

EDA322/ DIT797: Digital Design Exam - Aug 2020

Date: August 27, 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!

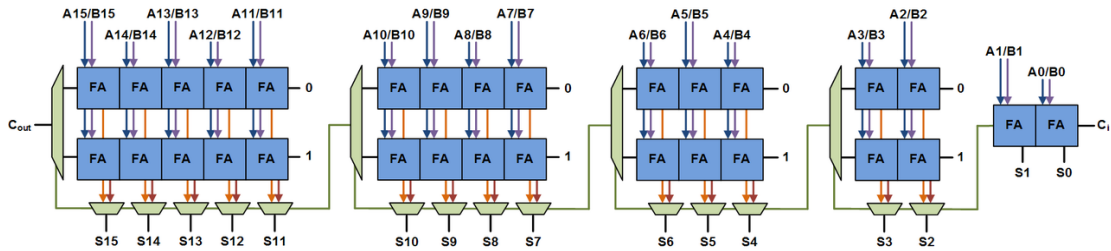
Good luck!

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

Question 1 Arithmetic & Timing: (10 points)

Consider a carry select adder of 8/16-bits. Who many blocks would you choose and what is the size of its block (number of added bits) in order to minimize the delay of the adder? Show the critical path considering the delay of a Full adder is X ns and the delay of a 2-to-1 multiplexer is Y ns.

Answer



For a 16bit CS adder and for FA and 2:1 mux delays being equal ($X=Y$) the solution is 5 blocks of sizes: 5, 4, 3, 2, and 2 bits.

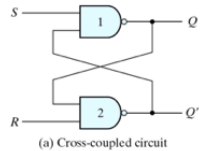
Question 2 Asynchronous circuits: (10 points)

Analyze an $S'R'$ latch (it uses NAND gates instead of the NOR used by the SR-latch). Identify and explain a critical race in the circuit.

Answer

Analysis of an S'R' latch (1)

- We can analyze an SR latch using the previous technique



(a) Cross-coupled circuit

S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

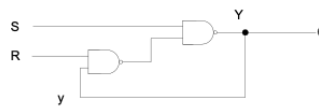
(b) Truth table

- Equations derived for secondary variable (same equation for output):

$$Y = \overline{S(Ry)} = \overline{S(R + \bar{y})} = \bar{S} + Ry$$

- Since we want to avoid the SR=00 situation, we can write:

$$Y = \bar{S} + Ry \text{ if } S'R' = 0$$



EDA322 Digital design, 2019-2020, Lecture 18

I. Sourdís, CSE, Chalmers

103

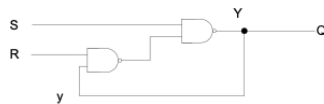
103



Analysis of an S'R' latch (2)

- Can derive the transition table and the flow table:

curr state	next state				output	curr state	next state				output
	SR=00	01	11	10			SR=00	01	11	10	
y	Y	Y	Y	Y		y	Y	Y	Y	Y	
0	1	1	0	0	0	a	b	b	a	a	0
1	1	1	1	0	1	b	b	b	a	a	1



EDA322 Digital design, 2019-2020, Lecture 18

I. Sourdís, CSE, Chalmers

104

104



Analysis of an S'R' latch (3)

- Note: We can see the undesirable case when SR=00 and inputs change.
- Depending on the various delays and assuming SR=00 => SR=11...
 - If SR=00 -> SR=10 -> SR=11, we get stable state with output of 0.
 - If SR=11 -> SR=01 -> SR=11, we get stable state with output of 1.
- So the stable state is unpredictable.

EDA322 Digital design, 2019-2020, Lecture 18

I. Sourdís, CSE, Chalmers

105

105

Question 3 Number representation: (10 points)

Suppose you need to represent time from 1 second (sec) to 10^4 seconds 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 $5 \cdot 10^{-2}$ sec.

