EDA321/EDA322: Digital Design
Exam - March 2015

Date: March 18, 2015

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

Examiner: Ioannis Sourdis

Department: Computer Science and Engineering

Inquiries: Ioannis Sourdis (extension 1744); will visit the room at **15:30** and at **17:00**

Results and grading review: See me in my office 4110 on **April 20th at 10:00**.

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: Blank paper and a calculator are allowed. No textbooks or lecture notes, etc. 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!

Question 1: (10 points)

Minimize the cost of function F (x3,x2,x1,x0)= Σ(3,4,8,11,14,15)+D(0,1,7,9,12,13) using Quine-McCluskey (x0 is the list significant bit, x3 is the most significant bit).

Measure the cost of the minimized function by counting the total number of 2-input gates of the circuit (inversions are not considered) e.g., a*b + c*d, has cost of 3.

Answer:

Truth table for the minterms Σ(3,4,8,11,14,15):

i	x3	x2	x1	x0	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Truth table for the don't care terms D(0,1,7,9,12,13):

i	x3	x2	x1	x0	h*
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

The minterms and don't care terms are put together:

Step 1:

Indexgruppe	Index
0	0 *
1	1 * 4 * 8 *
2	3 * 9 * 12 *
3	7 * 11 * 13 * 14 *
4	15 *

step2:

Indexgruppe	Index
0/1/2	
1/2/3	11,10,9,8 (1,2) * 13,12,9,8 (1,4) * 11,9,10,8 (2,1) * 14,12,10,8 (2,4) * 13,9,12,8 (4,1) * 14,10,12,8 (4,2) *
2/3/4	15,13,11,9 (2,4) * 15,11,13,9 (4,2) * 15,14,11,10 (1,4) * 15,11,14,10 (4,1) * 15,14,13,12 (1,2) * 15,13,14,12 (2,1) *
3/4/5	31,27,15,11 (4,16) P1 31,15,27,11 (16,4) 31,27,23,19 (4,8) P2 31,23,27,19 (8,4)

step3:

Indexgruppe	Index
0/1/2	9,8,1,0 (1,8) P1 12,8,4,0 (4,8) P2 9,1,8,0 (8,1) 12,4,8,0 (8,4)
1/2/3	11,9,3,1 (2,8) P3 11,3,9,1 (8,2) 13,12,9,8 (1,4) P4 13,9,12,8 (4,1)
2/3/4	15,11,7,3 (4,8) P5 15,7,11,3 (8,4) 15,13,11,9 (2,4) P6 15,11,13,9 (4,2) 15,14,13,12 (1,2) P7 15,13,14,12 (2,1)

P/M1	3	4	8	11	14	15
P1			*			
P2		*	*			
P3	*			*		
P4			*			
P5	*			*		*
P6				*		*
P7					*	*

P1= x2'x1'
P2= x1'x0'
P3= x2'x0
P4=x3x1'
P5=x1x0
P6= x3x0
P7=x3x2

F = P2+P3+P7 = x1'x0' + x2'x0 + x3x2
or
F = P2+P5+P7 = x1'x0'+ x1x0 + x3x2

Cost is 5.

Question 2: **(15 points)**

a) Draw the state diagram of a counter that has the following characteristics:

- 3-bit output (takes values 0-7),
- has a reset input that puts the counter value to zero,
- has an input *cnt_up* (1 bit).
    - If *cnt_up=1* then the counter counts up from 0 towards 7 and when reaching value 7 goes back to 0: 0->1->2->3->4->5->6->7->0->…
    - If *cnt_up=0* then the counter counts down from 7 towards 0 and when reaching value 0 goes back to 7: 0->7->6->5->4->3->2->1->0->…

Justify your choice of FSM type selected. (3 points)

b) Draw the state table and assign values to the states (state assignment). Select a state encoding and justify your choice. (3 points)

c) Retrieve the Boolean functions that implement the counter using D Flip-flops and draw the circuit of the counter at the gate-level (no need to show the internals of each flip-flop in your drawing). (9 points)

Answer:

Solution of book exercise 7.17 (although different type of FSM and state encoding could be used).

