EDA322/ DIT797: Digital Design
Exam - June 2020

Date: June 10, 2020

Time: **14:00-18:00**

Examiner: Ioannis Sourdis

Department: Computer Science and Engineering

Inquiries: contact through phone, phone extension 1744

Duration: 4 hours

Grading scale: 100 points in total

Chalmers:
0: 0%-49%, 3: 50%-64%, 4: 65%-84%, 5: 85%-100%
GU:
Fail (U): 0%-49%, Pass (G): 50%-79%, Pass with Distinction (VG): 80%-100%

Available references: a calculator, lecture notes, textbook, etc. are allowed.

General: Submit your solutions, in English, on blank paper sheets. Write legibly; feel free to use figures to get your point across.

The order of answering the questions does not matter (start with the easiest ones).

Please start the solutions for each problem on a new sheet. Please number the sheets so that the solutions are in numerical order.

Note that it is possible to receive partial credit for an answer even if it is not 100% correct.

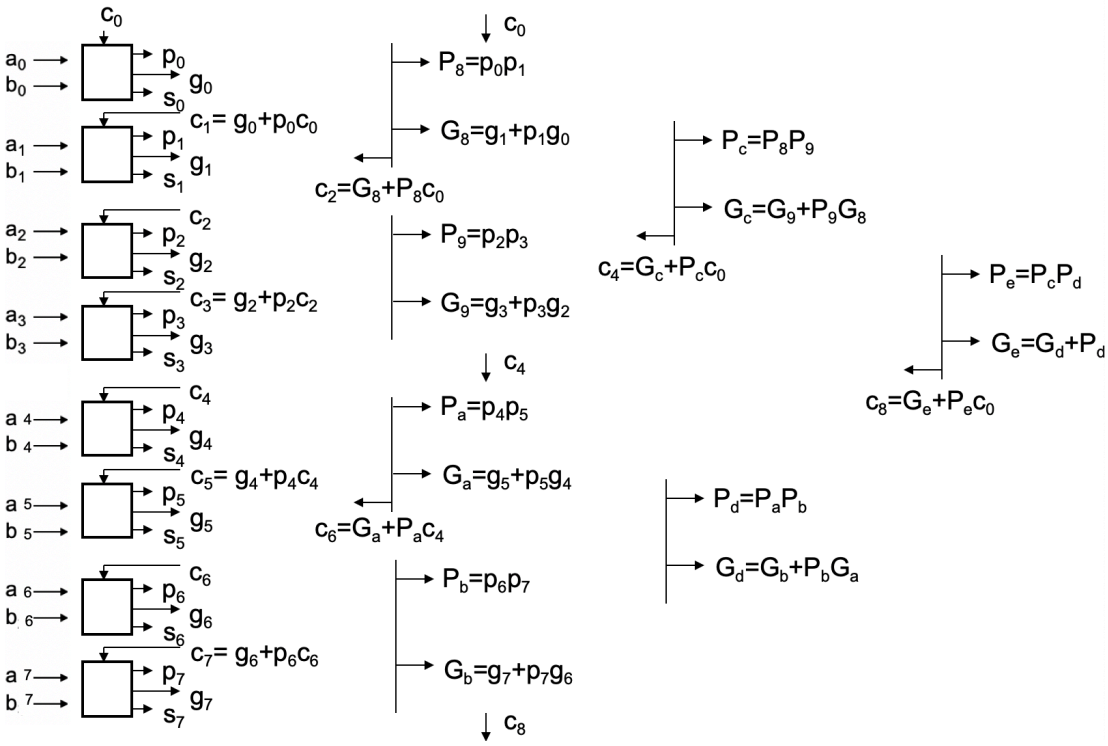
Your personal identity code is required on each submitted sheet!

Good luck!

