

EDA322: Digital Design

Exam - March 2017

Date: March 15, 2017

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 10th 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: (10 points)

Considering that the delay of a Full Adder (FA) is 1ns and the delay of a 2-to-1 multiplexer is also 1ns:

- a) Find in a 16-bit carry select adder the number of its blocks and the sizes of these blocks in order for the adder to have a delay of 6 ns. Draw the block diagram of the adder (without showing the internals of each block)
- b) Is this number of blocks and sizes of blocks optimum? Can you reduce the delay of the 16-bit carry select adder more, and why?

Answer:

Lecture on Logic Minimization, Adders, slide 50

b)

If the number of blocks increase to 6 then the multiplexers delay would be 5 ns and the adder delay would be 5ns plus at least one 1 ns of at least 1 FA in the first block would be again 6 ns.

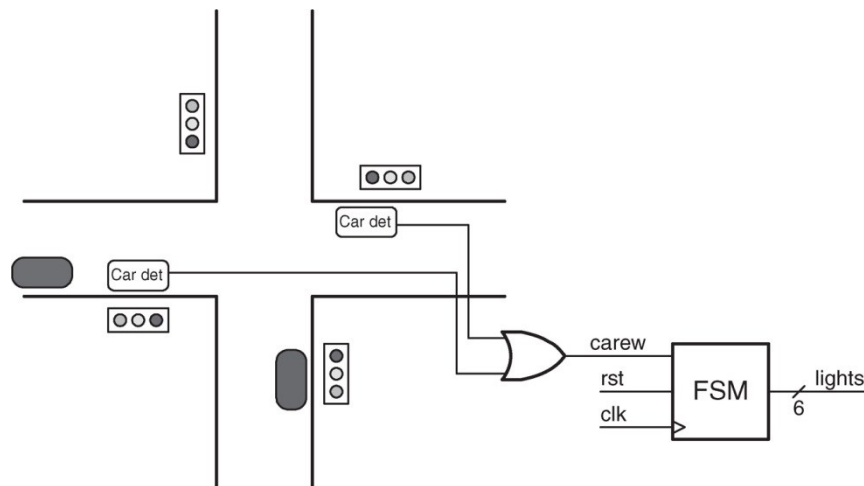
If the number of blocks decrease to 4 then at east 2 remaining blocks would need to have at least one more FA each. Then the delay of the adder would be 7 ns.

(The mentioned lecture slide can be found at the end of this document)

Question 2: (10 points)

Design a synchronous FSM (Finite State Machine) for traffic light control. The traffic light controller allows cars to go either in the direction north-south and south north, or in the direction west-east and east-west. Cars are not allowed to turn left or right. The controller has 3-bits output for each direction: **NS-green, NS-yellow, NS-red, EW-green, EW-yellow, EW-red**. The inputs besides the **clock** and the **reset** (reset=1 allows NS-green to be on) has 1 more input to indicate cars waiting in the EW direction: **carew** (1-bit). For the NS direction such input is not needed because NS is the default direction that is allowed when reset is on.

- a) Draw the state-diagram of the FSM
- b) Fill in the state-table of the FSM
- c) Make a state-assignment and derive the Boolean functions that produce the next state bits and the output.
- d) Draw the hardware design (block diagram) of the FSM.



Note: if there is any information missing in the above description of the FSM, feel free to define it yourselves. In such case, explain it in your answer.

Answer:

Book sections 14.3-14.5 (other solutions are acceptable, too)

Question 3: (10 points)

Draw the block diagram of an FPGA logic cell with 4-bits input and 1-bit output. Show its configuration when it implements the following logic function registered:

$$f = a * b + c * d$$

Answer:

Similar to Lecture on Technology – Reconfigurable Hardware, slide 18

Question 4: (10 points)

- a) Find the closest 10-bit integer to the real number 47,749 and calculate the absolute and relative errors between them.
- b) Find the closest floating-point number with 4-bits exponent and 6-bits mantissa to the same above real number and calculate the absolute and relative error between them.

Answer:

The closest integer is 48: 0000110000

The closest floating point should have an exponent $Exp = 0110$ so that $2^6 = 64$ because mantissa has to be a number between $[0.5, 1)$ ($2^5 = 32$ is not large enough, while $2^7 = 128$ would lower precision).

