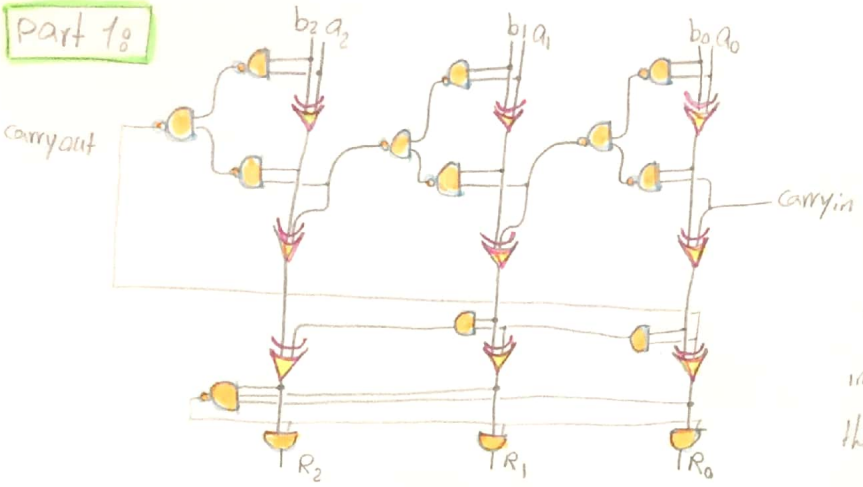
Mohammad Heidary 810197494 CA#4 DLD Dr.Navabi May 27.2020

part 1:



XOR # 18ns NOT # 6ns

NAND(3) # 12ns NAND(2) # 8ns

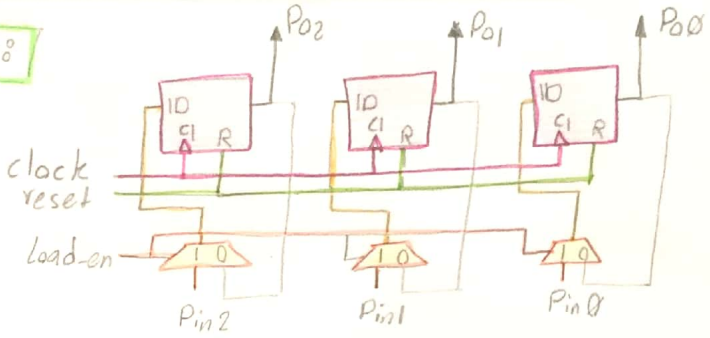
worst case Delay # 18+16+16+16+14 +14+18+12+14 = 38ns

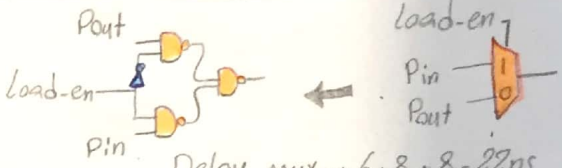
inputs → 3-bit a _ 3-bit b carry in

outputs → 3 bit R

in this part we have a residue-7 to show us the result of operation

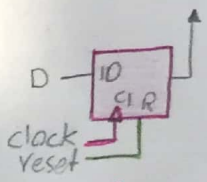
part 2:





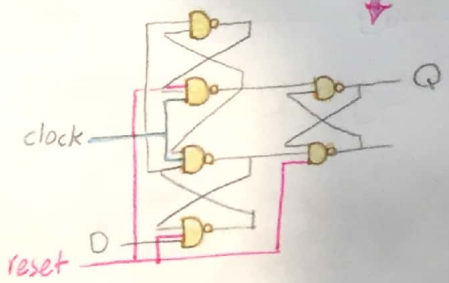
Delay mux # 6+8+8 = 22ns

delay flip flop # 12+12+12+8 = 44ns

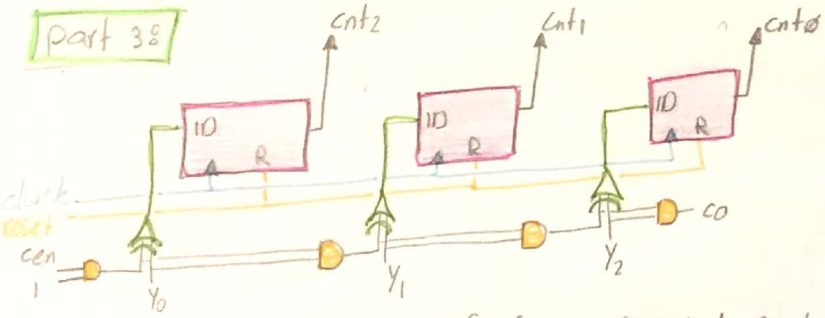


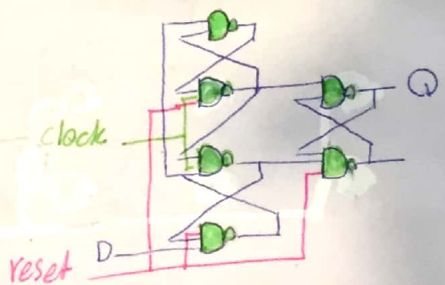
worst case delay = worst case delay Mux + worst case delay FlipFlop = 22+44 = 66ns

in this part we designed a circuit with (inputs → reset-clock-load_en - 3-bit Pin) (outputs → 3-bit Po). this circuit can save our data that we gave.