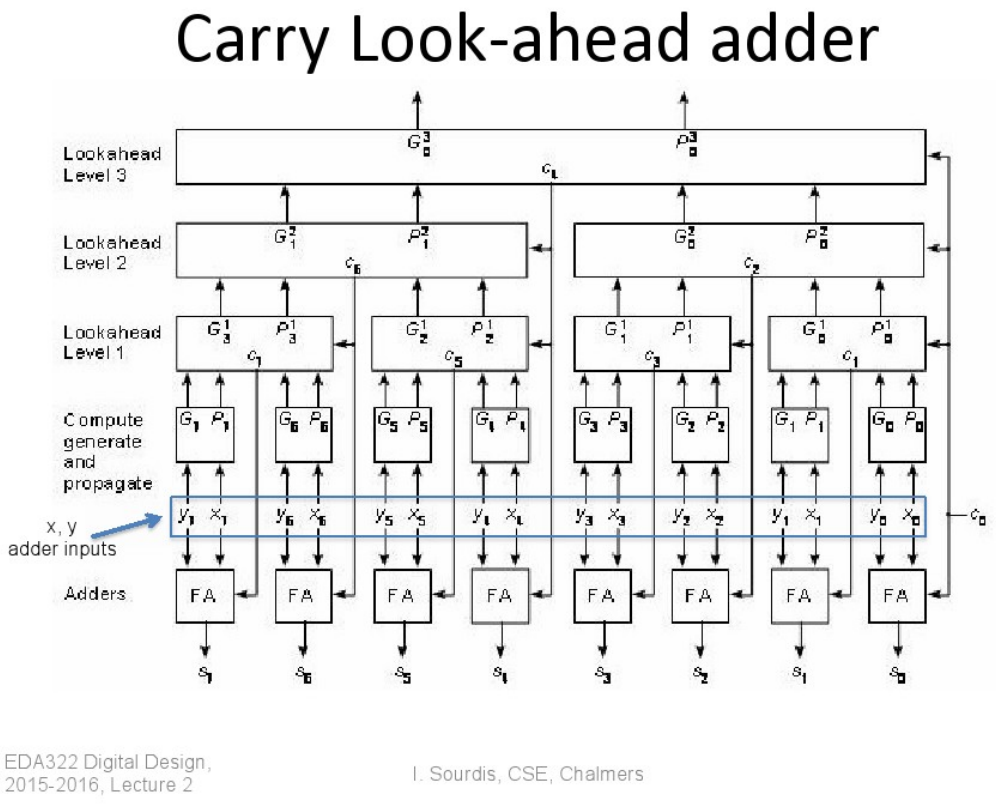
Note: *These are example answers to the exam questions for one possible set of randomized variables.*

Question 1 Arithmetic: (10 points)
Consider a carry lookahead adder of [a]-bits. Show the critical path considering the gate-level representation of the adder. Measure the critical path if the latency a 2-input OR gate is [b], of a 2-input AND gate is [c] and 2-input XOR gate is [d].

Note: the figure bellow shows an 8-bit carry lookahead adder.



Answer



For a 8-bit CLA the critical path is:
from x0, y0 -> G_0, P_0 -> G_0^1, P_0^1 -> G_0^2, P_0^2 -> c4 -> c6 -> c7 -> s7

the Boolean equations of the above stages need to be written and their delays need to be calculated based on the give gate delays.

Question 2 Hazards: (10 points)

1. What causes a hazard in a digital circuit?
2. Does it affect synchronous, asynchronous or both types of sequential circuits? Justify your answer.
3. What ways do we have to mitigate (fix) hazards?

Answer

1. The different delays of different paths from inputs to an output of a circuit.
2. Synchronous are not affected if the output of a circuit goes to a flipflop, because by the time the signal is registered is already stable. Asynchronous circuits can have serious problems with hazards as they may put them to the wrong state.
3. Hazards are fixed by either (i) implementing them in a two level circuit (PoS or SoP) and adding more product or sum terms, respectively, or by adding latches at the output of a circuit or (ii) by rearranging the logic and adding latches.

Question 3 Number representation: (10 points)
Suppose you need to represent temperature from 1 degree Celsius (cm) to 100 degrees Celsius with an accuracy of 5%.

a) Find a fixed point representation that fits the above specification
b) Find a floating point representation that fits the above specification
c) Which type of the above number representations is better and why?