Considering 6 bits of mantissa then there are two options:

1. consider that the MSbit is implied to be always "1" followed by 6 more bits
2. consider that the 6 bits of mantissa include the MSbit that is always one:

for case 1 we have 7 bits in total then the mantissa is multiple of $1/128$, in other words mantissa $M = x/128$

$47,749 = \text{Exp} * M = 2^6 * x/128 = x/2 \Leftrightarrow x = 47,749$, the closest integer is 95 so mantissa is $M=1011111$

for case 2 we have 6 bits in total then the mantissa is multiple of $1/64$, so $M = x/64$

$47,749 = 2^6 * x/64 = x \Leftrightarrow x=47,749$, the closest integer is 48 so mantissa is $M=110000$

Errors are calculated based on the formulas in the lecture on "Arithmetic Units", slide 32.

Question 5: (10 points)

Describe the flow control interface using a timing diagram to show the data transfer between a sender and a receiver.

Answer:

Lecture on System Design and Interfaces, slides 36, 37

Question 6: (10 points)

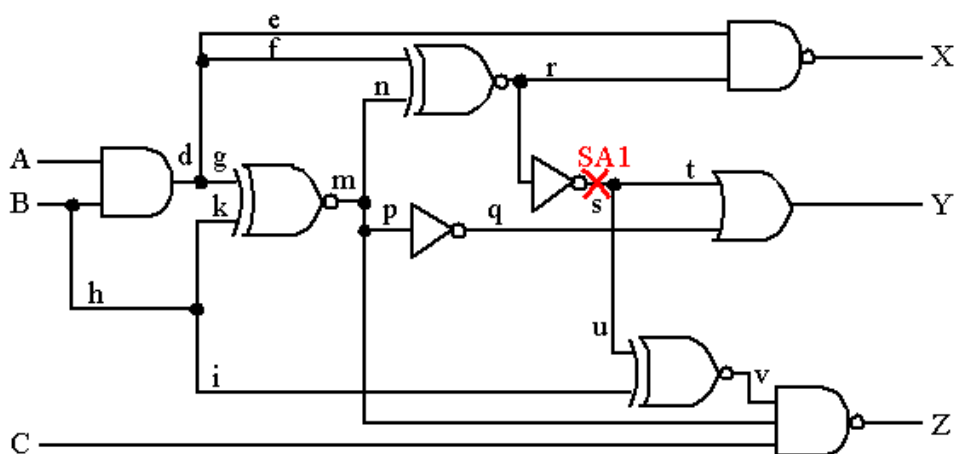
- a) Using memory blocks of $4k \times 16$ bits, design a memory that is $16k \times 64$ bits.
- b) Describe what is bit-slicing and banking in memories. What is the use of each one of them?

Answer:

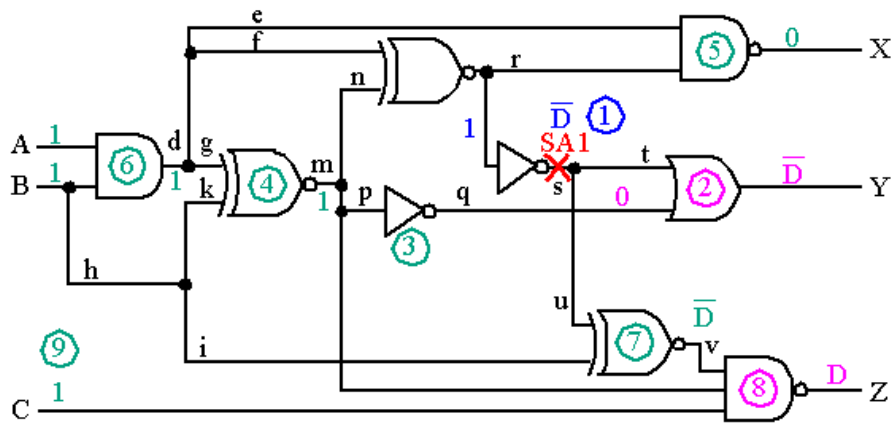
Lecture on Interconnect and Memories slides 21-23

Question 7: (10 points)

Show how to activate and propagate the stuck-at-1 fault in wire "s". What is the test vector that achieves that?