Question 3: (15 points)
Draw the block diagram of a multiplier that performs multiplications of a 4-bit value B(3:0) with the constant "5" (The multiplier output is *Mul_out = B*5*) .

The multiplier should use a ripple-carry adder. On a separate drawing show a gate-level diagram of a single-bit full adder and, using this full adder as a building block, show the block diagram of the ripple carry adder.

Answer:
Lecture 11 slide 6 and Lecture 2 slide 73.

(Above mentioned lecture slides can be found at the end of this document)

Question 4: (10 points)

a. Name the two general types of Volatile Memory technologies. (2 points)
b. What are the main differences between the two Memory types? (4 points)
c. Explain the Processor-Memory performance gap. (4 points)

Answer:
Lecture 11 slides 43-44.

(Above mentioned lecture slides can be found at the end of this document)

Question 5: (15 points)
a) Explain the differences between the following three computing alternatives:

- ASIC
- Reconfigurable hardware (FPGA)
- Software running on a General Purpose Processor

(4 points)

b) Explain how the following equation would be computed in hardware (spatial-based execution) and in software (temporal-based execution):

$F=A*x^2 + B*x + C,$

where:

- For hardware, A, B, C, and x are inputs and F is output.
- For software, A, B, C, and x are values in memory and F is value to be stored back to memory.

Show with a drawing how the above computation would be performed in hardware. Show also how (assembly-like) instructions would be used to compute the same equation in software on some ALU that is able to perform additions and multiplications. You can use an assembly format of your choice or pseudocode that perform the following operations:

- Loads values to registers, e.g. LD $tmp1, A
- Additions, e.g., ADD $tmp2, $tmp2, B
- Multiplications, e.g., MUL $tmp3, $tmp3, C
- Stores registers to memory, e.g., ST D, $tmp3

(7 points)

c) When would computing in hardware be better than computing in software? (4 points)

Answer:
Lecture 12 slides 3-4.
(temporal computation is not necessary to be shown with the exact instructions in slide 4 or the ALU diagram)

(Above mentioned lecture slides can be found at the end of this document)

Question 6: **(10 points)**

a) Draw the block diagram of a 4-input FPGA logic cell and show the logic cell's configuration bits on it. (5 points)
b) What would be values of the configuration bits in the above logic cell in order to implement the function F= (x1*x2)+(x3*x4), with the output of function F being registered. (5 points)

Answer:
Lecture 12 slides 16 and Lecture 12 similar to slide 18.

(Above mentioned lecture slides can be found at the end of this document)

Question 7: **(7 points)**

a) What are the parameters that affect the gate delay? (3 points)
b) What is the relation between the length and the delay of a wire? (1 point)
c) Describe the timing constraints of a flip-flop (3 points)

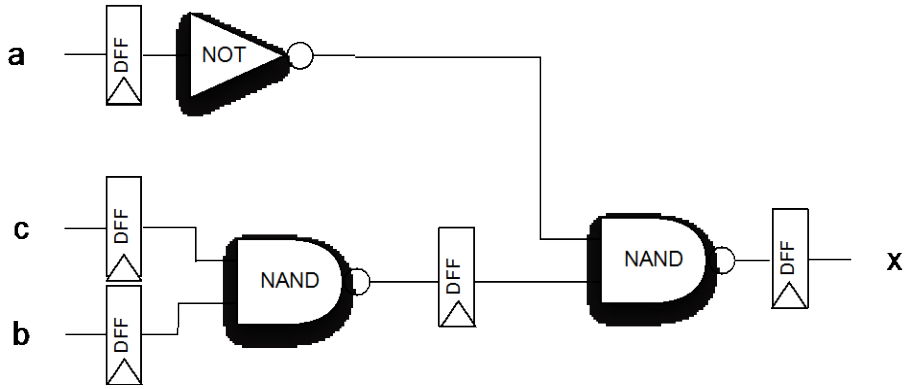
Answer:
Lecture 16 slides 9, 17, and 18.

