

EDA322: Digital Design

Exam - June 2017

Date: June 9, 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 **June 29th 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)

Design in gatelevel a multiplier that multiplies a 4-bit integer $A(3:0)$ with the constant decimal value "10" ('1010' in binary).

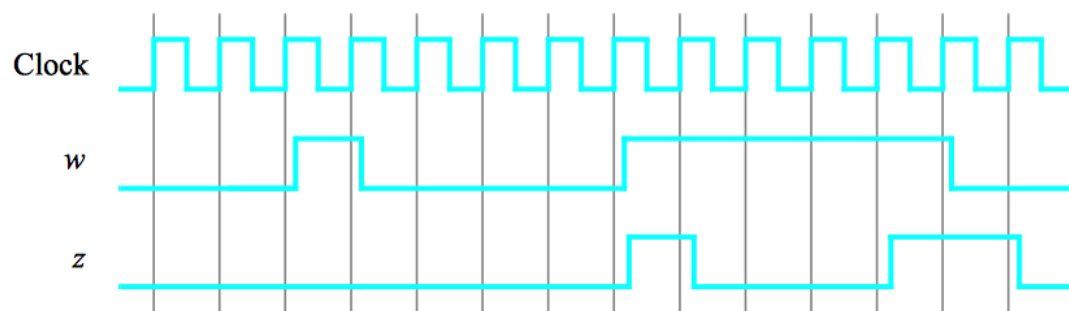
Answer:

Lecture on Arithmetic Units, slide 6 with an additional least significant bit that is always zero. In addition the internals of the 4-bit adder should be shown, which could be a ripple carry adder.

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

Question 2: (10 points)

Design a synchronous FSM (Finite State Machine) that recognizes two specific sequences of applied input symbols, namely four consecutive 1s ("1111") or four consecutive 0s ("0000"). There is an input w and an output z . Whenever $w = 1$ or $w = 0$ for four consecutive clock pulses the value of z has to be 1; otherwise, $z = 0$. Overlapping sequences are allowed, so that if $w = 1$ for five consecutive clock pulses the output z will be equal to 1 after the fourth and fifth pulses. The Figure bellow illustrates the required relationship between w and z .



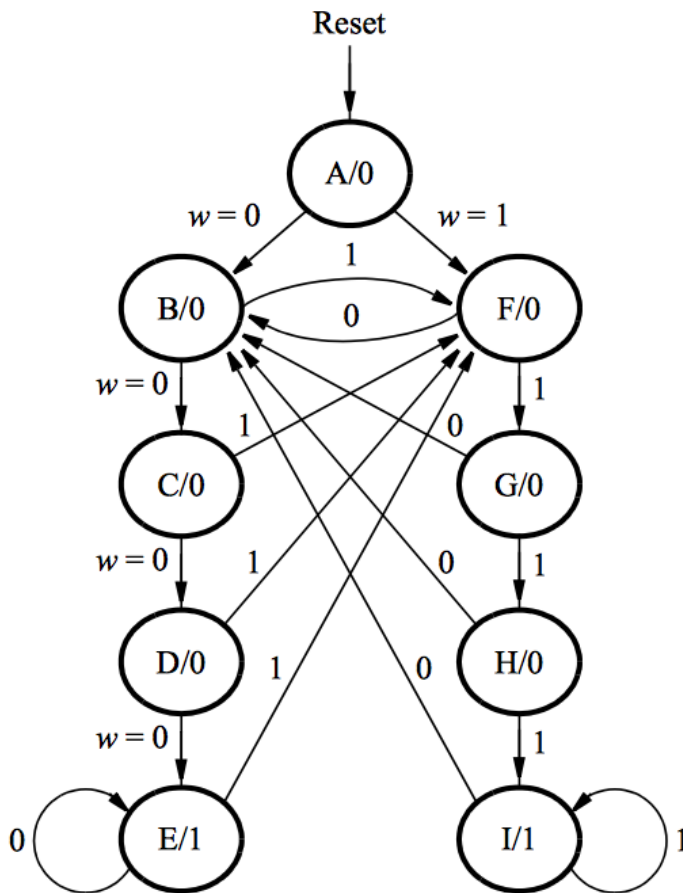
- Draw the state-diagram of the FSM
- Fill in the state-table of the FSM
- Make a state-assignment and derive the Boolean functions that produce the next state bits and the output.
- Draw the hardware design (block diagram) of the FSM.

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:

Similar to the example of lecture on Finite State Machines (FSMs) slides 6-12

One possible state diagram:



Question 3: (10 points)

- What is an FPGA configuration bitstream (also called datastream)? Which FPGA parts do the bits of a bitstream configure and how do they control the functionality of these parts?
- What types of reconfiguration memories are there? What are the advantages and disadvantages of each?