This needs a resolution of $2 \cdot 5 \cdot 10^{-2} = 1/10$ sec

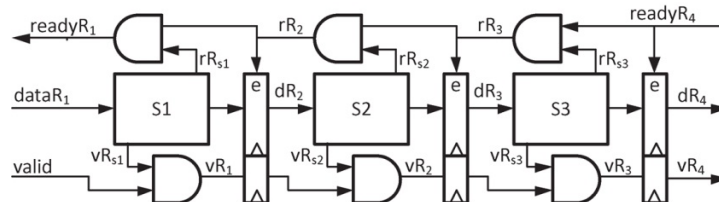
If seconds is the integer part we need, then we need 14 bits for integer (to count between 1-104 sec, $2^{14} = 16k$). Then, the fraction needs 4 bits to count $1/24 = 1/16$ sec (which is a bit smaller than $1/10$ sec resolution). So the fixed-point representation would require 18 bits (14.4).

Alternatively, we can count $1/10$ seconds which is the required resolution. 104 seconds = 105 tenths of seconds for which we need 17 bits ($2^{17} = 128 \cdot 1024 = 131072$).

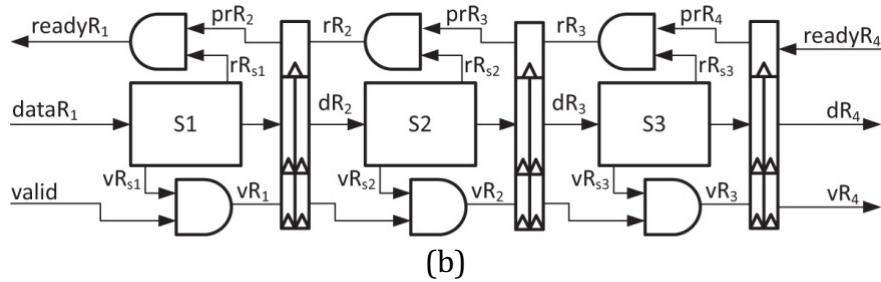
- b) For a floating-point representation, the mantissa needs to be only 4 bits to offer accuracy of 5%. So, the resolution needs be double that: $10\% = 1/10$; $1/16 = 1/24$ should then be enough (4 bits mantissa). The dynamic range is from 1 seconds to 104 seconds so it is 104, which is smaller than $16k = 2^{14}$. Then, 4 bits are enough to count up to 14, so with 4-bits exponent we can represent a range of 214. We finally set the bias to 1 because we need the exponent to go up to $2^{15} - 1 = 24$. So, the floating point format is 4 bits mantissa and 4 bits exponent, 8 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 & Timing: (10 points)

- i. Explain the differences between the two following methods for stalling a pipeline:



(a)



How do they differ in terms of circuit delay and area?

ii. Which method of the two would you choose if:

- the delay of a 2-input AND gate is 1ns, and
- the delay of each pipeline stage (S1, S2, S3) is 2/3/4 ns

X: and your primary objective is to maximize the operating frequency of the pipeline?

Y: and your primary objective is to minimize the latency of the pipeline?

Consider the setup and propagation times of the flipflops zero.

Answer

i.

(a) has longer delay which doesn't scale well (its linear) to the number of stages.

(b) scales better to the number of stages as it is pipelined so it has the same delay (per stage) for any number of stages in the pipeline. However, it adds registers to (1) pipeline the feedback signal and (2) for double buffering.

ii. The sum of the delays of the 3 AND gates is 3ns.

If the stage delay is 3ns or larger then choice (a) is better because it does not increase the clock cycle of the design and has lower area costs than (b)

If the stage delay is less than 3ns then choice (b) is faster as otherwise with choice (a) the clock period has to be 3ns.