(Above mentioned lecture slides can be found at the end of this document)

Question 8: **(8 points)**
Calculate the maximum delay of the circuit (critical path delay), considering the following:

- a NOT gate has a delay of 1 ns
- a NAND gate has a delay of 2 ns
- Propagation time (clock to output) for a flip-flop 1ns
- Setup time for a flip-flop 2ns
- Wires have zero delay

- All inputs (a, b, c) have zero delay

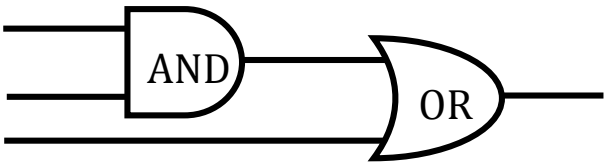


Answer:

From a to x:
(DFF propagation delay) + (NOT delay) + (NAND delay) + (DFF setup time) = 1ns +1ns + 2ns + 2ns = 6 ns

Question 9: (10 points)

a) What is the total number of single stuck-at faults, counting both stuck-at-0 and stuck-at-1, in the following circuit? (5 points)
b) Which faults are left after collapsing equivalent faults? What is the ratio between the final number of faults that need testing and the initial set of faults? (5 points)



Answer:
Similar to Lecture 13 slides 30, 33, 34. 6/10 = 60%

(Above mentioned lecture slides can be found at the end of this document)

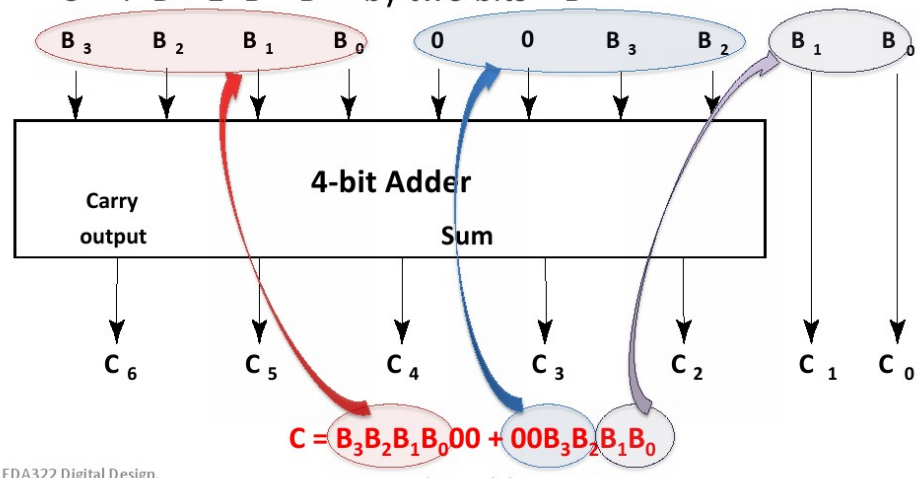
END of EXAM

Lecture Slides

Question 3:

Multiplication by a Constant

- Multiplication of B(3:0) by 101
- C = 4*B + 1*B = B<< by two bits + B

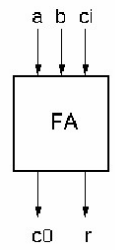


Ripple-Carry Adder

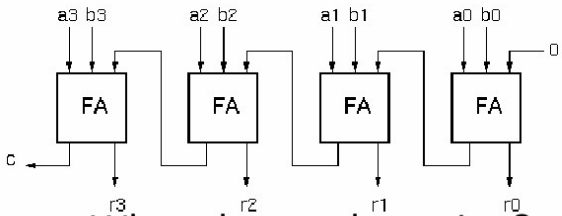
- Each cell:

  $r_i = a_i$ XOR b_i XOR c_{in}

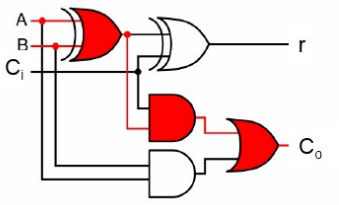
  $c_{out} = c_{in}(a_i$ XOR $b_i) + a_i b_i$



- 4-bit adder:



"Full adder cell"



- What about subtraction?

