EDA322/ DIT797: Digital Design
Exam - March 2018

Date: March 14, 2018

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: room 4128 EDIT on **April 17th 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!

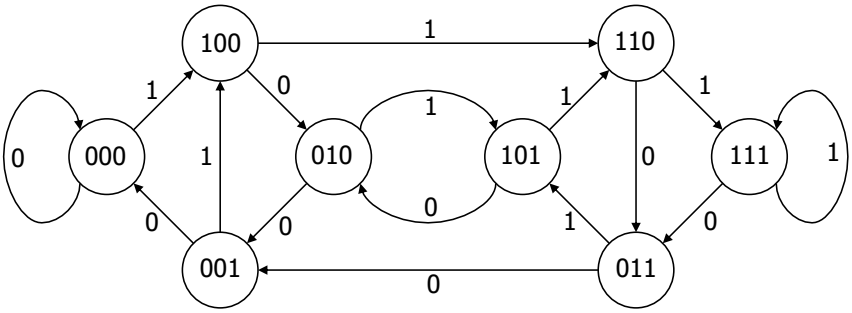
Question 1 Sequential Circuits: (10 points)
Draw the block diagram of a D-latch and explain its functionality for every combination of input values.

Answer
Lecture on Sequential circuits, slide 13.

Question 2 Interfaces: (10 points)
What is the difference of a simple flow control interface with a serialized flow control interface? Show the difference in a timing diagram.

Answer
Lecture on Interfaces, slide 36 and 39
Serialization considers that every time valid data are sent multiple (fixed number of) words are sent to the receiver, taking equal amount of cycles to be sent.

Question 3 FSMs: (10 points)
Derive and draw the sequential circuit that implements the FSM described by the following stage diagram:

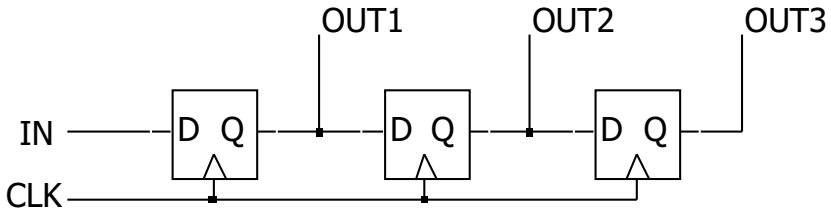


Consider the following:
- Starting state is 000
- The output of the FSM is equal to the state bits.

Answer
The above state diagram describes a 3 bit shift register.

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

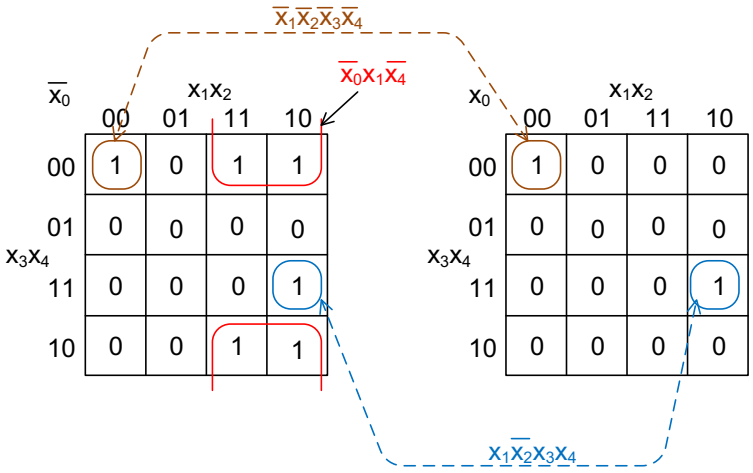
N1 := In
N2 := C1
N3 := C2



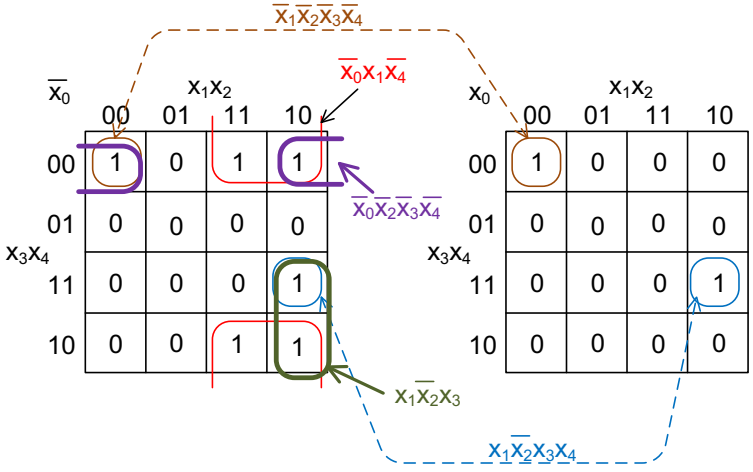
Question 4 Asynchronous: (10 points)