Answer:



Assign PDF Propagate Consistency

C can be both 0 or 1, if C=1 then Z is D

Question 8: (10 points)

Consider a 3-stage pipeline with stage-1 and stage-3 having 3 ns delay each, and stage-2 having 5 ns delay.

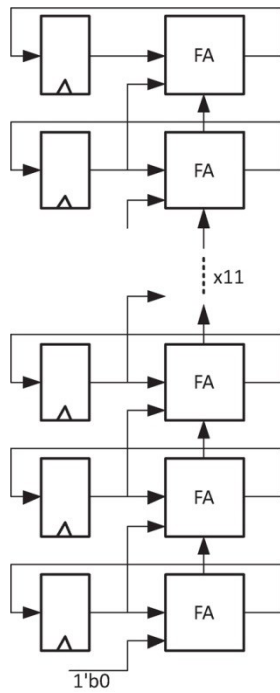
- What is the throughput of the pipeline? Compare it with a non-pipelined version of the design.
- Describe the 2 alternative techniques to improve the throughput of the pipeline. When would you use each one of them? What is the pipeline throughput then?
- Consider now that stage-2 has a variable delay of either 3ns or 5 ns (50% probability to have a 3ns or a 5 ns delay) and none of the alternatives you described in (b) are an option. Then, what can one do to improve the throughput of the pipeline? What would be the best-case throughput then?

Answer:

- 1 output every 5 ns (200 Mops), non pipelined version would be 1 output every 11 ns (90 Mops)
- Parallelism or pipeline of stage 2, then throughput would be 1 output every 3 ns (333 Mops)
- Adding a FIFO between stage-1 and stage-2 would give us at best (if FIFO is not full and the stage-2 delay is 50%-50% 3 ns or 5 ns) 1 output every 4 ns (250 Mops)

Question 9: (10 points)

Find the critical path of the following design (16-bit ripple carry adder with registered input/outputs) considering that a full-adder (FA) has a delay of 3 ns and a flip-flop has propagation delay of 1 ns and setup time of 1 ns, while the input 1'b0 of the design has no delay. What is the maximum frequency of the design?



Answer:

- a) $1 + 3 \cdot 16 + 1 = 50$ ns, 20 MHz

Question 10: (10 points)

Analyze an SR-latch as an asynchronous circuit:

- Draw the block diagram of the SR latch showing its current state (y) and next state (Y). Note that only output Q is needed (Q' is not necessary).
- What is the equation of the next state Y ? Consider that S and R could be "1" at the same time.
- Draw the transition table and flow table of the SR latch?
- What is the output of the SR latch when SR change from "11" to "00"?

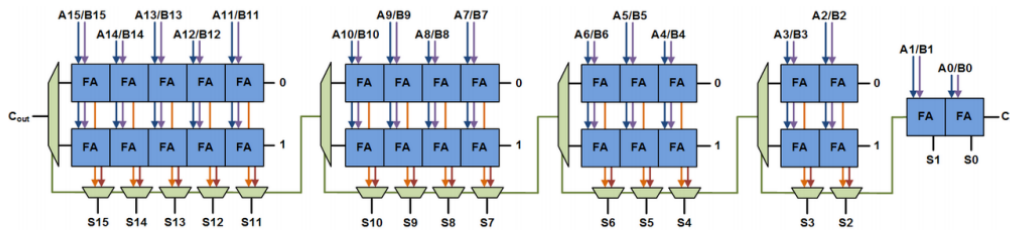
Answer:

Lecture on Asynchronous Sequential Logic slides 18-20

END of EXAM

Question 1:

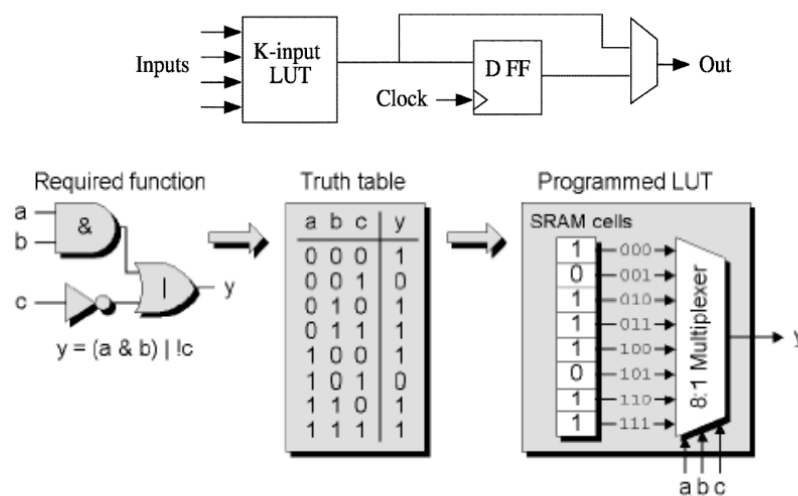
Carry Select Adder variable size blocks



16-bit adder Carry select adder with variable block sizes [5 4 3 2 2]

