EDA322/ DIT797: Digital Design
Exam - August 2018

Date: August 30, 2018

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

Examiner: Ioannis Sourdis

Department: Computer Science and Engineering

Inquiries: visiting the room at **15:30** and at
17:00
(contact through phone: phone extension 1032)

Results and grading review: room 4128 EDIT on **Sept 21st at 11:00**.

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: a calculator is 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.

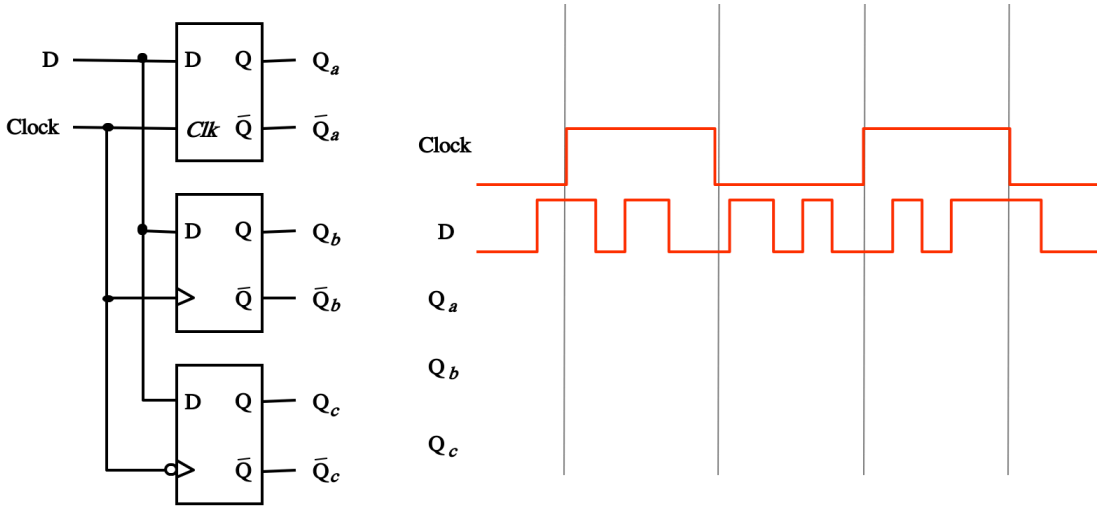
Your personal identity code is required on each submitted sheet!

Good luck!

Question 1 Sequential Circuits: (10 points)

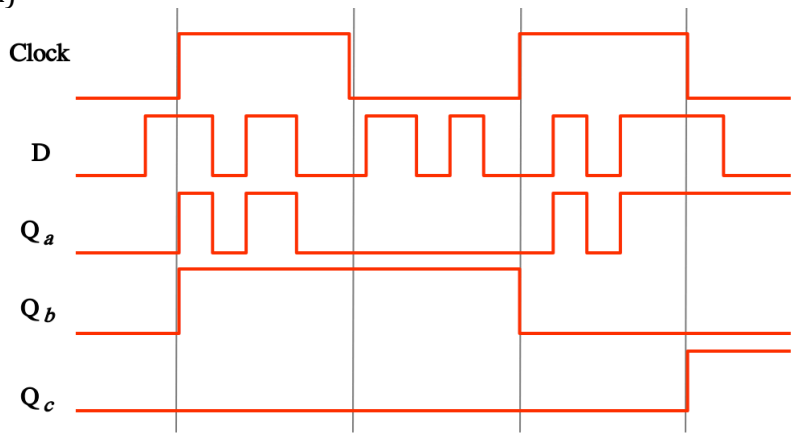
a) Draw I the waveform below the output of the following latch (Q_a) and D-flipflops (Q_b, Q_c) according to the input D below:

b) What are the timing requirements for a D-flipflop to operate correctly?