Answer:

Lecture on Technology – Reconfigurable Hardware, slides 46-48

Question 4: (10 points)

Draw and describe a bus, a crossbar and an interconnection network. What are their differences?

Answer:

Lecture on Interconnect and Memories, slides 5-8

A bus needs less area (logic and wires) than a crossbar or a network but allows one sender to use the bus at any point in time. Its performance becomes poor as the number of clients increase

A crossbar requires more area than a bus. The area grows in $O(n^2)$ where n is the number of clients). It allows every input to send data to a different output in one hop (cycle).

A network scales its area better than a crossbar (as the number of clients increase). It allows multiple clients to concurrently send data to the same or different destinations. Each transaction may take multiple hops to complete.

Question 5: (10 points)

Draw the block diagram of a 4x4 memory array (4 rows of 4-bits) using D-flip-flops. Find the maximum clock frequency of the memory block considering that:

- a 2-input AND gate or a 2-input OR gate has a delay of 1 ns,
- the setup time of a D flip flop is 0.5 ns,
- the hold time of a D flip flop is 0.1 ns,
- the propagation delay of a D flip flop is 0.5 ns,
- inverters and wires have zero delay

Answer:

Lecture on Interconnect and Memories slide 15,
6 ns -> 166.66 MHz.

Question 6: (10 points)

The cost of a chip is 15 SEK when the yield is 75%. What should be the yield for the chip price to go down to 12 SEK?

Answer:

The cost of a chip (good or bad) is:
 $\text{chip-cost} = \text{Wafer cost} / \text{total\#chips}$

the cost of good chips is

$\text{Good-chip-cost} = \text{Wafercost} / (\text{total\#chips} * \text{yield})$

Wafercost (WC) and total#chips (n) are constants

So with yield 0.75 the yield = $\text{WC} / \text{n} * 15\text{SEK}$

With yield y yield = $\text{WC} / \text{n} * 12\text{SEK}$

So:

$$y = 0.75 * [(\text{WC} / \text{n} * 12\text{SEK}) / (\text{WC} / \text{n} * 15\text{SEK})] = 93.75\%$$

Question 7: (10 points)

Considering that the delay of a full adder (FA) is 1 ns, the setup time of a flip-flop is 0.1 ns, the hold time of a flip-flop is 0.01ns and the propagation time of a flip-flop is 0.1 ns. Find the latency, throughput and minimum clock period of the following 8-bit ripple carry adder designs:

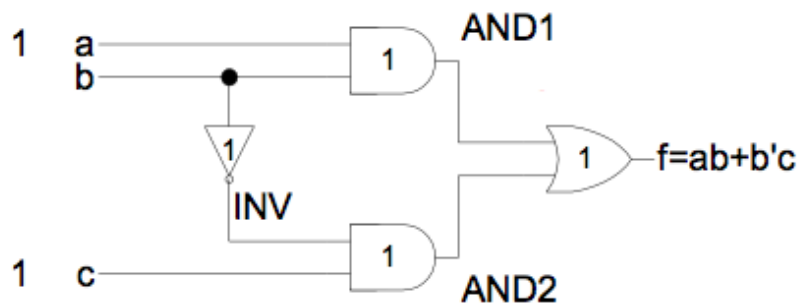
- a) Non-pipelined (1 stage of 8-bits)
- b) Pipelined in 2 stages of 4-bits each
- c) Pipelined in 4 stages of 2-bits each
- d) Pipelined in 8 stages of 1-bits each

Answer:

- a) 8 ns latency, 1 operation every 8 ns (125 MOPS), 8ns period
- b) $2*(4+0.2)= 8.4$ ns latency, 1 operation every 4.2 ns (238 MOPS), 4.2ns period
- c) $4*(2+0.2)= 8.8$ ns latency, 1 operation every 2.2 ns (454 MOPS), 2.2ns period
- d) $8*(1+0.2)= 9.6$ ns latency, 1 operation every 1.2 ns (833 MOPS), 1.2ns period

Question 8: (10 points)

Find the hazard in the circuit below and fix it



Answer:

Lecture on Asynchronous Sequential Logic, slides 71-74

Question 9: (10 points)

Describe the following terms and explain what can be done to minimize their impact:

- a) clock skew
- b) metastability

Answer:

Lecture on Timing, Delay, Power, slides 23-24 (+ avoid adding delays to the clock trees), 28-32

Question 10: (10 points)

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

Answer:

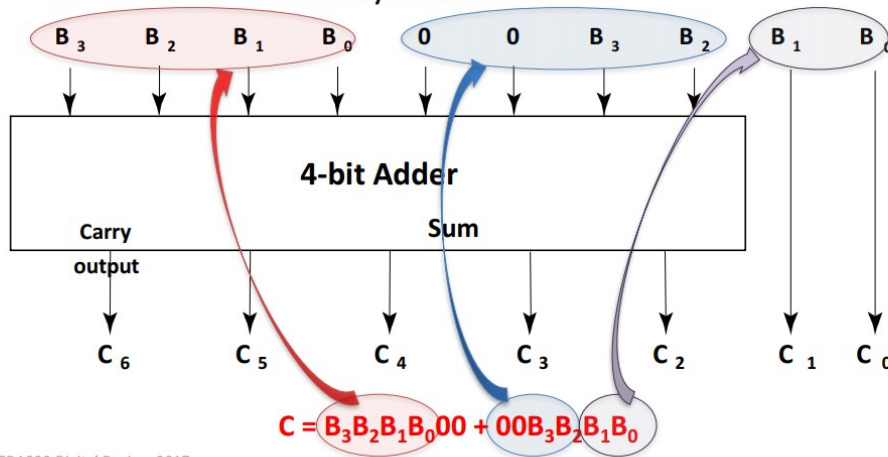
Based on lecture titled "System Design and Interfaces", slide 36-37

END of EXAM

Question 1:

Multiplication by a Constant

- Multiplication of B(3:0) by 101
- $C = 4*B + 1*B = B \ll \text{by two bits} + B$



EDA322 Digital Design, 2017-2018, Lecture 12

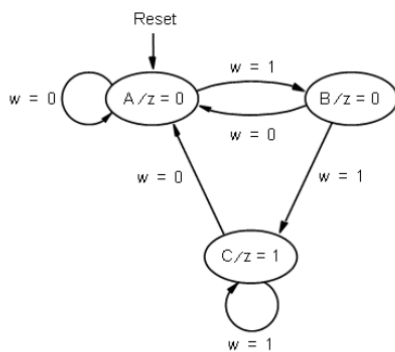
I. Sourdis, CSE, Chalmers

6

Question 2:

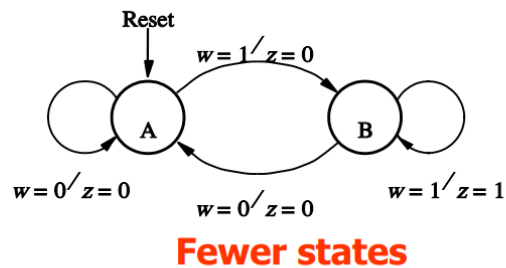
Example: 1-1 detector: state diagram

Moore-type



EDA322 Digital Design, 2017-2018, Lecture 8

Mealy-type



Fewer states

I. Sourdis, CSE, Chalmers

6

Example: 1-1 detector state table

Moore-type

Present state	Next state		Output z
	w = 0	w = 1	
A	A	B	0
B	A	C	0
C	A	C	1

Mealy-type

Present state	Next state		Output z	
	w = 0	w = 1	w = 0	w = 1
A	A	B	0	0
B	A	B	0	1

Present state	Next state		Output z
	w = 0	w = 1	
$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	
A 00	00	01	0
B 01	00	10	0
C 10	00	10	1
	dd	dd	d

Present state	Next state		Output	
	w = 0	w = 1	w = 0	w = 1
y	Y	Y	z	z
A 0	0	1	0	0
B 1	0	1	0	1

EDA322 Digital Design, 2017-2018, Lecture 8

I. Sourdis, CSE, Chalmers

7

Example: 1-1 detector Boolean expressions derivation (Moore)