Answer

a) 5% accuracy is $5*10^{-2}$C.
This needs a resolution of $2*5*10^{-2}$= 1/10 C
If Celsius is the integer part we need, then we need 7 bits for integer (to count between 0-100 degrees). Then, the fraction needs 4 bits to count $1/2^4$ = 1/16 C (which is a bit smaller than 1/10 C resolution). So the fixed-point representation would require 11 bits (7.4).
Alternatively, we can count 1/10 Cs which is the required resolution. 100C = 1000 tenths of C for which we need 10 bits (2^{10}=1024).

b) For a floating-point representation, the mantissa needs to be only 4 bits to offer accuracy of 5%. So, the resolution needs be double that: 10% = 1/10; 1/16= $1/2^4$ should then be enough (4 bits mantissa). The dynamic range is from 1 C to 10^2 C so it is 10^2, which is smaller than 128 = 2^7. Then, 3 bits are enough to count up to 7, so with 3-bits exponent we can represent a range of 2^7. We finally set the bias to zero (0) because we need the exponent to go up to 2^7. So, the floating point format is 4 bits mantissa and 3 bits exponent, 7 bits in total.

c) Floating point needs fewer bits than fixed point, so it is a better choice. That happens because the required accuracy is a percentage and not a fixed absolute value.

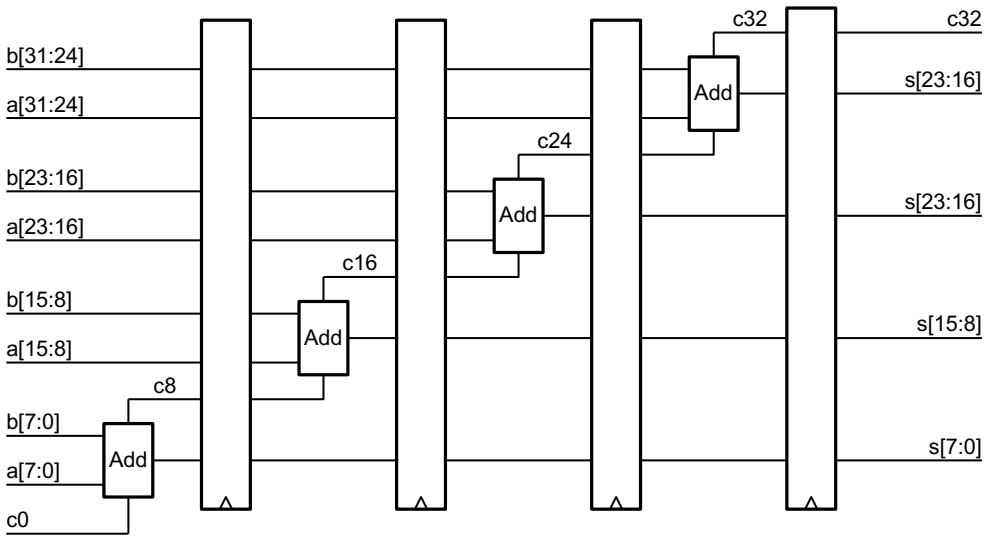
Question 4 Pipeline: (10 points)
Consider the pipelined adder of the figure that adds two numbers a and b and has latency L and throughput T.
Modify the pipelined adder to add 3/4 numbers (a, b, c, / a, b, c, d)

- so the throughput of the new adder remains T and latency is not longer than 5*L/4 .
- so the throughput of the new adder is 2*T and latency is not longer than 5*L/4 .
- so the throughput of the new adder is 4*T and latency is not longer than 5*L/4 .

Consider the following:

- the adder block in the figure is a ripple carry adder

- a FA has a fixed delay d
- setup and propagation delays of flipflops are too small and can be considered zero when calculating latency.



Answer
If 4 numbers are added then we need to double the above design for each pair (a,b) and (c,d), then the partial sums (if 4 numbers added) are added in the next stage.

If 3 numbers are added then c is added to the partial sum of a+b in the next stage.

In both cases one stage is added.

That maintains throughput T because the latency of the longest stage remains the delay of an 8-bit adder. The latency of the design is now 5 stages instead of 4 increasing overall latency to 5L/4.