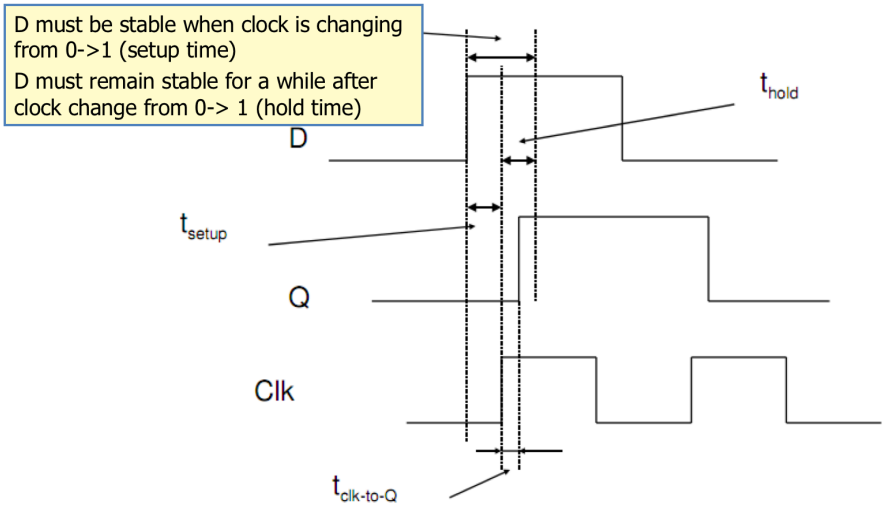


Answer

a)



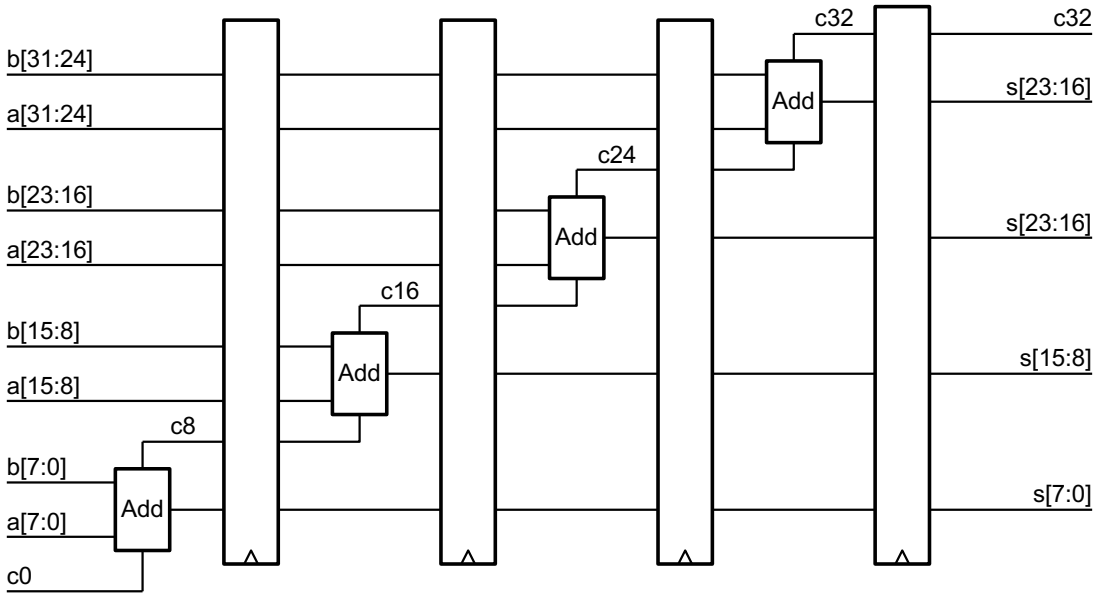
b)



Question 2 Pipelining: (10 points)
Consider a 32-bit ripple-carry adder and its pipelined version (shown below), pipelined in 4 stages (8 bits ripple carry addition in each stage).