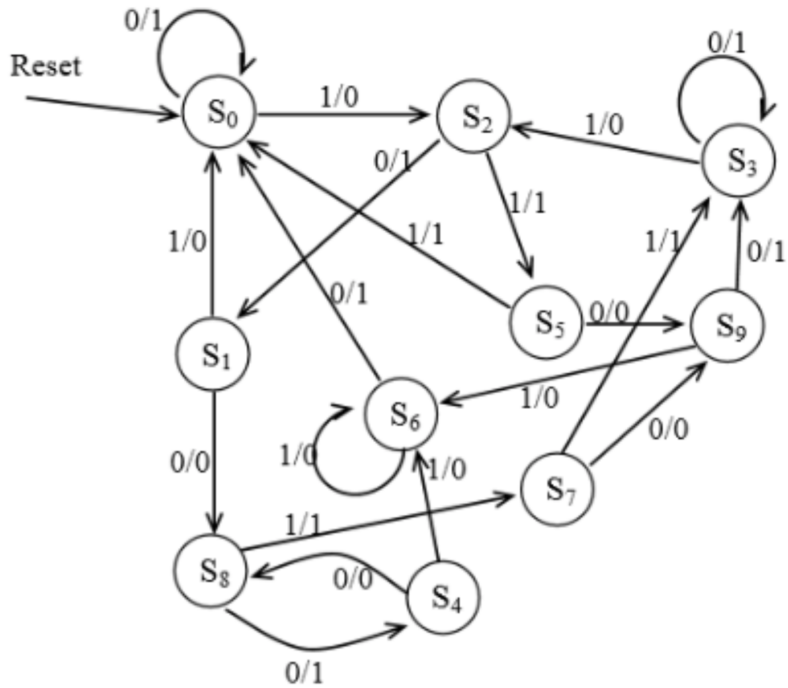
Question 5 FSMs: (10 points)

The state diagram defines an FSM with an Input X and an output Z.

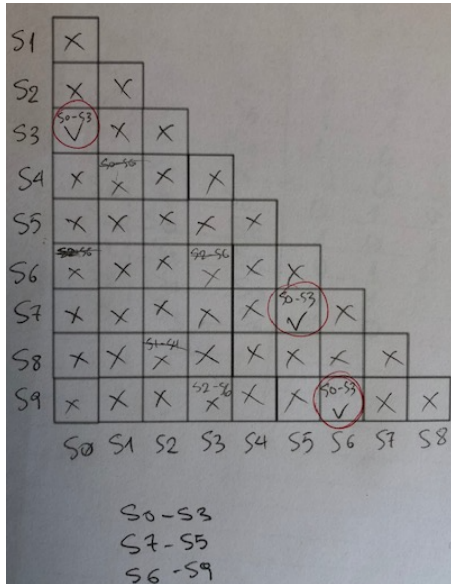
a) Minimize the number of states of this FSM and list the equivalent states.

b) Draw the state diagram and state table of the reduced FSM.

c) derive the Boolean functions for the next state bits and the output of the reduced FSM.



Answer



Pres.	Next State		Output	
	x=0	x=1	x=0	x=1
S0	S0	S2	1	0
S1	S8	S0	0	0
S2	S1	S5	1	1
S3	S3	S2	1	0
S4	S8	S6	0	0
S5	S9	S0	0	1
S6	S0	S6	1	0
S7	S9	S3	0	1
S8	S4	S7	1	1
S9	S3	S6	1	0

Pres State	Next State		Output	
	X=0	X=1	X=0	X=1
S ₀	S ₀	S ₂	1	0
S ₁	S ₈	S ₀	0	0
S ₂	S ₁	S ₅	1	1
S ₄	S ₈	S ₆	0	0
S ₅	S ₆	S ₀	0	1
S ₆	S ₀	S ₆	1	0
S ₈	S ₄	S ₅	1	1

Pres State	Next State		Output(z)	
	X=0	X=1	X=0	X=1
S ₀	000	010	1	0
S ₁	001	000	0	0
S ₂	010	001	1	1
S ₄	011	110	0	0
S ₅	100	101	0	1
S ₆	101	000	1	0
S ₈	110	011	1	1
	111	d	d	d