From state-assigned table: we have the following Karnaugh maps

$y_2 y_1$	00	01	11	10
0	0	0	d	0
1	1	0	d	0

$$Y_1 = w y_1 \bar{y}_2$$

y_1	0	1
0	0	0
1	1	d

$$z = y_2$$

$y_2 y_1$	00	01	11	10
0	0	0	d	0
1	0	1	d	1

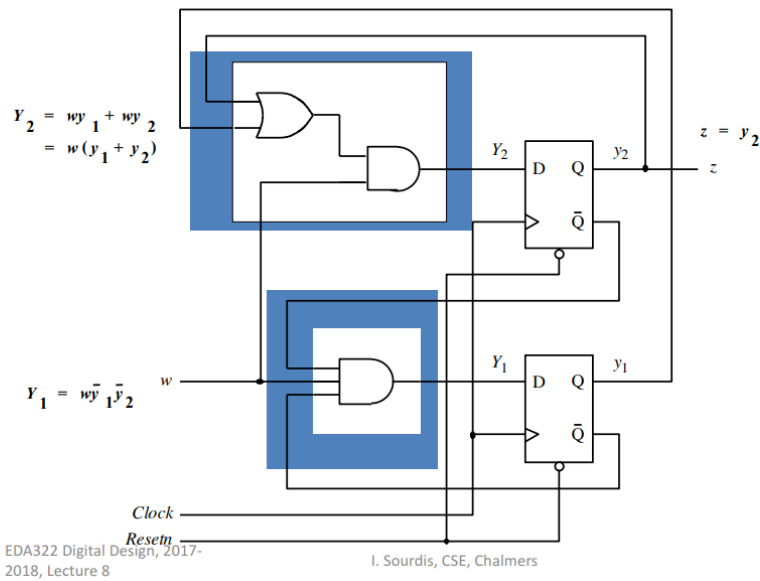
$$Y_2 = w y_1 + w y_2 = w (y_1 + y_2)$$

EDA322 Digital Design, 2017-2018, Lecture 8

I. Sourdis, CSE, Chalmers

8

Example: 1-1 detector Final implementation (Moore)

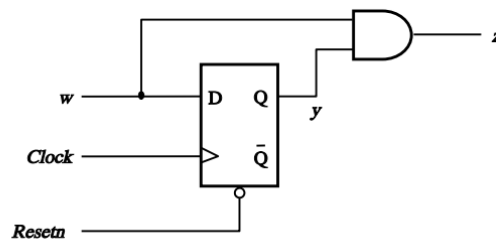


9

Example: 1-1 detector Final implementation (Mealy)

$$Y = w$$

$$z = wy$$



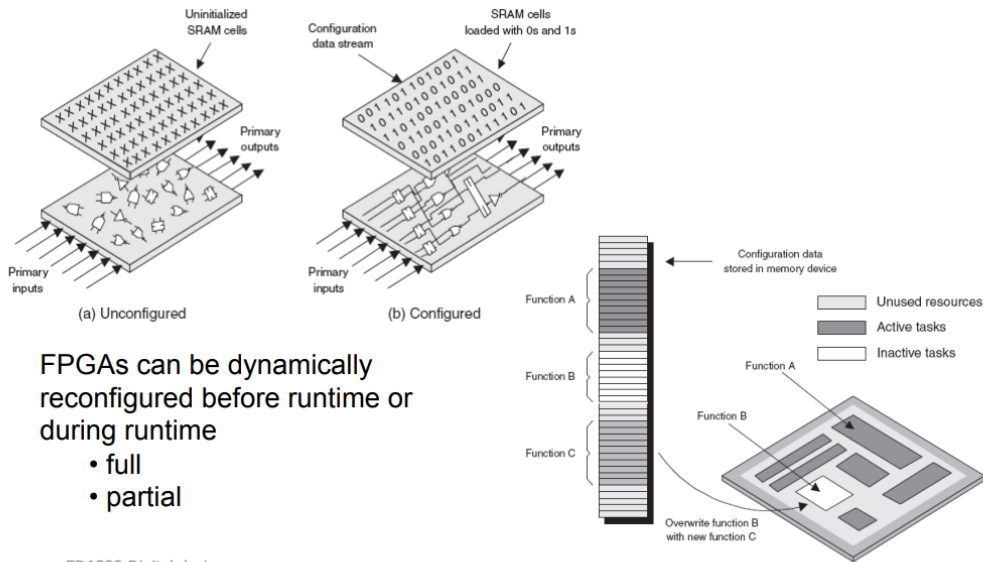
EDA322 Digital Design, 2017-2018, Lecture 8

I. Sourdis, CSE, Chalmers

10

Question 3:

Reconfiguring FPGAs



FPGAs can be dynamically reconfigured before runtime or during runtime

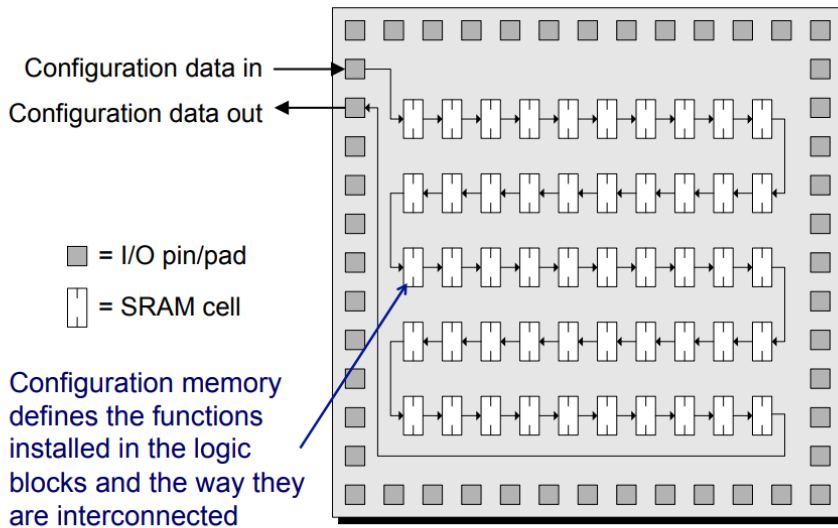
- full
- partial

EDA322 Digital design, 2016-2017, Lecture 11

I. Sourdis, CSE, Chalmers

46

Reconfiguring an FPGA



EDA322 Digital design, 2016-2017, Lecture 11

I. Sourdis, CSE, Chalmers

47

Reconfiguration Memory

Anti-fuse



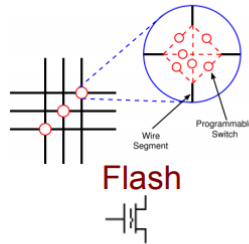
by default is OFF; when programmed it is ON (creating a short-circuit between the endpoints).

Advantages:

- negligible delay
- small area overhead
- no soft-errors and bit flips

Disadvantages:

- not really reconfigurable; one time programmable



Flash



Advantages:

- programming not lost when device is turned off.
- Fewer transistors than SRAM
- Lower power than SRAM

Disadvantages:

- Limited writes (~millions)
- Slower writes than SRAM
- Higher voltage than circuits

SRAM



SRAM bit cell stores the programmability of the device

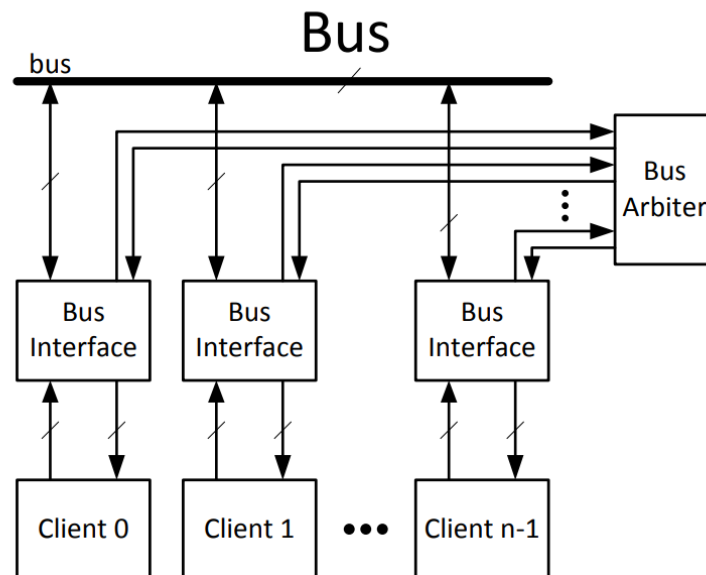
Advantages:

- can be reconfigured quickly and as repeatedly as required
- no special fabrication steps

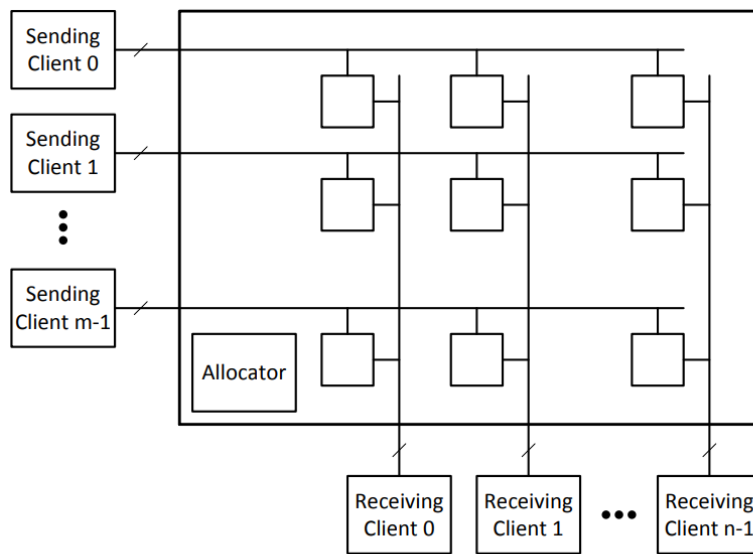
Disadvantages:

- takes more area, and power
- loses charge when turned off

Question 4:



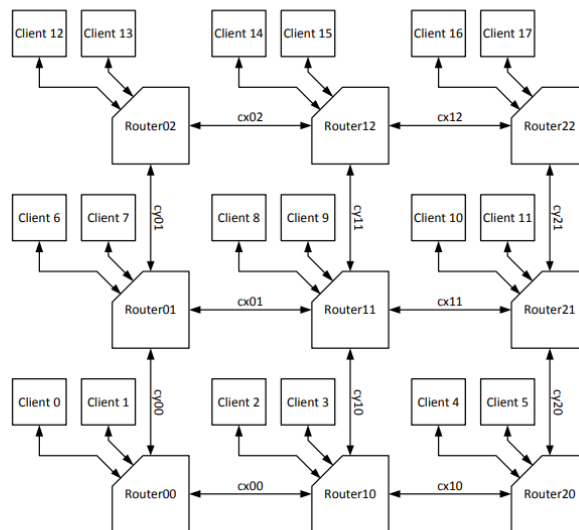
Crossbar Switch



EDA322 Digital Design, 2017-2018, Lecture 14
I. Sourdis, CSE, Chalmers

6

Interconnection Networks



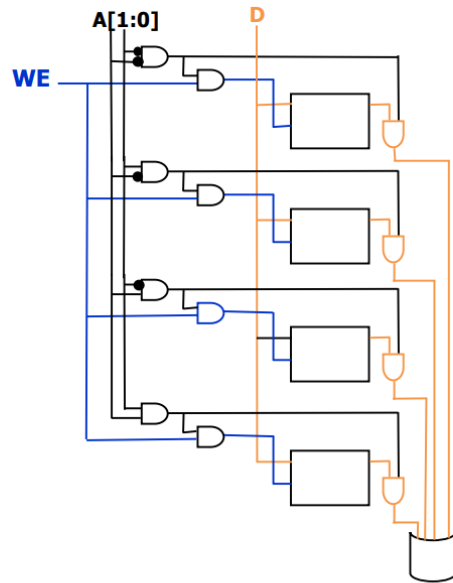
EDA322 Digital Design, 2017-2018, Lecture 14
I. Sourdis, CSE, Chalmers

7

Question 5:

Building a Memory

- Each bit
 - is a gated D-latch
- Each location
 - consists of w bits (here $w = 1$)
 - $w = 8$ if the memory is byte addressable
- Addressing
 - n locations means $\log_2 n$ address bits (here 2 bits \Rightarrow 4 locations)
 - decoder circuit translates address into 1 of n locations



EDA322 Digital Design, 2017-2018, Lecture 14

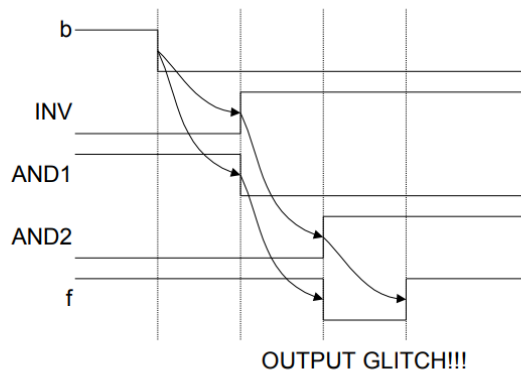
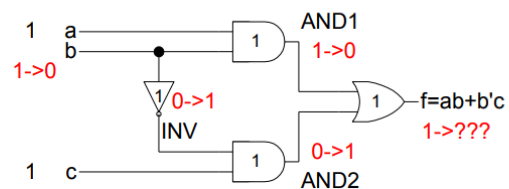
I. Sourdis, CSE, Chalmers

15

Question 8:

Illustration

- Consider the following circuit with delays where only one input (input b) changes...
- Draw a timing diagram to see what happens at output with delays.
- From the logic expression, we see that b changing should result in the output remaining at logic level 1...
- Due to delay, the output goes 1- \rightarrow 0- \rightarrow 1 and this is an output glitch; **we see a static-1 hazard.**



EDA322 Digital design, 2017-2018, Lecture 19

I. Sourdis, CSE, Chalmers

71

Fixing hazards (2-level circuits) (1)

- When circuits are implemented as **2-level SOP (2-level POS)**, we can detect and remove hazards by inspecting the K-Map and **adding redundant product (sum) terms**.

		bc	00	01	11	10
a	0	0	1	0	0	
	1	0	1	1	1	

$$f=ab+b'c$$

- Observe that when input b changes from 1->0 (as in the previous timing diagram), that we “jump” from one product term to another product term.
 - If adjacent minterms are not covered by the same product term, then a HAZARD EXISTS!!!**

EDA322 Digital design, 2017-2018, Lecture 19

I. Sourdis, CSE, Chalmers

72

Fixing hazards (2-level circuits) (2)

		bc	00	01	11	10
a	0	0	1	0	0	
	1	0	1	1	1	

$$f=ab+b'c + ac$$

- The extra product term does not include the changing input variable, and therefore serves to prevent possible momentary output glitches due to this variable.