Calculate the latency and throughput of the unpipelined and pipelined version of the adder considering that:

- An 1-bit addition has a latency of 100 ps. Consider that the latency of the ripple carry adder increases linearly to the number of bits (an n-bit ripple carry adder has a latency of n*100ps)
- The setup time of a D-flipflop is 80ps
- The propagation time of a D flip-flop is 120 ps



Answer

- Suppose before pipelining the delay of our 32b adder is 3200ps (100ps per bit) and this adder can do one problem each 3200ps for a *throughput* of 1/3200ps = 312 Millions of operations (additions) per second (Mops)

- What is the delay (latency, t_{pipe}) and throughput (Θ) of the adder with pipelining?
  Suppose td_{CQ} = 120ps, t_{setup} = 80ps, (200ps Overhead)

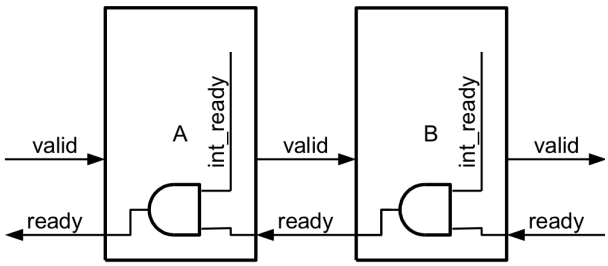
$t_{pipe} = n(t_{stage} + t_{dCQ} + t_s) = 4(800 + 200) = 4000$

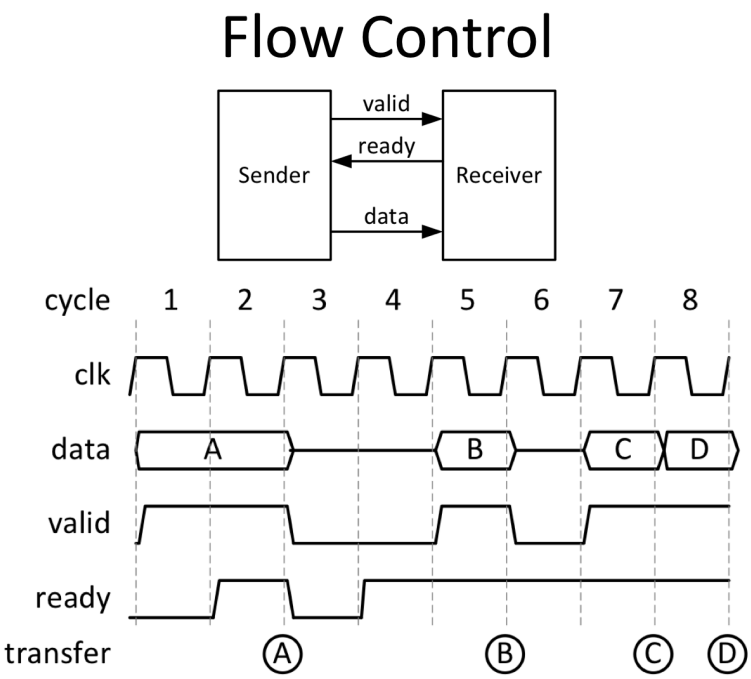
$\Theta = 1/(t_{stage} + t_{dCQ} + t_s) = 1/1000 = 1\text{Gops}$

Question 3 Interfaces: (10 points)
What kind of interface is used between two pipeline stages so to allow pipeline stalling. Show the timing.

Note: no need to explain the mechanism (e.g. double buffering) just the signals used at the interface.

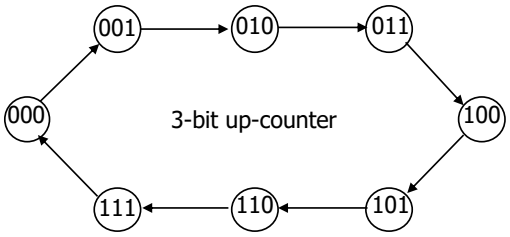
Answer
It's a full flow control interface. The previous stage sends data with a valid signal when the next stage indicates it is ready (the ready signal is the logic AND of the internal ready and the ready signal coming from the next stage):





Question 4 Sequential circuits: (10 points)
Design and implement a 3-bit binary counter using D-flip-flops. Show the gatelevel representation of the circuit except of the D-flipflops.