For increasing throughput to 2*T or 4*T each stage needs to be split to 2 or 4 stages, respectively, each stage adding 4-bits or 2-bits of the numbers. The overall latency will remain 5L/4 because although the stages increase by 2x or 4x, their latency is reduced by ½ or ¼, respectively.

Question 5 FSMs: (10 points)
Design a Moore FSM for a vending machine that receives coins of type-1 (5SEK) and type-2 (10 SEK) and opens when it receives the total amount of 20SEK. If it receives a larger amount it resets and gives back all the coins it has collected so far. The inputs COIN1 and COIN2 are set when coins of type-1 and type-2 are received, respectively. When the correct amount is received, the output open (O) is set to "1" and the machine is reset. The output returned-back (B) is set to "1" and all coins are returned when a larger amount than the expected one is received.

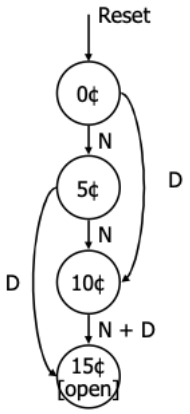
Note that two coins cannot be received at the same time. In addition, when the correct amount is collected, no more coins can be received until the machine resets again.

Make the minimal state diagram and state table, state assignment and implementation of the FSM drawing the gatelevel circuit using D-flip-flops.

Alternative values:
Coins of 1 and 2, to sum 5
Coins of 2 and 5 to sum 10
Coins of 2 and 5 to sum 12
Coins of 5 and 10 to sum 25
Coins of 5 and 10, to sum 20

Answer

Similar to the example bellow, with the change that receiving a coins that makes the sum exceeding the expected amount gets the machine to the initial state where B=1.



present state Q0 Q1	inputs D N	next state P1 P0	present output
0 0	0 0	0 0	0
	0 1	0 1	0
	1 0	1 0	0
	1 1	– –	–
0 1	0 0	0 1	0
	0 1	1 0	0
	1 0	1 1	0
	1 1	– –	–
1 0	0 0	1 0	0
	0 1	1 1	0
	1 0	1 1	0
	1 1	– –	–
1 1	– –	1 1	1

K-map for P1

DN \ Q1Q0	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	1	1

D, N, Q0

K-map for P0

DN \ Q1Q0	00	01	11	10
00	0	1	1	0
01	1	0	1	1
11	X	X	X	X
10	0	1	1	1

D, N, Q0

K-map for Open

DN \ Q1Q0	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	1	0

D, N, Q0

$P_1 = Q_1 + D + Q_0N$

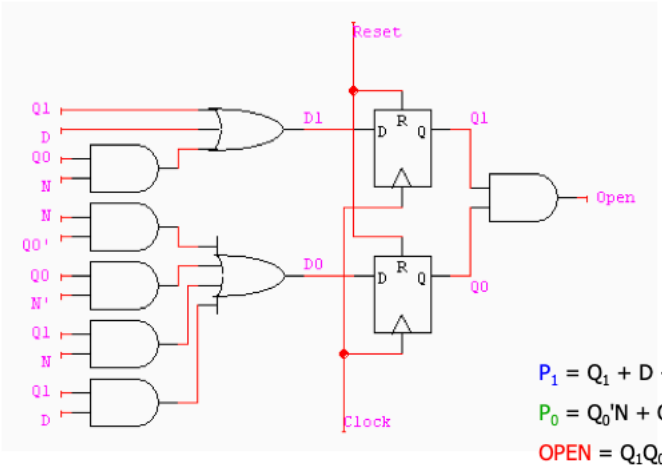
$P_0 = Q_0'N + Q_0N' + Q_1N + Q_1D$

$OPEN = Q_1Q_0$

if FFs do not have a reset pin then

$P_1 = reset'(Q_1 + D + Q_0N)$

$P_0 = reset'(Q_0'N + Q_0N' + Q_1N + Q_1D)$



$P_1 = Q_1 + D + Q_0N$

$P_0 = Q_0'N + Q_0N' + Q_1N + Q_1D$

$OPEN = Q_1Q_0$

Question 6 Faults: (10 points)

The manufacturing process of a factory that fabricates chips has a yield of 80% and a defect level of 10 parts per million (ppm). How much is the average cost of single good chip considering that:

- A wafer that contains 100 chips costs 10 dollars, and
- The cost of selling a faulty chip to a customer is 1000 dollars?