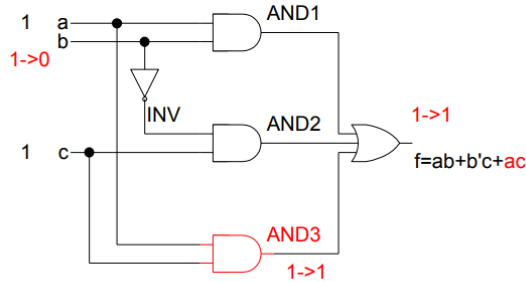
EDA322 Digital design, 2017-2018, Lecture 19

I. Sourdis, CSE, Chalmers

73

Fixing hazards (2-level circuits) (3)

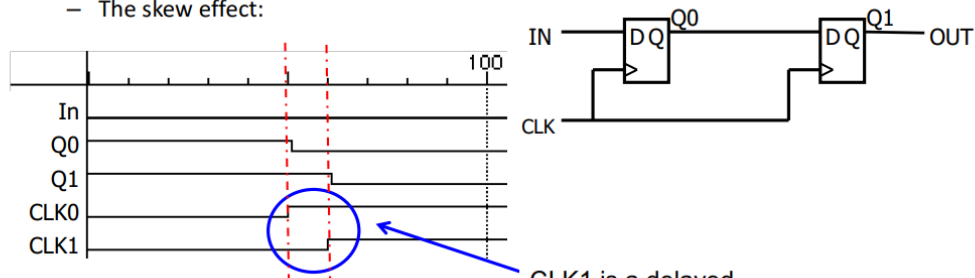
- The redundant product term is not influenced by the changing input.



Question 9:

Clock Skew

- The Problem
 - correct behavior requires that the next state of all memory elements are determined by all the memory elements thus at the same time
 - This is difficult in high performance systems since the time it takes for the clock to arrive, are of the same magnitude as the delay through the logic
 - The skew effect:

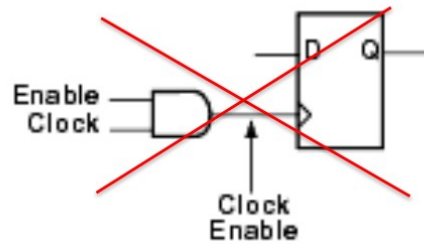


Initial state: IN = 0, Q0 = 1, Q1 = 1
 Due to skew next state is : Q0 = 0, Q1 = 0,
 instead of Q0 = 0, Q1 = 1

CLK1 is a delayed version of CLK0

Treating the clock right

- No clock-gating unless using dedicated synthesis libraries!
 - Otherwise clock skew is introduced
 - Use instead Enable

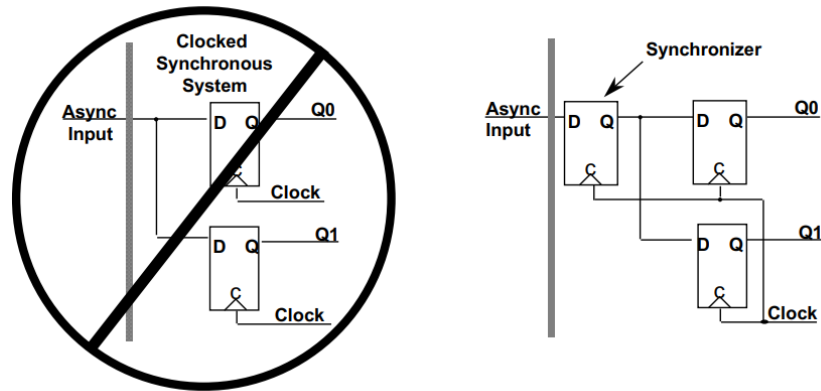


Metastability and Asynchronous Inputs

- Asynchronous Inputs Are Dangerous!
 - Since they take effect immediately, glitches can be disastrous
 - Synchronous inputs are greatly preferred!
 - But sometimes, asynchronous inputs cannot be avoided
 - e.g., reset signal, memory wait signal

Metastability and Asynchronous Inputs

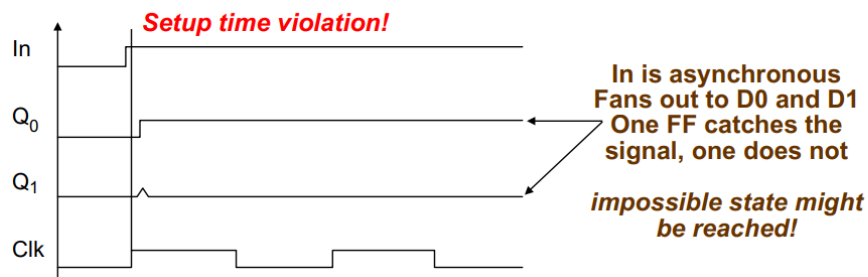
Handling Asynchronous Inputs



Never allow asynchronous inputs to be fanned out to more than one FF within the synchronous system