Answer
State diagram:



State table:

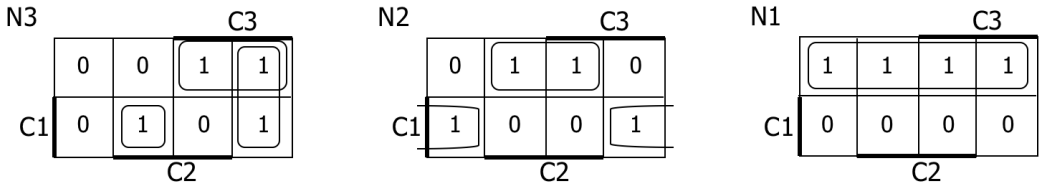
current state		next state	
0	000	001	1
1	001	010	2
2	010	011	3
3	011	100	4
4	100	101	5
5	101	110	6
6	110	111	7
7	111	000	0

A D-flip-flop is used for each state bit of the counter.
The combinational logic is based on the encoding of the counter which is binary.

C3	C2	C1	N3	N2	N1
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

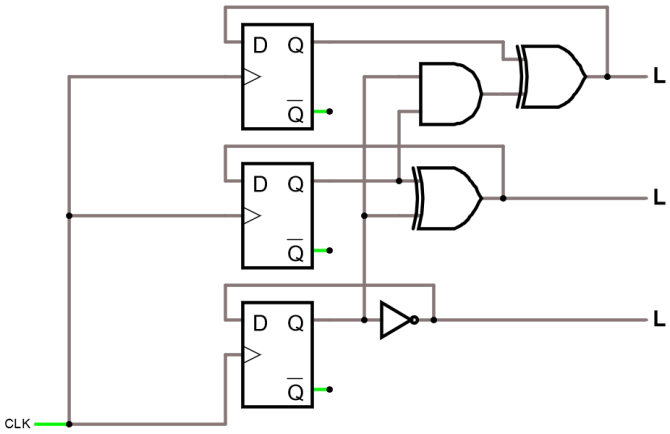
notation to show function represent input to D-FF

N1 := C1'
N2 := C1C2' + C1'C2
:= C1 xor C2
N3 := C1C2C3' + C1'C3 + C2'C3
:= C1C2C3' + (C1' + C2')C3
:= (C1C2) xor C3



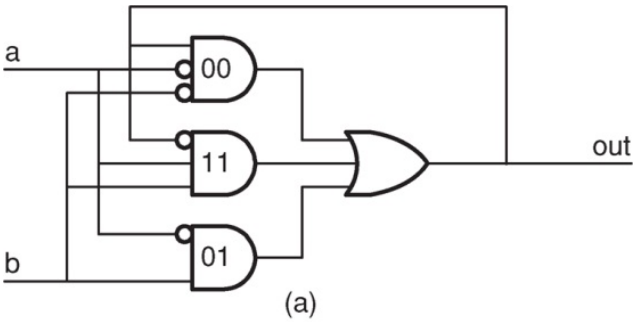
The output of the 3 D-flipflops is the output of the counter. The equations N1, N2, and N3 are the inputs of the Dflipflops.

The gatelevel design is showed below:

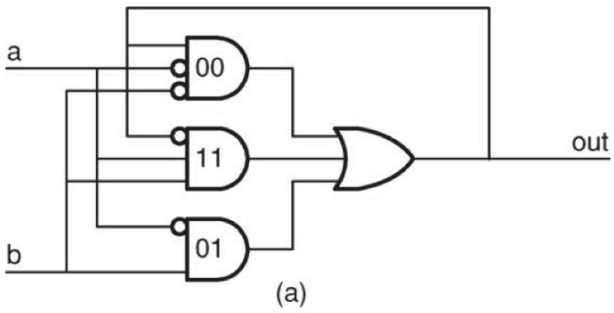


A running simulation can be viewed at this link.

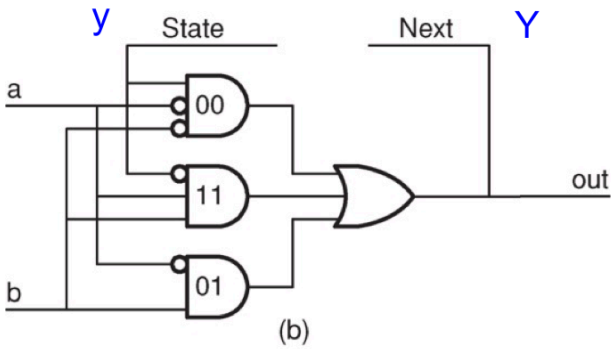
Question 5 Asynchronous: (10 points)
Analyze the following circuit.
Detect the stable and unstable states.
Write the logic equations of the output, next state and current state.