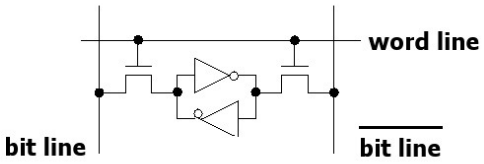
Question 4:

Example Memory Components:

- Volatile:
  - Random Access Memory (RAM):
    - DRAM "dynamic"
    - SRAM "static"
- Non-volatile:
  - Read Only Memory (ROM):
    - Mask ROM "mask programmable"
    - EPROM "electrically programmable"
    - EEPROM "erasable electrically programmable"
    - FLASH memory - similar to EEPROM with programmer integrated on chip

Volatile Memory Comparison

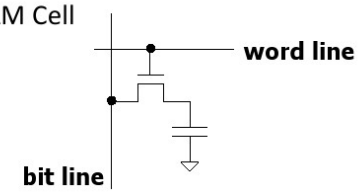
- SRAM Cell



- Larger cell ⇒ lower density, higher cost/bit
- No refresh required
- Simple read ⇒ faster access
- Standard IC process ⇒ natural for integration with logic

- **Read:** Pre-charge bit-lines, Raise wordline, cell pulls one bit line low, sense the difference
- **Write:** Drive data onto bit-lines (one 0, one 1), Raise wordline

- DRAM Cell

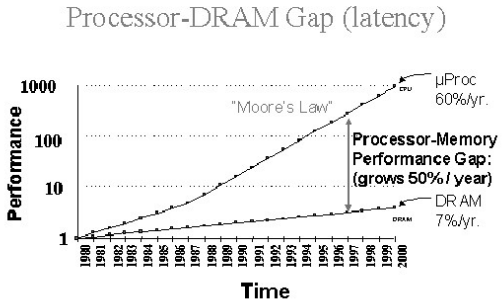


- Smaller cell ⇒ higher density, lower cost/bit
- Needs periodic refresh, and refresh after read
- Complex read ⇒ longer access time
- Special IC process ⇒ difficult to integrate with logic circuits

- **Read:** Pre-charge bit-lines with Vdd/2, Raise wordline, bit-line gets the read value, value sensed, and written back to the cell
- **Write:** put the value to write in the bit-line, Raise wordline
- **Refresh:** a dummy read to every cell.

In Desktop Computer Systems:

- **SRAM** (lower density, higher speed) used in CPU register file, on- and off-chip caches.
- **DRAM** (higher density, lower speed) used in main memory



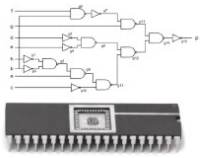
Closing the GAP: Innovation targeted towards higher bandwidth for memory systems:

SDRAM - synchronous DRAM
RDRAM - Rambus DRAM
EDORAM - extended data out SRAM
3D-stacked DRAM
hyper-page mode DRAM video RAM
multibank DRAM

Question 5:

Computing alternatives

Hardware (Application Specific Integrated Circuits)



Advantages:
• very high performance and efficient
Disadvantages:
• not flexible (can't be altered after fabrication)
• High NRE Cost

Reconfigurable computing

Advantages:
•much higher performance than software / lower performance than ASIC
• higher level of flexibility than hardware / more difficult to program than SW

•**fills the gap between hardware and software**

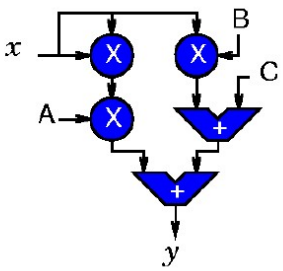
Software-programmed processors

Advantages:
• software is very flexible to change
Disadvantages:
• performance can suffer if clock is not fast
• fixed instruction set by hardware

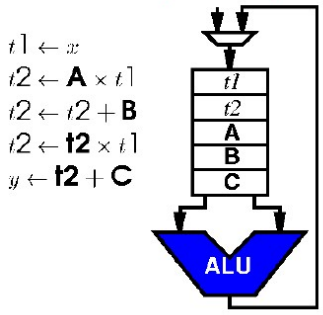
Spatial vs. Temporal computing

$F = Ax^2 + Bx + C$

Spatial-based execution (hardware)



Temporal-based execution (software)