Minimize function $f(x_0, x_1, x_2, x_3, x_4)=\Sigma(0, 8, 10, 11, 12, 14, 16, 27)$, detect hazards and remove them.

Answer



$f(x_0, x_1, x_2, x_3, x_4)=x_1'x_2'x_3'x_4' + x_0'x_1x_4' + x_1x_2'x_3x_4$

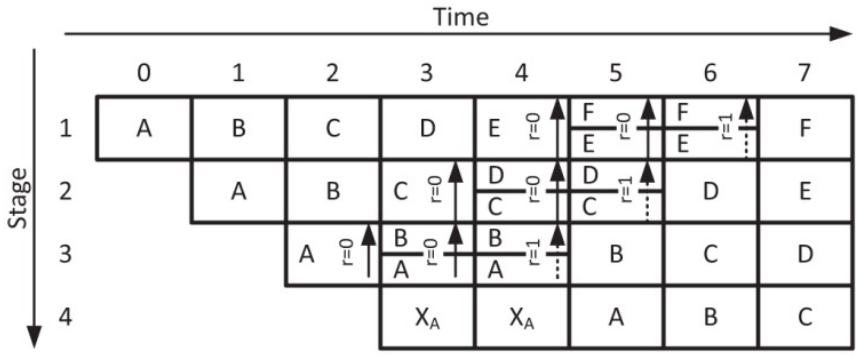


$f(x_0, x_1, x_2, x_3, x_4)=x_1'x_2'x_3'x_4' + x_0'x_1x_4' + x_1x_2'x_3x_4 + x_0'x_2'x_3'x_4' + x_1x_2'x_3$

Question 5 Pipelining: (10 points)

a) Describe the block diagram of a pipeline stage that uses double buffering as a stalling mechanism.

b) Describe what are the contents of the double buffers (register A and register B) in stage 2 of a pipeline with 4 stages that has the following timing. Describe the contents of the two registers during the cycles 0-7.



Note: *r* is the ready signal of a stage

Answer:

a) Lecture on pipelining, slide 34
b) Cycle 0: reg A = X, reg B = X
   Cycle 1: reg A = A, reg B = X
   Cycle 2: reg A = B, reg B = X
   Cycle 3: reg A = C, reg B = X
   Cycle 4: reg A = C, reg B = D
   Cycle 5: reg A = C, reg B = D
   Cycle 6: reg A = D, reg B = D
   Cycle 7: reg A = D, reg B = E

Question 6 Testing: (10 points)
Draw the gate-level block diagram of a (1-bit) Full adder with registered inputs and outputs. Add to the full adder a scan chain and explain its modes of operation.

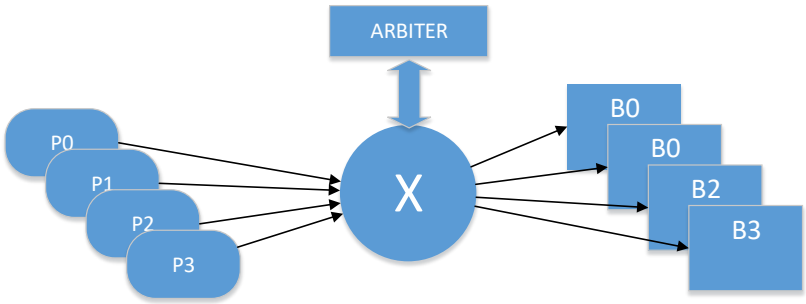
Answer
The full adder is explained in Lecture 2 (slide 43).
The way to apply to it a scan chain is explained in Lecture on Testing and Design for Testing.

Question 7 Timing: (10 points)

a) What are the timing parameters of a flip-flop?
b) What are the constraints for the above timing parameters so that two cascaded flip-flops can form a correctly functioning 2-bit shift register?

Answer
Lecture on Timing, slides 19-20.

