder**: give **+1ec points** if it works
- test the **SR latch** (S should set Q to 1, R should reset Q to 0): give **+1ec points** if it works
- test the **flip-flop** (should store the data bit when the strobe is 1): give **+1ec points** if it works

- test the memory circuit:
- first, write 5 to memory line 1
- then, write 7 to memory line 0
- then, clear the input bits and read-out the data from memory line 1 (should be 5)
- then, clear the input bits and read-out the data from memory line 0 (should be 7)
- Give **+12ec points for a fully-working** 12nG bits of memory! **[total of +15pts]**
- If they've made a start, give a proportional amount of credit...