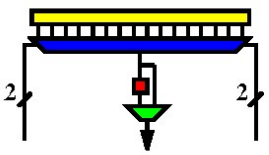


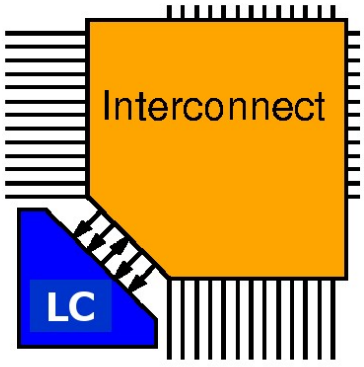
- Extract parallelism (or concurrency) from algorithm descriptions is one of the keys to acceleration using reconfigurable computing

Question 6:

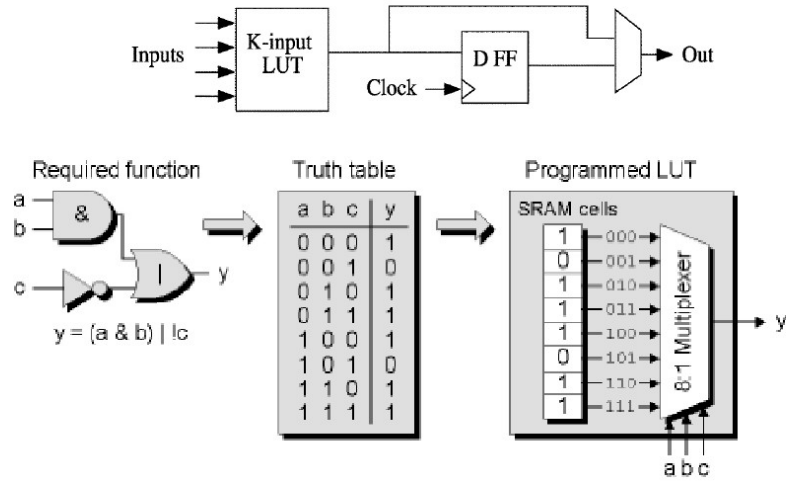
Logic Cells: Conventional FPGA Tile

K-LUT (typical k=4) w/ optional output Flip-Flop





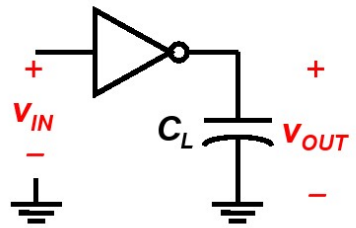
Mapping logic to a Logic Cell

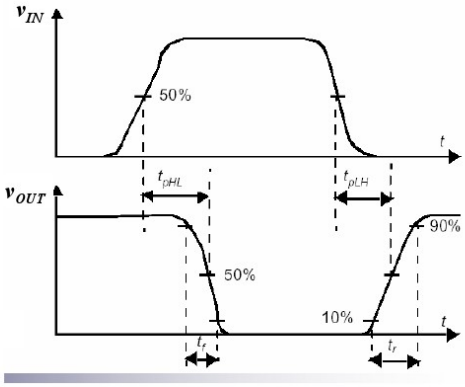


Question 7:

Gate Delay

- The time needed for the output of a gate to change from the moment an input of the gate changes
- Depends on:
  - Transistor parameters
  - Fan-out: how many wires it will drive
  - Fan-in: number of inputs of a gate

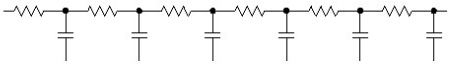




Wire Delay

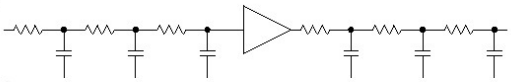
Even in cases where the transmission line effect is negligible:

- Wires posses distributed resistance **R** and capacitance **C**



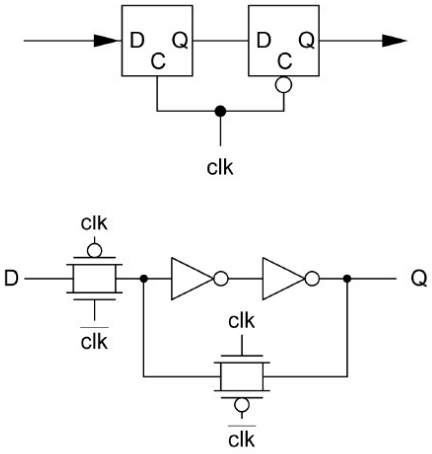
- Time constant associated with distributed RC is proportional to the ***square* of the length $O(L^2)$**
- For short wires on ICs, resistance is insignificant (relative to effective R of transistors), but C is important.
- Typically around half of C of gate load is in the wires.