Question 8 Memories & Interconnects: (10 points)
Consider 4 processors (P0, P1, P2, P3) connected through a crossbar switch to the 4 banks (B0, B1, B2, B3) of a memory system as shown in the figure. Each processor sends requests to access (read or write) a bank of the memory. Each bank can serve one request at a time.



a) What is the maximum number of requests that the crossbar switch allows in parallel to the memory?
b) If the processors and memory banks were interconnected with a bus (instead of a crossbar switch) what would be the maximum number of parallel requests to the memory then?

Let us consider again the example of the figure where the processors are interconnected with the banks through a crossbar switch. The crossbar switch takes 1 cycle to send a request from a processor to a particular memory bank. Each processor sends 4 requests (one request to each bank) as follows:

P0 requests to access B0, B1, B2, and B3.
P1 requests to access B0, B1, B2, and B3.
P2 requests to access B0, B1, B2, and B3.
P3 requests to access B0, B1, B2, and B3.

c) Consider that the system cannot deal with the head of line blocking problem and that a bank access takes one cycle. Describe the head of line blocking problem and show how many cycles are needed to serve all 16 above requests (4 bank access requests coming from each one of the 4 processors) if they appear in the above order.
d) What can be done to avoid the head of line blocking? How many cycles would be needed then to serve the 16 requests?

Answer
Lecture on Interconnects and Memory
a) Up to 4
b) Up to 1
c) The arbiter sees only the first request from each processor (the head of the line). It will take 7 cycles since the arbiter will see in the first cycle that all processors request access to B0, the second cycle P0 will request access to B1 and the rest of the processors to B0, etc. as shown in the first table of slide 25 of the lecture

d) The head of line blocking can be solved if the arbiter is able to see all requests from each processor and then choose to serve first requests to different banks from each processor. It would then take 4 cycles to serve all 16 requests as shown in the second table of slide 25 of the lecture.

Question 9 Arithmetic: (10 points)

a) Draw the block diagram of a 4-bit carry lookahead adder with inputs: carry-in, a[0-3], b[0-3], and outputs sum[0-3] and carry-out and write the Boolean functions that describe it.
b) Perform a timing analysis of the circuit and find the critical path considering that 2-input AND and OR gates have a delay of 1nsec, and a 2-input XOR gate has a delay of 2nsec (NOT gates can be considered having zero delay).
c) Considering the above gate delays, what is the critical path delay of a 4-bit ripple carry adder? It is longer is shorter than the above carry lookahead adder?

Note: there is no need to minimize the Boolean function you will derive. Just measure the latencies of each equation and find the critical path.

Answer

$p_0 = a_0 \oplus b_0$
$g_0 = a_0\, b_0$
$p_1 = a_1 \oplus b_1$
$g_1 = a_1\, b_1$
$p_2 = a_2 \oplus b_2$
$g_2 = a_2\, b_2$
$p_3 = a_3 \oplus b_3$
$g_3 = a_3\, b_3$

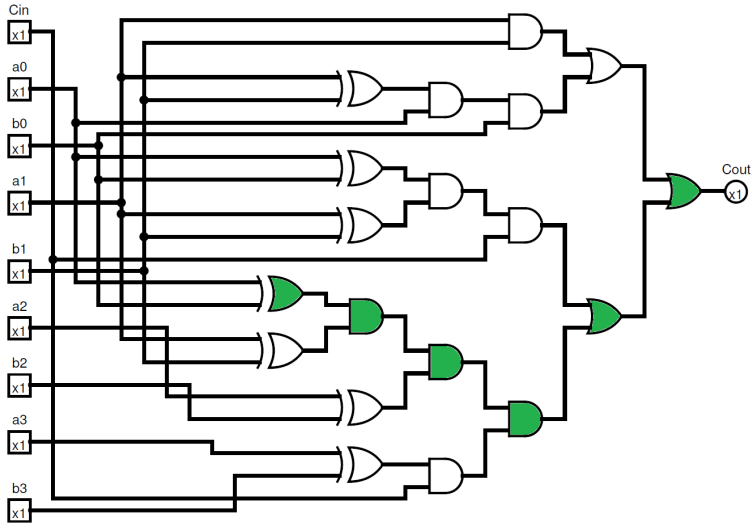
$P_{0-1} = p_0 p_1 = (a_0 \oplus b_0)(a_1 \oplus b_1)$
$G_{0-1} = g_1 + p_1 g_0 = a_1\, b_1 + (a_1 \oplus b_1)\, a_0\, b_0$
$P_{2-3} = p_2 p_3 = (a_2 \oplus b_2)(a_3 \oplus b_3)$
$G_{2-3} = g_3 + p_3 g_2 = a_3\, b_3 + (a_3 \oplus b_3)\, a_2\, b_2$