Question 3:

Mapping logic to a Logic Cell



Question 4:

Floating point numbers

- Mantissa m : binary fraction
- Exponent e : binary integer

$$v = m \times 2^{e-x} = r \sum_{i=0}^{n-1} m_i 2^{i-n} \times 2^{\sum_{i=0}^{k-1} e_i 2^k - x}$$

E.g.: $m = 10010$, $e = 011$ (10010E011): $v = 18/32 \times 8$

Could also be written 01001E100: $v = 9/32 \times 16$,

but this is not normalized form of a floating point number

(mantissa should start with 1 unless exponent is 0)

Floating point number: $e_{k-1}, e_{k-2}, \dots, e_1, e_0 \mid m_{n-1}, m_{n-2}, \dots, m_1, m_0$

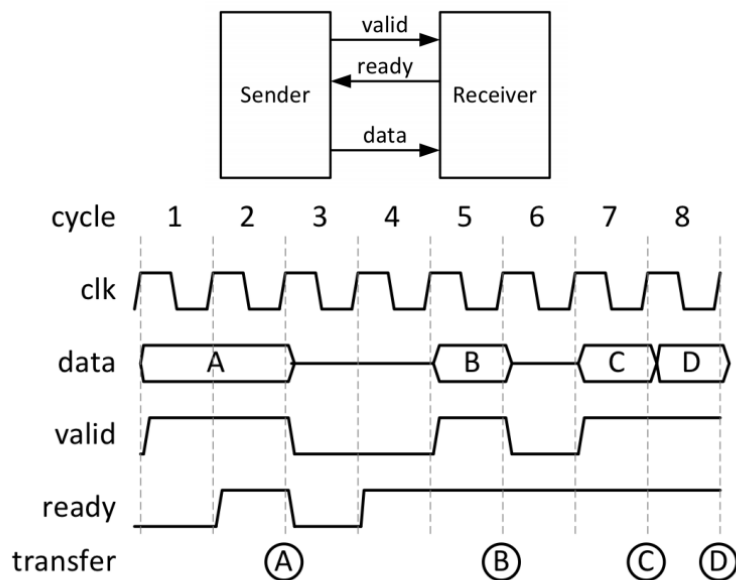
EDA322 Digital Design, 2017-2018, Lecture 12

I. Sourdis, CSE, Chalmers

32

Question 5:

Flow Control



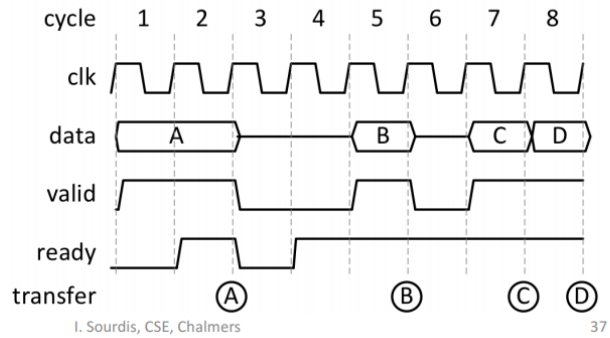
EDA322 Digital Design, 2017-2018, Lecture 13

I. Sourdis, CSE, Chalmers

36

Flow-control

- Valid – Tx has data available
- Ready – Rx able to take data
- Push flow control – assume Rx always Ready
- Pull flow control – assume Tx always Valid