x y ₂	y ₁ y ₀			
	00	01	11	10
00				1
01	1		d	1
11		1	d	
10			1	

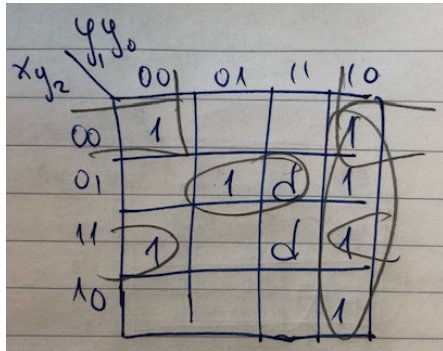
$$Y_0 = x'y_1y_0' + x'y_2y_0' + xy_2y_0 + xy_1y_0$$

x y ₂	y ₁ y ₀			
	00	01	11	10
00		1	1	
01			d	1
11			d	
10	1			

$$Y_1 = xy_2'y_1'y_0' + x'y_2'y_0 + x'y_2y_1$$

x y ₂	y ₁ y ₀			
	00	01	11	10
00		1	1	
01			d	
11	1	1	d	1
10			1	1

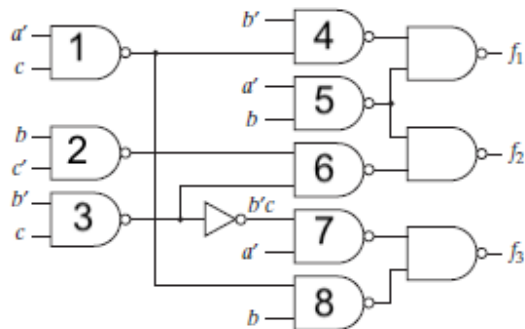
$$Y_2 = x'y_2'y_0 + y_1y_0 + xy_2 + xy_1$$



$$Z = y_1 y_0' + x' y_2' y_0' + x' y_2 y_0 + x y_2 y_0'$$

Question 6 Testing: (10 points)

Activate, propagate and justify the stuck-at-[x] fault at the output of gate [y].



Answer

Similar to Lecture on Testing, slides 50-54

Question7 Memory: (10 points)

Use memory blocks that have 2^x entries and each entry has [y] Bytes, to construct a memory of which has 2^{x+z} entries and [j]*[y] Bytes.

Draw the block diagram of the memory and show how the address bits are connected.

Answer

Similar to Lecture on memory, slide 23

Question8 Arithmetic: (10 points) Francisco

Show the contents of the registers in a serial 4-bit multiplier/divider that performed it performs the operation [x]/[y]

Answer

Lecture on Arithmetic slide 11 for multiplication and slide 25 for divider.

Question 9 Reconfigurable Technologies & Timing: (10 points)

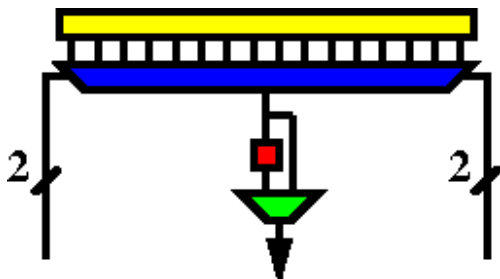
Consider the following 4-input logic cell with optional registered output. The cells of its LUT (yellow box) as well as the register of the output are composed of D-flipflops.

Given that:

- the delay of a 2-to-1 AND or OR logic gate is 1x ns and that
- the setup time of the flip flop as well as its propagation time are 1 ns each

find:

1. the delay of the critical path of the logic cell.
2. How does it compare to the slowest non-programmable (ASIC) gate with the same number of inputs?