$P_{0-3} = P_{0-1} P_{2-3} = (a_0 \oplus b_0)(a_1 \oplus b_1)\,(a_2 \oplus b_2)(a_3 \oplus b_3)$
$G_{0-3} = G_{2-3} + P_{2-3} G_{0-1} = a_3\, b_3 + (a_3 \oplus b_3)\, a_2\, b_2 + (a_2 \oplus b_2)(a_3 \oplus b_3)[a_1\, b_1 + (a_1 \oplus b_1)\, a_0\, b_0]$

$c_1 = g_0 + p_0 c_{in} = a_0\, b_0 + (a_0 \oplus b_0)\, c_{in}$
$c_2 = G_{0-1} + P_{0-1} c_{in} = a_1\, b_1 + (a_1 \oplus b_1)\, a_0\, b_0 + (a_0 \oplus b_0)(a_1 \oplus b_1)\, c_{in}$
$c_3 = g_2 + p_2 c_2 = a_2\, b_2 + (a_2 \oplus b_2)[\, a_1\, b_1 + (a_1 \oplus b_1)\, a_0\, b_0 + (a_0 \oplus b_0)(a_1 \oplus b_1)\, c_{in}]$

$$\mathbf{c_{out} = G_{0-3} + P_{0-3} c_{in} = a_1\, b_1 + (a_1 \oplus b_1)\, a_0\, b_0 + (a_0 \oplus b_0)(a_1 \oplus b_1)\, c_{in} + (a_0 \oplus b_0)(a_1 \oplus b_1)(a_2 \oplus b_2)(a_3 \oplus b_3)\, c_{in}}$$

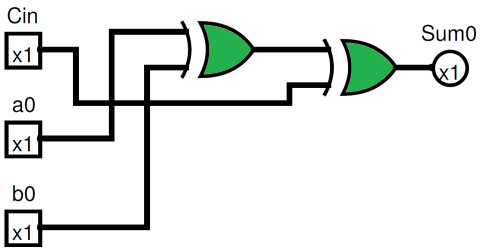
c_{out} delay 7 ns (can also be less than 7 ns by arranging differently the gates)



Cout logic circuit using 2-input gates

$Sum_0 = a_0 \oplus b_0 \oplus c_{in}$

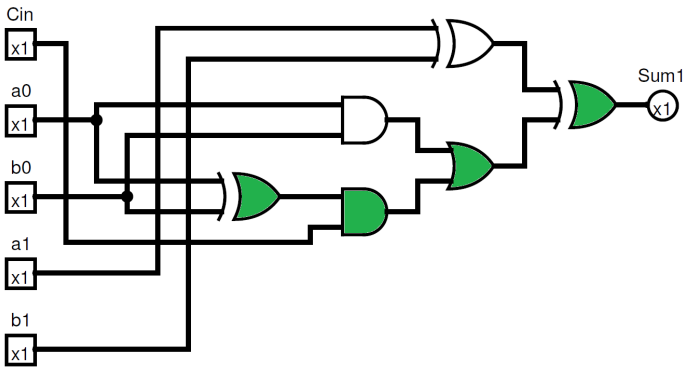
Sum_0 delay 4ns



Sum $_0$ logic circuit using 2-input gates

$Sum_1 = a_1 \oplus b_1 \oplus c_1 = a_1 \oplus b_1 \oplus [\, a_0\, b_0 + (a_0 \oplus b_0)\, c_{in}\,]$