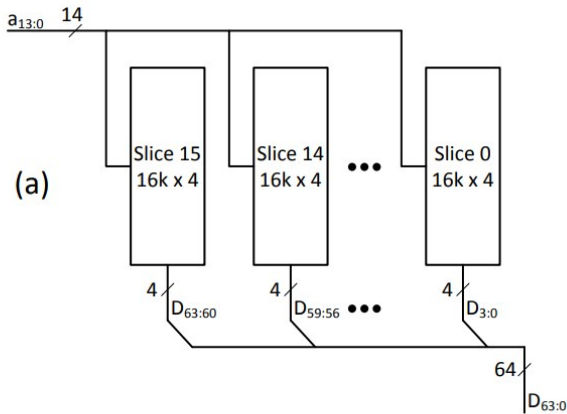
EDA322 Digital Design, 2017-2018, Lecture 13

I. Sourdis, CSE, Chalmers

37

Question 6:

Bit-Slicing

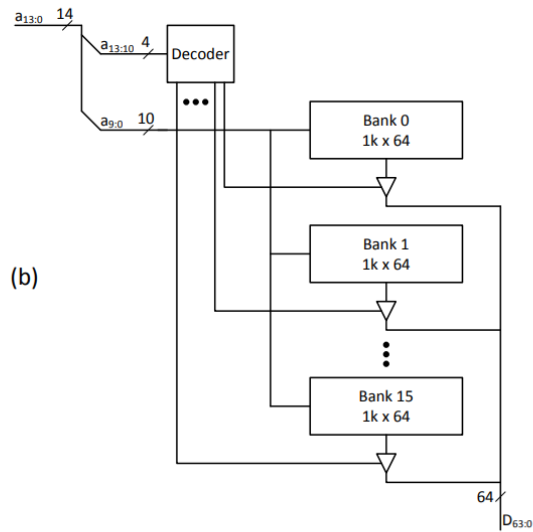


EDA322 Digital Design, 2017-2018, Lecture 14

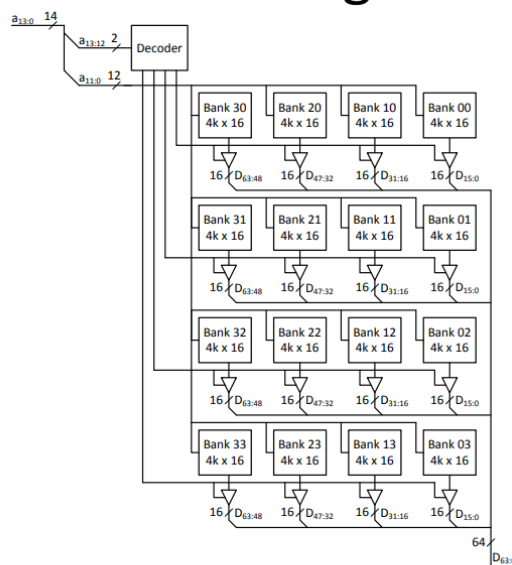
I. Sourdis, CSE, Chalmers

21

Banking



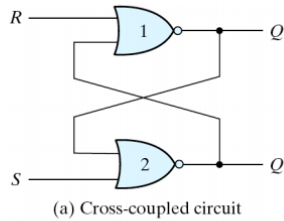
Bit slicing & banking



Question 10:

Analysis of an SR latch (1)

- We can analyze an SR latch using the previous technique



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

(b) Truth table

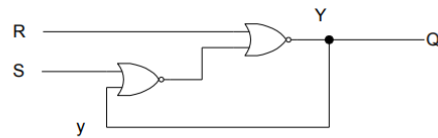
(After SR = 10)
(After SR = 01)

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

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

- Since we want to avoid the SR=11 situation, we can write:

$$Y = S + \overline{R}y \text{ if } SR = 0$$

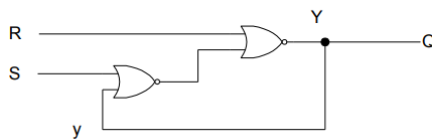


Analysis of an SR latch (2)

- Can derive the transition table and the flow table:

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

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



Analysis of an SR latch (3)

- Note: We can see the undesirable case when $SR=11$ and inputs change.
- Depending on the various delays and assuming $SR=11$ changes to $SR=00$...
 - If $SR=11 \rightarrow SR=10 \rightarrow SR=00$, we get stable state with output of 1.
 - If $SR=11 \rightarrow SR=01 \rightarrow SR=00$, we get stable state with output of 0.
- So the stable state is not predictable.
- Conclusion is that we need to be careful if we (possibly) need to transition from one state to another and we (somehow) pass through state 11.