Answer



(a)



(b)

flow table

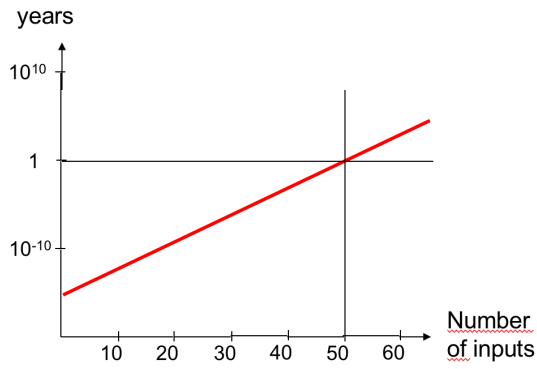
State	Next			
	00	01	11	10
0	(0)	1	1	(0)
1	(1)	(1)	0	0

(c)

When next state is the same as the current state it is a stable state.
Otherwise it is an unstable or transient state.

Question 6 Testing: (10 points)
Explain why exhaustive testing is not practical for a digital circuit that is larger than a few gates. What other alternatives are there? Explain in detail two of them.

Answer
Exhaustive testing is not practical because for a circuit of n inputs 2^n test vectors need to be applied, so the number of tests and the time needed for testing them grows exponentially to the number of inputs, e.g.:



(Test speed 500MHz)

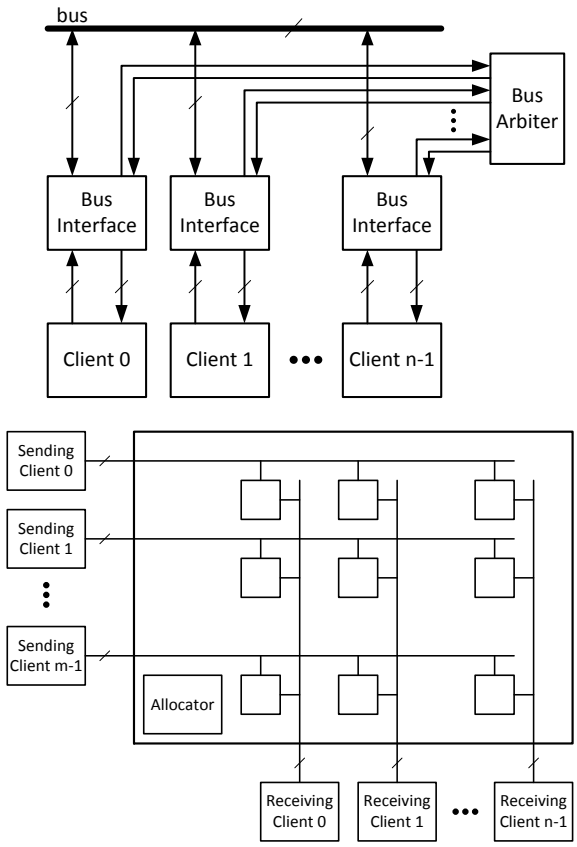
Alternative solutions use a subset of inputs for testing. The inputs subset to be use for testing can be selected in one of the following:

- Select random test vectors
- Based on the logical function of the circuit (functional testing). E.g. check boundary conditions which make your circuit operate differently
- Based on the design structure (structural testing) E.g. partition your hardware in smaller blocks (exploit the hierarchy of your hardware design)
- Based on the fault model. Group a huge number of things that can go wrong in a much smaller finite set of fault models.
    - One well known fault model is the "stuck-at" model. It abstracts all potential faults as resulting in a logical node of the circuit being either stuck at logic "0" or stuck at logic "1".