Metastability and Asynchronous Inputs

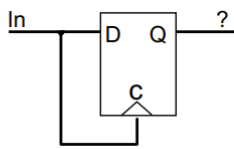
What Can Go Wrong



Single FF that receives the asynchronous signal is a synchronizer

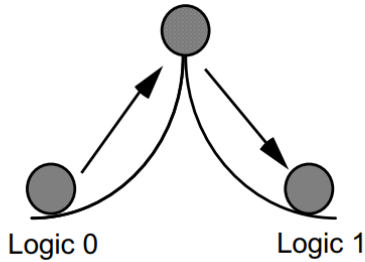
Metastability and Asynchronous Inputs

Synchronizer Failure

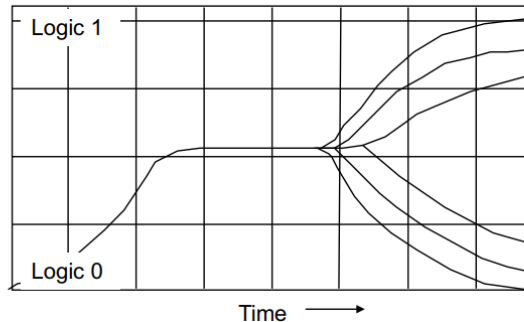


When FF input changes close to clock edge, the FF may enter the *metastable state*: neither a logic 0 nor a logic 1

It may stay in this state an indefinite amount of time, although this is not likely in real circuits



Small, but non-zero probability that the FF output will get stuck in an in-between state

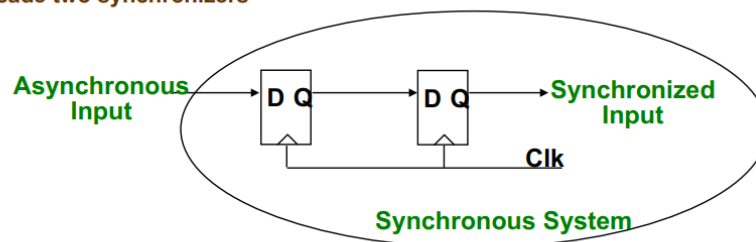


Oscilloscope Traces Demonstrating Synchronizer Failure and Eventual Decay to Steady State

Metastability and Asynchronous Inputs

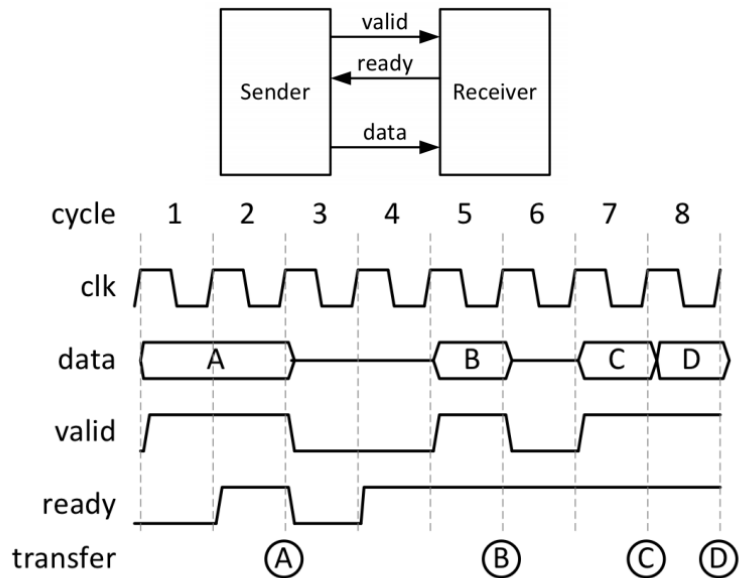
Solutions to Synchronizer Failure

- the probability of failure can never be reduced to 0, but it can be reduced
- slow down the system clock
this gives the synchronizer more time to decay into a steady state
synchronizer failure becomes a big problem for very high speed systems
- use fastest possible logic in the synchronizer
this makes for a very sharp "peak" upon which to balance
S or AS TTL D-FFs are recommended
- cascade two synchronizers



Question 10:

Flow Control



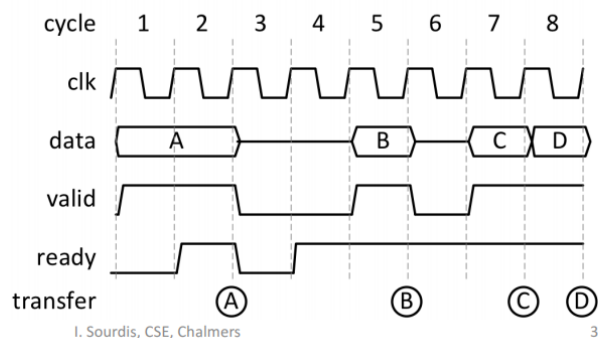
EDA322 Digital Design, 2017-2018, Lecture 13

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