Answer

A wafer makes 80 good chips that cost 10 dollars, so the wafer cost per good (based on the test process) chip is 0,125 dollars.

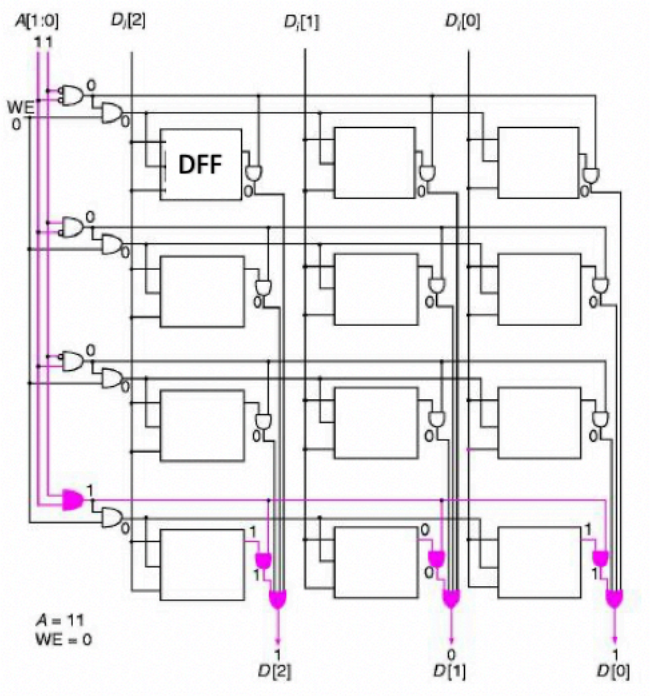
Selling 1 000 000 chips will end up selling 10 faulty ones because the test process is not perfect (DL = 10 ppm), which costs 10 000 dollars. That is on average 0,01 dollar cost per sold chip on average.

So the cost of a sold chip should the sum of its costs based on the wafer and based on the DL: 0,125+0,01 = 0,135 dollars

Question7 Memory: (10 points)

1) Consider the memory block of the figure. What is the highest operating frequency it can operate at. Consider that:

- A 2-input AND gate has a delay of x
- A 2-input OR gate has a delay of y
- The D-flipflip setup time is z
- The D-flipflip hold time is j
- The D-flipflip propagation time is k



2) What will be the operating frequency if the number of memory rows doubles (2x)?
3) What will be the operating frequency if the number of memory columns doubles (2x)?

Answer
1) all paths from input to the DFF, input to output (paths that don't go through DFF), and from DFF to output need to be analyzed according to the given gate delays.
2) larger number of rows increases the delay of the decoder and the output OR gate.
3) multiple columns do not affect timing

Question8 Sequential circuits: (10 points)
What is the difference between a Synchronous and an Asynchronous sequential circuit?
Is an SR latch a synchronous or an Asynchronous circuit, and why?

Answer
The operation of the synchronous sequential circuits are synchronized with a clock or pulse, while the asynchronous circuits do not have a synchronization mechanism.
An SR latch is an asynchronous sequential circuit because it doesn't have a clock and it has feedback signal loops.

Question 9 Reconfigurable Technologies: (10 points)