Question 7 Interconnects: (10 points)
Compare a bus and a crossbar that interconnect 4 clients. Considering a message takes one cycle to be sent through a bus or a network count the number of cycles needed to send the following messages:

- Msg1: Client1->Client2
- Msg2: Client2->Client3
- Msg3: Client3->Client4
- Msg4: Client4->Client1
- Msg5: Client1->Client2
- Msg6: Client3->Client2
- Msg7: Client4->Client2
- Msg8: Client2->Client1

Answer

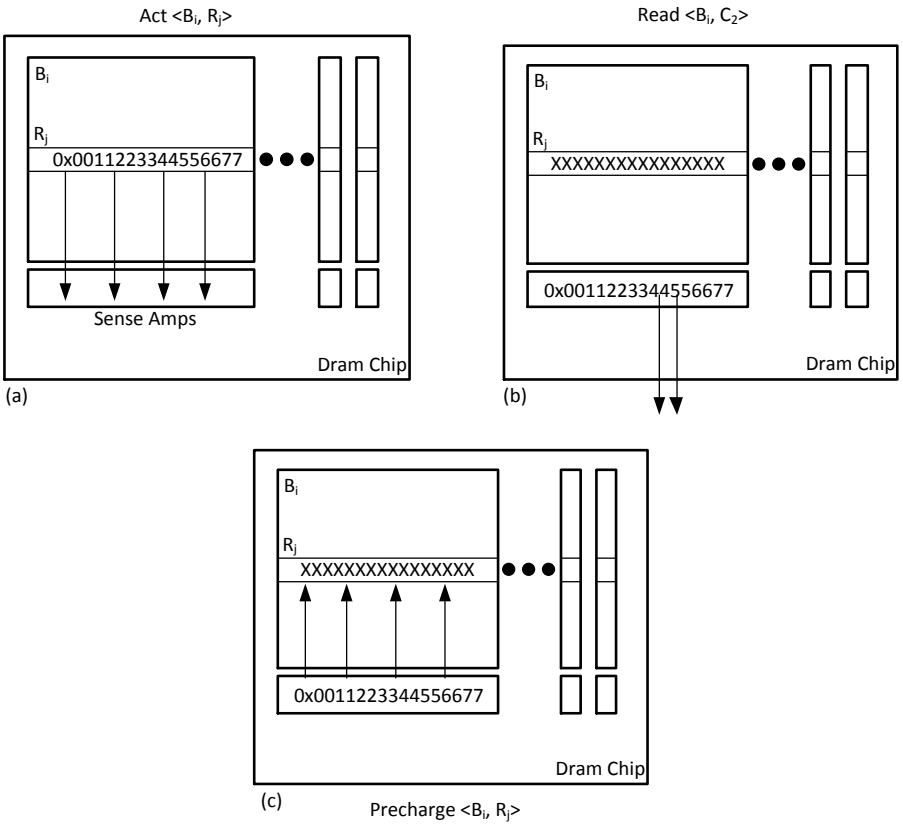


The bus requires 8 cycles (one per message)

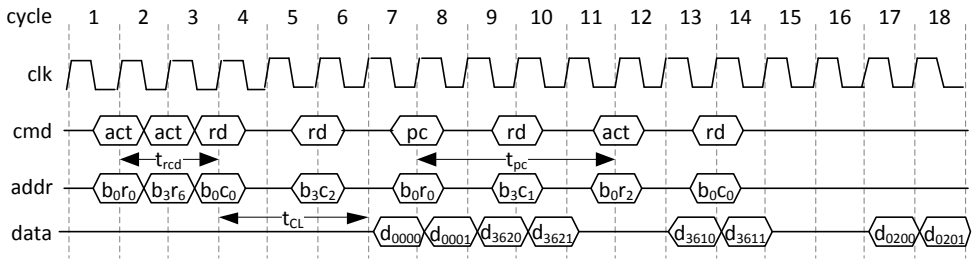
The crossbar will need one cycle for the first 4 messages as each sending client sends a message to a different receiving client. Then messages 5-7 need 3 cycles (one each) as clients1, 3, 4 send a message to the same client (client 2). In parallel to client 2 is free to send a message to client 1. So, in total 1+3=4 cycles are needed for the crossbar.

Question 8 Memory: (10 points)
Describe the basic DRAM operation and show the timing of reading a DRAM word.