- For long wires on ICs:
  - busses, clock lines, global control signal (e.g. reset), etc.
  - distributed RC (and therefore long delay) significant
  - For long wires signals need to be "rebuffered" which contributes in the delay, too.



Wire delay is proportional to the ***square* of its length $O(L^2)$**

Delay in Flip-flops



T_{su}= setup time
(the time the input of a flip flop needs to be stable before the clock edge to be stored correctly)

T_h= hold time
(the time the input of a flip flop needs to be held after the clock edge to be stored correctly)

T_p=propagation time (clock to Q)
(the time a flip-flop needs to propagate a value stored to the output)

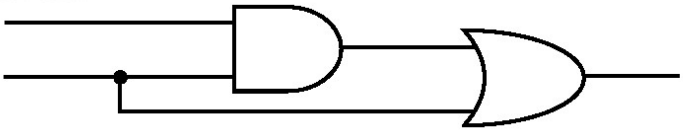
Question 9:

Quiz 13-1

http://m.socrative.com/student/#joinRoom

room number: 713113

- Q1: What is the yield of a wafer that has 3 out of 10 chips defective:
  a) 30%
  b) 70%
  c) 130%
- The cost of a chip is $1.00 when its yield is 50%. What will be its cost if you increased the yield to 80%.
- What is the total number of single stuck-at faults, counting both stuck-at-0 and stuck-at-1, in the following circuit?
- Which faults are left after equivalence fault collapsing? What is the collapse ratio?



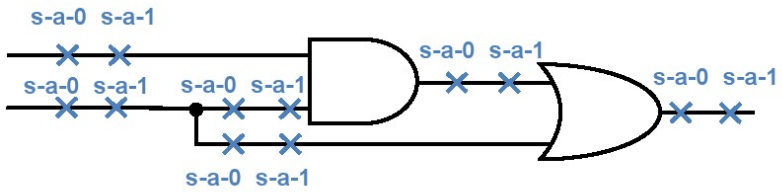
Answers Continued

- What is the total number of single stuck-at faults, counting both stuck-at-0 and stuck-at-1, in the following circuit?

Counting two faults on each line,

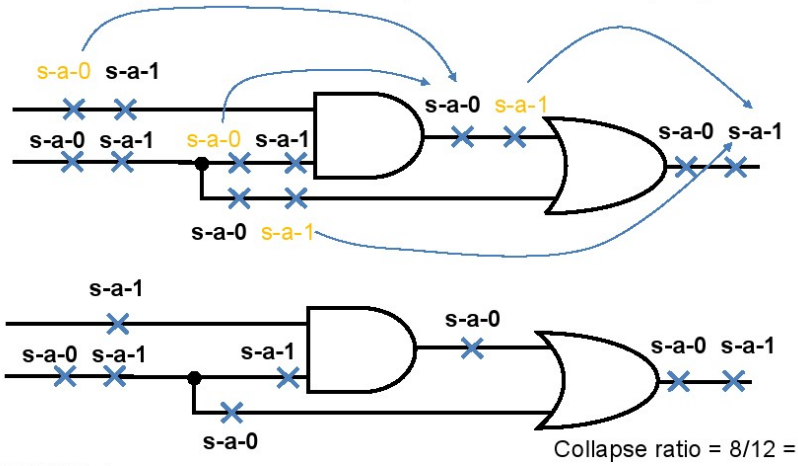
Total number of faults = 2 × (#PI + #gates + #fanout branches)

= 2 × (2 + 2 + 2) = 12



Answers Continued

- How many faults are left after equivalence fault collapsing?



Collapse ratio = 8/12 = 0.67