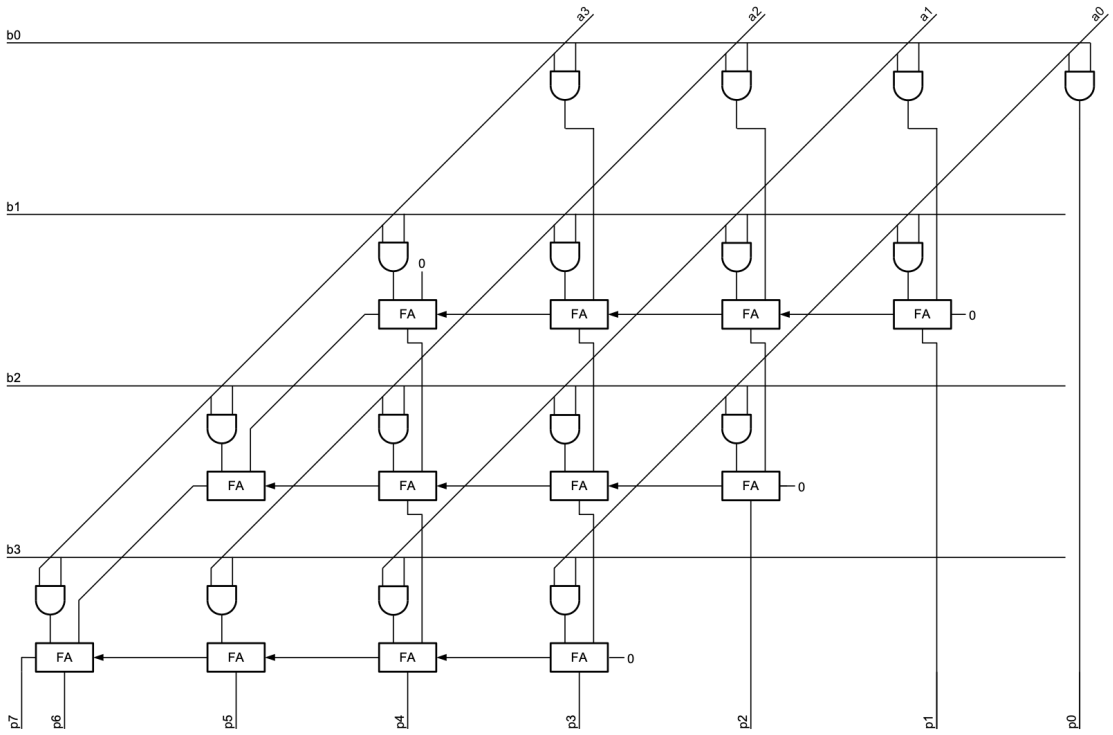
- Why Reconfigurable hardware is more flexible than ASIC and less flexible than software?
- Why Reconfigurable hardware can be faster than software and is slower than ASIC that offers the same computations?
- How does it compare the time to develop a system in reconfigurable hardware with the time to develop a software and the time to develop an ASIC for the same functionality?
- How does it compare in terms of cost to develop a system in reconfigurable hardware with the cst to develop software and the cost to develop an ASIC for the same functionality?
  Explain your answers.

Note: repeating what is stated in the slides without explanation will not be regarded as correct. A more insightful answer is requested.

Answer

- Reconfigurable hardware is more flexible than ASIC because it can be reconfigured to implement any arbitrary hardware design described in a hardware description language (e.g. VHDL), while ASIC cannot be changed after fabrication. It is less flexible than software because it is more difficult to reconfigure it than just changing a software program and recompiling.
- In many cases it is faster than software because it offers the spatial computing of hardware, which offers more parallelism than the temporal computing of software. It is slower than an ASIC implementing the same computations (Boolean functions) because offering reconfigurability makes the circuit slower: In other words mapping the logic of a circuit in LUTs and already fabricated wires is slower than a custom or semicustom ASIC implementation of the circuit.
- ASIC development is the slowest as it requires many long steps of VLSI design and implementation to fabricate a chip. Development of a system in Reconfigurable hardware is shorter as it requires development of the hardware description (e.g. in VHDL language) and then mapping it to an existing FPGA device, which is faster than ASIC chip fabrication. Software development is even faster as it involves only programming (and debugging which is easier than in ASIC or in reconfigurable hardware).
- ASIC development is the most expensive as it has a high non-recurring engineering cost. FPGAs come next as they include (besides the hardware description) the cost of the FPGA device (much lower than ASIC fabrication). Software is the cheapest as it can run in existing computers so it only has the programming-engineering costs and no cost for hardware (assuming a general purpose computer or cloud access is already available).

Question 10 Arithmetic & Timing: (10 points)
The array multiplier of the figure below multiplies to unsigned 4-bit numbers: a*b. Modify the design adding an input bit NEG, when NEG=1 the multiplier should output the 2's compliment of a*b , -(a*b), when NEG=0 it should output a*b.
What is the critical path of the new design?