part 3:





delay d-Flipflop # 12+12+12+8 = 44 ns

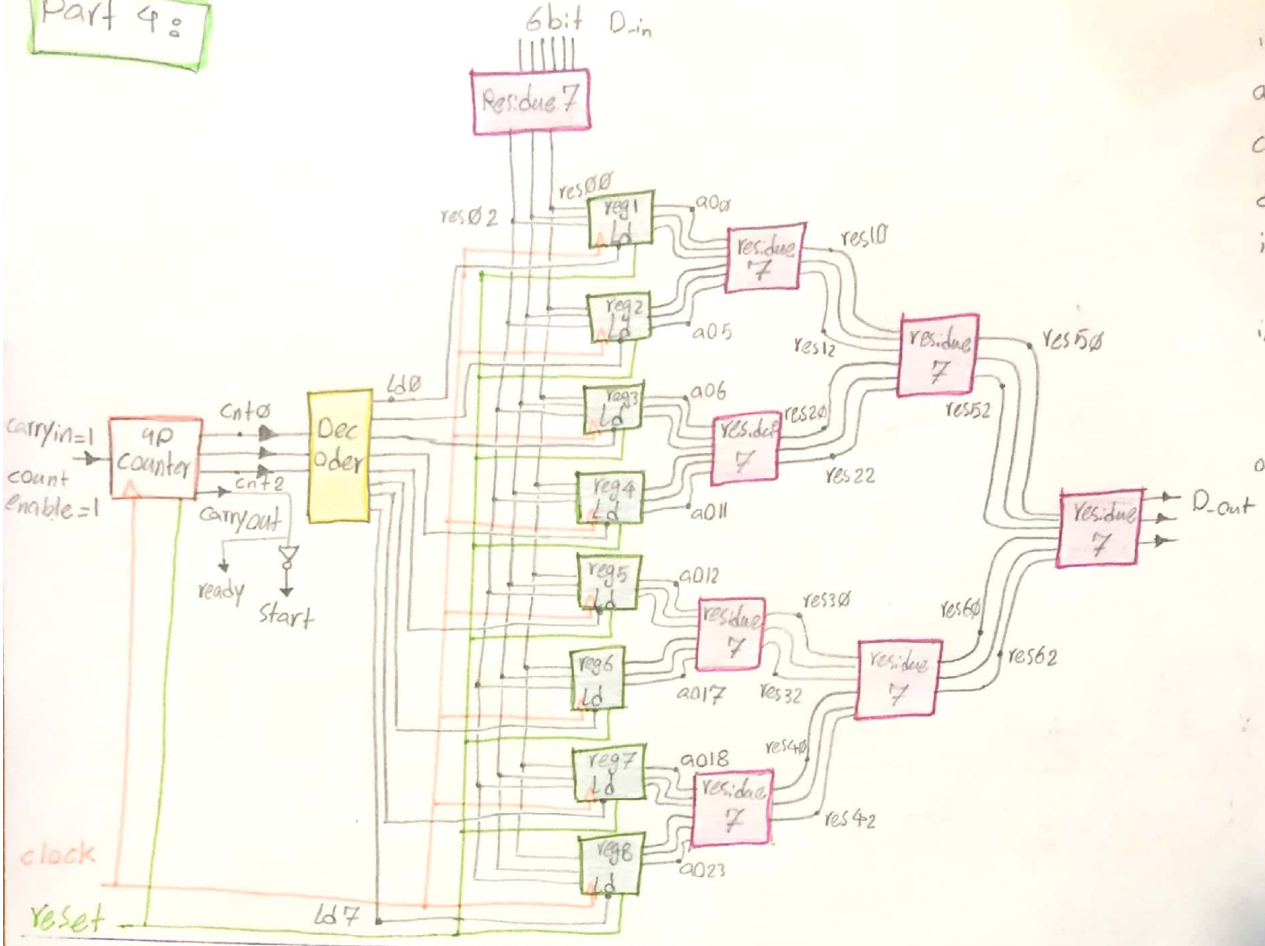
worst case Delay of a flipflop # from last part = 44 ns (co delay # 14+14+14+14 = 56ns)

worst case Delay of this circuit (3-bit up counter) → delay count = 14+14+14+18+44 = 104ns (D-FlipFlop)

In this part we made a 3-bit up counter that can count from 000 to 111 (ø to 7) modulo-8 binary up-counter with (input → count enable = cin _ rising edge clock _ an asychronous reset and output → carry out)

total delay = delay flipflop + dlay XOR gate + and

Part 4:



in this part we designed a 48-bit residue 7 circuit. the 48-bits of data for this circuit come in 6-bit chuncks.

inputs { reset, clock, 6-bit D_in

outputs { 3-bit D_out, start

Part 5: in this part we passed all of 48-bit to the residues with suitable we set each inputs to drive residue. in this case we also set the reset on one cycle. in order to make a unexpensive circuit, we only used a Residue-7 & give our inputs to the residue step by step so we have a residue & 8 register which is controled by a upcounter & a decoder instead of 8 residue-7s. in this part we need to give the 6-inputs of 48-inputs beror press the clock as we need the 0000 for the initial value in our circuit so we press the reset to becomes 1 & D_out becomes 3-bit 0. then we give the inputs, for example first 6-bits= 58 and second 6-bits= 8 & =63, 10, 12, 2, 50. & 9 in dessimal numbers so they are our first residue inputs & output of every 6-bit inputs is 3 bit that named res0. every 3-bits of our out put is given to a single register for storing our result that we need to give to other residues. in this example our final result is 2 =010 (in binary num.). the start signal is 1 when our circuit is ready to show us the final result & it's become 0 when the output shows us the final result.