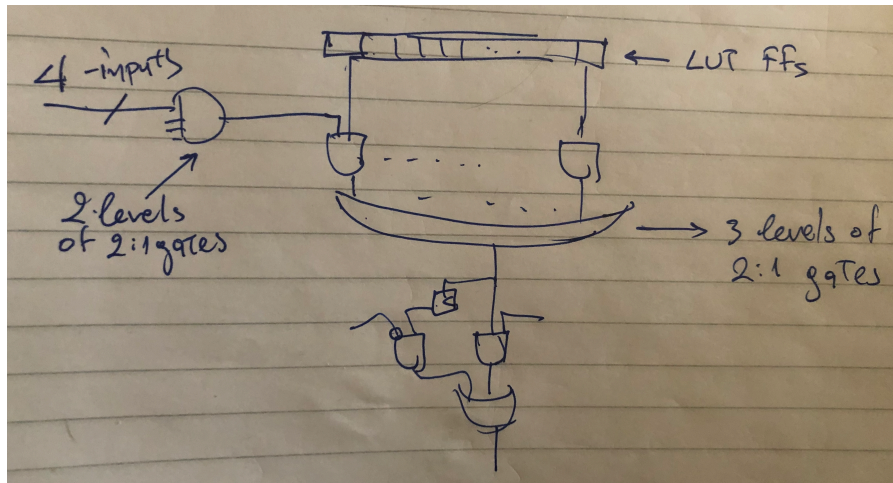


Answer

The critical path of the logic cell starts from its 4 inputs which are used as a select of the 16:1 multiplexer, the select needs to be decoded in 4-input AND gate, go through a 2-input AND gate and a 16 input OR gate. That is a delay of seven 2:1 gates.

Then, the delay of the 2:1 mux is added (two 2:1 gates). So in total the critical path has a delay of 9 2:1gates which is 9 ns.

For a 4 input AND or OR gate one would need two levels of 2:1 AND/OR gates (2 ns). However, in order to calculate the worst case delay of any 4:1 boolean function, expressed in sum of products then one needs to think of the worst case k-map that can be created. That one would have a delay of a 4-input AND gate followed by an 8 input OR gate. This needs 5 levels of 2:1 AND/OR gates and has a delay of 5 ns.



Question 10 Delay & Power: (10 points)

Compare the dynamic power of an 8-bit (i) ripple carry adder, (ii) carry lookahead adder, and (iii) a carry select adder 2 blocks of 4-bits each).

Consider that the all of them compute a new result as fast as they can.

Answer:

Based on the equation:

$$P_{avg} = n \cdot \alpha_{avg} \cdot f \cdot c_{avg} V_{dd}^2$$

Where:

n is the number of gates and is proportional to the area, $[\alpha \cdot f]$ indicates how often the gates switch, which is based on the delay of the adders (as there is no clock).

So, the number of gates is proportional to the area. So the larger the area of the adder the larger the number of gates (n) and the larger the P .

α is reversely proportional to the delay. The smaller the delay the larger the activity factor and the larger the P .

So you need to find how the adders compare with respect to area and delay:

If b is the number of bits added in the adders, then:

Ripple carry adder Area is:

$$A_{RC} = O(b)$$

Carry select adder is:

$$A_{CS} \sim 1.5 \cdot A_{RC}$$

Carry lookahead adder is:

$$A_{CL} = O(b \cdot \log b) \text{ so } A_{CL} \sim 3 \cdot A_{RC}$$

Ripple carry adder delay is:

$$D_{RC} = O(b)$$

Carry select adder delay is:

$$D_{CS} \sim (4/8) \cdot D_{RC} + 1/4 D_{RC} \text{ (for delay of a 2:1 mux)} = 0.625 \cdot D_{RC}$$

Carry lookahead adder delay is:

$$D_{CL} = O(\log b) \text{ so } D_{CL} \sim 3/8 \cdot D_{RC}$$

The activity of each adder is 1/delay.

If P_{RC} is the dynamic power of the RC adder then

Looking again at the equation:

$$P_{avg} = n \cdot \alpha_{avg} \cdot f \cdot c_{avg} V_{dd}^2$$

The power of the RC adder is $P_{RC} = [A_{RC} / D_{RC}] \cdot CV^2$

The power of the CS adder is $P_{CS} = [A_{CS} / D_{CS}] \cdot CV^2 = [1.5 / 0.625] \cdot P_{RC} = 2.4 \cdot P_{RC}$

The power of the CL adder is $P_{CL} = [A_{CL} / D_{CL}] \cdot CV^2 = [3 / (3/8)] \cdot P_{RC} = 8 \cdot P_{RC}$

END of EXAM