Answer



For reading a DRAM word that is in Bank B, Row R and Column C, first the bank B and row R is activated, after activation (which takes t_{rcd} time) a read command is send for the particular column C of the bank B. Then, the row R of bank B is precharged, writing back the row form the row buffer to the DRAM cells.

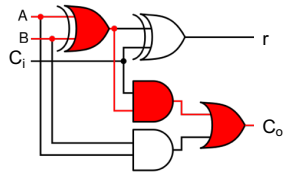


Question 9 Arithmetic & Timing: (10 points)

- Design the fastest possible 16-bit carry-select adder considering that a 2-input XOR gate has a latency of 2 ns and a 2-input OR or AND gate has a latency of 1 ns. Show the critical path.
- Compare the latency of this design with a 16-bit ripple carry adder.
- Compare the area cost of this design with a 16 bit ripple carry adder considering that a XOR gate is double than an OR or an AND gate.

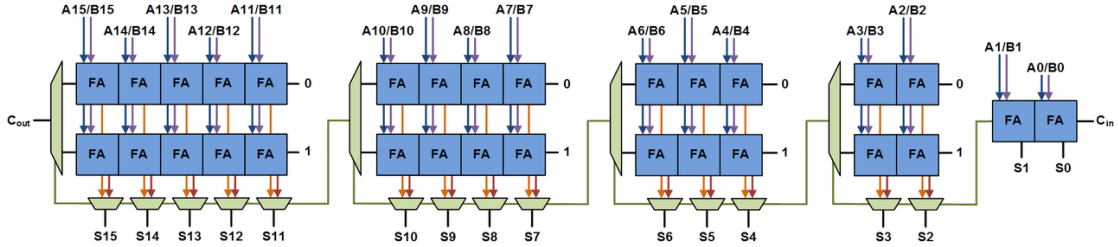
Answer

A full adder:



The fastest 16bit carry select adder has blocks of 2, 2, 3, 4, and 5-bit ripple carry adders.

A ripple carry adder has a critical path that starts from the XOR gate and the OR and AND gate of the first FA, and then goes through the AND and OR gates of the rest of the full adders. So, the latency of a n-bit ripple carry adder is the latency of a XOR + n*(AND+OR)



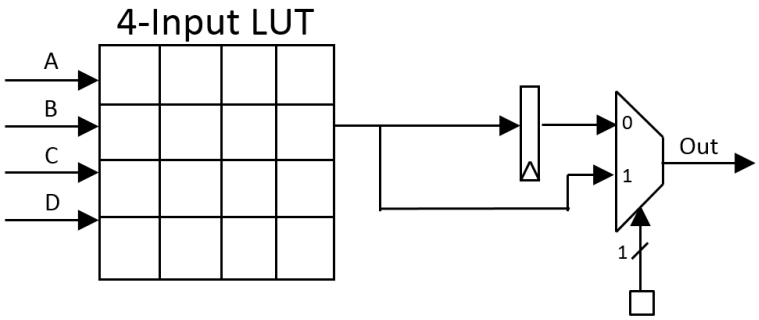
The 16 bit ripple carry adder has a latency of 2+16*(1+1) = 34ns
The above 16-bit carry select adder has a latency of 14ns

If the area cost of an AND gate is n then the area cost of the 16 bit ripple carry adder is 7*16n = 112n
The cost of the above 16-bit carry select adder is 30*7+18*3 = 210+54= 264n
So, the carry select adder is 2,36 times larger than the ripple carry adder.