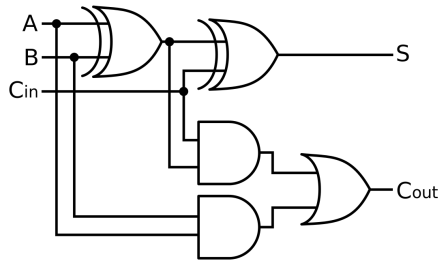


Describe from which inputs it starts, through which gates/modules it goes through, and were it ends.

Consider that a 2-input XOR gate has a delay of [x] ns and a 2-input AND/OR gate have a delay of [y] ns.

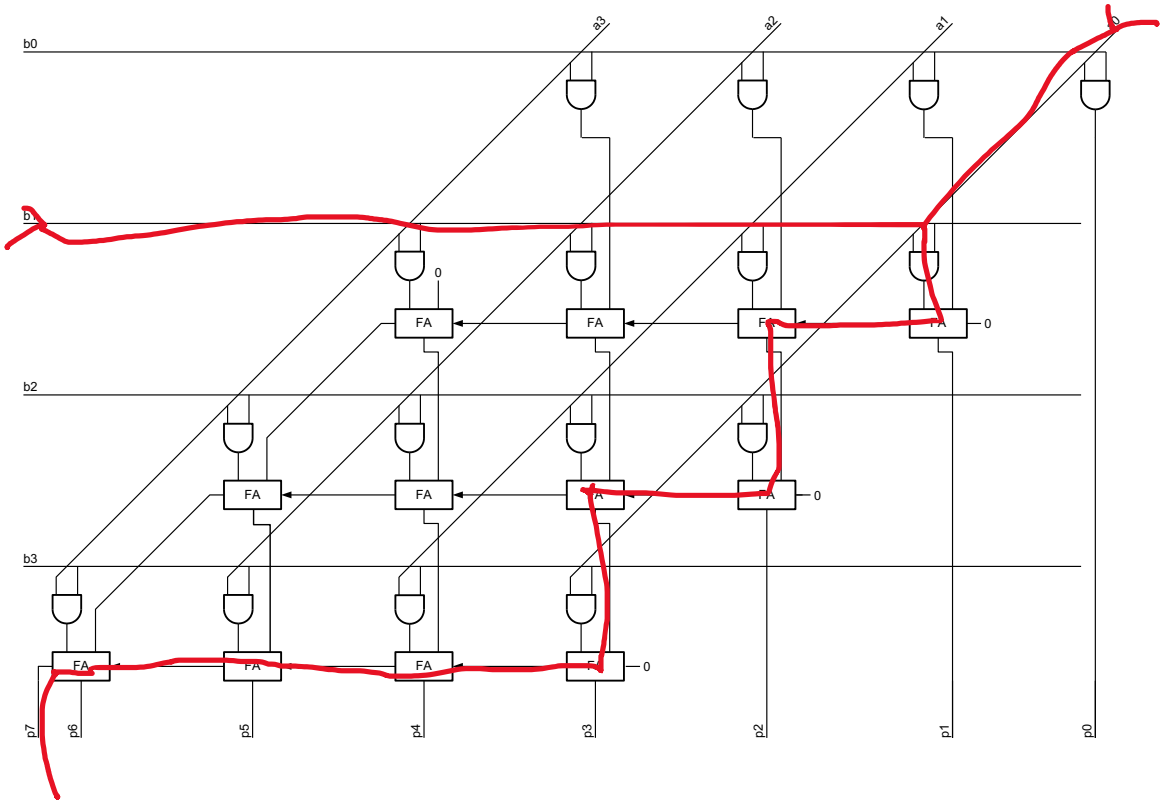
How much longer is the delay of the new critical compared to the initial design?

You can also consider the following full adder circuit:



Answer:
Additions needed is a FA to add a0*b0 with NEG, its output is the new P0 and its carry out goes as a carry in to the existing FA that adds a0*b1 and a1*b0. Then all bits p0…p7 need to be XORed with NEG.

The added delay to the critical path shown bellow is 2*x+2*y.



END of EXAM