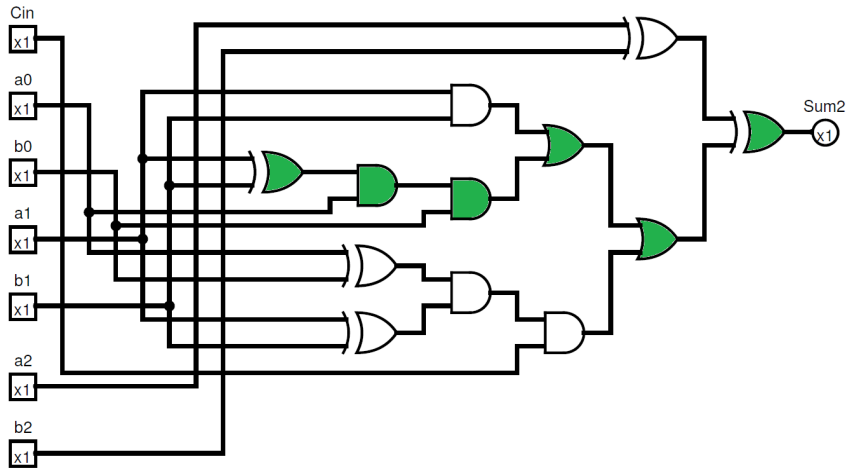
Sum_1 delay 6ns



Sum $_1$ logic circuit using 2-input gates

$Sum_2 = a_2 \oplus b_2 \oplus c_2 = a_2 \oplus b_2 \oplus [\, a_1 b_1 + (a_1 \oplus b_1)\, a_0 b_0 + (a_0 \oplus b_0)(a_1 \oplus b_1)\, c_{in}]$

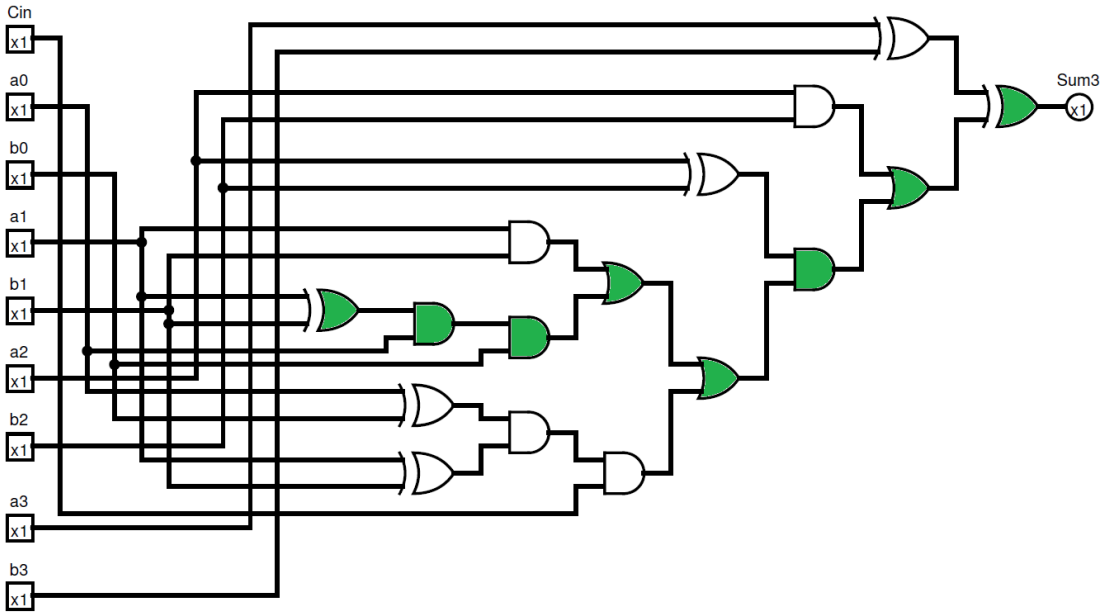
Sum_2 delay 8ns (can also be less than 8 ns by arranging differently the gates)



Sum $_2$ logic circuit using 2-input gates

$Sum_3 = a_3 \oplus b_3 \oplus c_3 = a_3 \oplus b_3 \oplus [\, a_2 b_2 + (a_2 \oplus b_2)[\, a_1 b_1 + (a_1 \oplus b_1)\, a_0 b_0 + (a_0 \oplus b_0)(a_1 \oplus b_1)\, c_{in}]\,]$

Sum_3 delay 10 ns (can also be less than 10 ns by arranging differently the gates)



The critical path is Sum_3 with 10 ns of delay

The delay of a 4-bit ripple carry adder would be also 10 ns, so it would be as fast as the above carry lookahead adder.

Question 10 Reconfigurable Hardware: (10 points)
What is the difference between reconfigurable and (re-)programmable?
Is an ALU reconfigurable or (re-)programmable and why?

Answer
Lecture on Reconfigurable hardware, slide 58.

END of EXAM