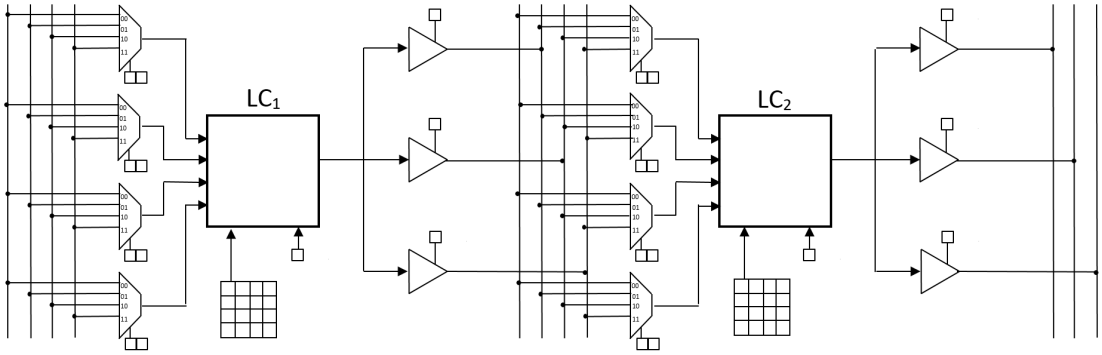
Question 10 Reconfigurable Hardware: (10 points)
Draw the block diagram of a 4-input FPGA logic cell showing all of its configuration bits. Make an array of 2 of the above logic cells (2x1). Show how a 5-input XOR gate is implemented in this array and generate the values of the configuration bits needed.

Answer

The block diagram of a logic cell is shown in the following figure:

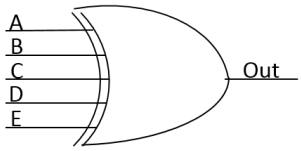


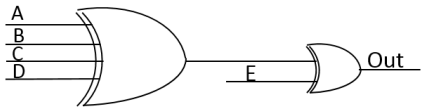
A 4-input LUT, flipflop and a 2-to-1 multiplexer that selects either the registered output of the LUT or the unregistered one. The configuration bits are the 16 bits composing the LUT and a bit driving the select of the multiplexer, in total 17 bits.



Putting together 2 logic cells to form a 2x1 FPGA array is shown in the figure above. Each input of each logic cell selects one of multiple wires (4 in this drawing) using a "connection box". The logic cell output selects one (out of 3 wires) to drive through tri-states. Since the array is composed of only 2 logic cells we don't need switches.

A 5 input XOR gate would be partitioned in a 4 input XOR followed by a 2 input XOR in order to fit in the 2 logic cells as shown in the following figure.





Then, the truth tables for the 4-input and 2-input XOR gates are as follows:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

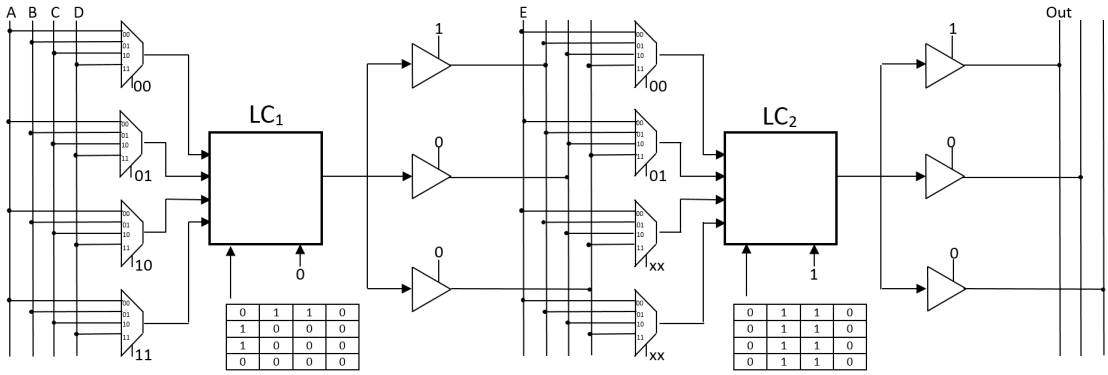
Truth table of 4-input XOR gate (LC1)

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Truth table of 2-input XOR gate (LC2)

Note that since the LUT is 4-input and the second XOR is 2 input, we use only two of the inputs (the two leftmost ones) and the others are don't care. So the truthtable is identical every 4 lines (output values are a repeated sequence of 0, 1, 1, 0 values).

The figure bellow show the above configuration of the 2 LUTs in the logic cells and the configurations needed for the connection boxes to connect wires with the logic cells (through the multiplexers and tristates). Finally note that the LC_2 configuration bit of the 2-to-1 mux, selects the registered output of the LUT because the 5-input XOR gate is registered.



0	1	1	0
1	0	0	0
1	0	0	0
0	0	0	0

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

END of EXAM