Final

E3.05
ISE3.19

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2010

EEE/ISE PART III/IV: MEng, BEng and ACGI

DIGITAL SYSTEM DESIGN

Friday, 7 May 10:00 am

Time allowed: 3:00 hours

There are SIX questions on this paper.

Answer FOUR questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : C. Bouganis
Second Marker(s) : T.J.W. Clarke

Special information for invigilators:
None

Information for candidates:
In the figures showing digital circuits, all components have, unless explicitly indicated otherwise, been drawn with their inputs on the left and their outputs on the right. All signals labelled with the same name are connected together. All circuits use positive logic. The least significant bit of a bus signal is labelled as bit 0, and the most significant bit with the highest integer number. Therefore the signal X[7:0] is an eight bit bus with X7 being the MSB and X0 the LSB.

Hexadecimal numbers are prefixed with $. For example the decimal number 10 is written as $A.

In questions involving circuit design, you may use any standard digital circuits that are not explicitly forbidden by the question provided that you fully specify their operation.

Marks may be deducted for unnecessarily complex designs unless you are explicitly instructed not to simplify your solution.

The Questions

1. a) Figure 1.1 shows the bit-parallel iterative implementation of a CORDIC processor. Hence or otherwise, derive the iterative equations of the rotation mode CORDIC algorithm as implemented by the circuit shown.

[10]

b) Table 1 illustrates a set of angles and their corresponding arctan values relevant to the CORDIC algorithm. Justify the reason for selecting those elementary angles. If the only angles that are available to our CORDIC implementation are the ones displayed in Table 1, what is the maximum angle that can be approximated with zero error?

[4]

c) Explain how the CORDIC algorithm can be used to evaluate $\tan(a)$. Assume that the value of K is known.

[3]

d) State with justification how many CORDIC iterations are required to evaluate $\cos(a)$, where $a = 18.4350^o$. Assuming the architecture of Figure 1.1, how many clock cycles are required for the above evaluation? The elementary angles used in the CORDIC implementation are illustrated in Table 1.

[3]

Table 1 Elementary angles and their tangents

e	**atan(e) in degrees**
1	45.000
0.5	26.565
0.25	14.036
0.125	7.125
0.0625	3.576
0.03125	1.790
0.0115625	0.895

[continued on the following page]

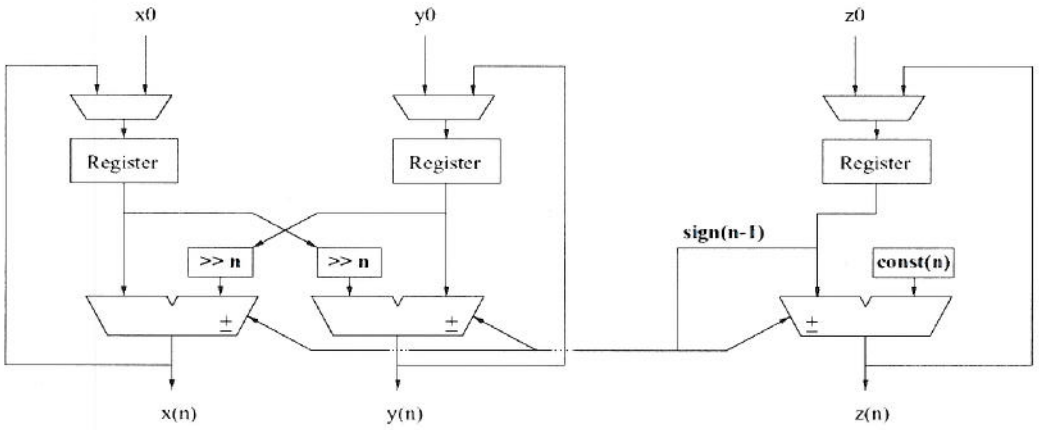


Figure 1.1 CORDIC architecture

2. a) Figure 2.1 illustrates a common Phase Locked Loop (PLL) system. Provide a description of its operation.

[5]

b) Compare the given PLL system to a Delay Locked Loop (DLL) system.

[5]

c) Derive the relationship between the output frequency F_{OUT} and the input frequency F_{IN} as a function of N, M, and C.

[5]

d) An input clock signal with frequency 30MHz is given. State the values of N, M, and C, if F_{VCO} is in the range 100-200MHz, in order to produce a clock signal with target frequency of 75MHz.

[5]

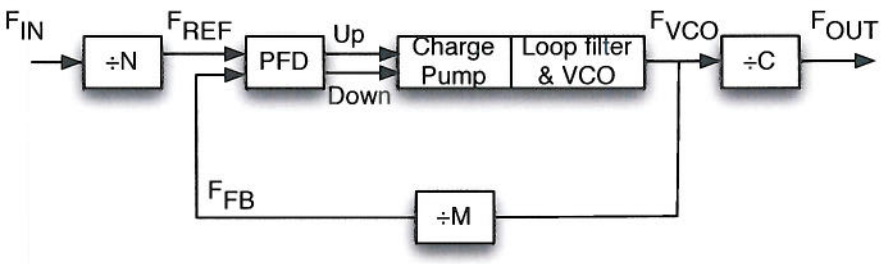


Figure 2.1 A Phase Locked Loop

3. a) Design a system that performs the 4-tap FIR filtering defined by the following expression

$$y(n) = 5 * x(n) + 3 * x(n-1) + 2 * x(n-2) + x(n-3)$$

where x is the input signal and y is the output signal. Your design should be based on a bit-serial implementation that employs distributed arithmetic. The input samples enter the system LSB first and take integer values in the range $[0, 255]$. You do not need to draw the system control signals.

[12]

b) State the length of the adder used in the design.

[2]

c) Assuming that there is a constant flow of input data, comment on the throughput of the system (evaluations / clock cycles).

[4]

d) Specify the size of the RAM used in the design (number of entries and bits per entry), Write its contents in decimal format.

[2]

4. a) Figure 4.1 depicts the state diagram of a 8-state finite state machine (FSM), with an output A. The output signal is set high in state 7, and low in all other states. Using three D flip-flops and logic, design a circuit that implements the FSM.

[10]

b) Assuming that the logic is mapped to 4-LUTs when the target device is an FPGA, estimate the number of 4-LUTs required to implement the FSM.

[5]

c) Estimate the maximum frequency of the design, given that the propagation delay of a 4-LUT is $2ns$, and the set-up time, the hold time and propagation time of a D flip-flop are $2ns$, $1ns$ and $3ns$ respectively.

[3]

d) Discuss whether it is possible to increase the clock frequency of the design by employing pipeline techniques without affecting its functionality.

[2]

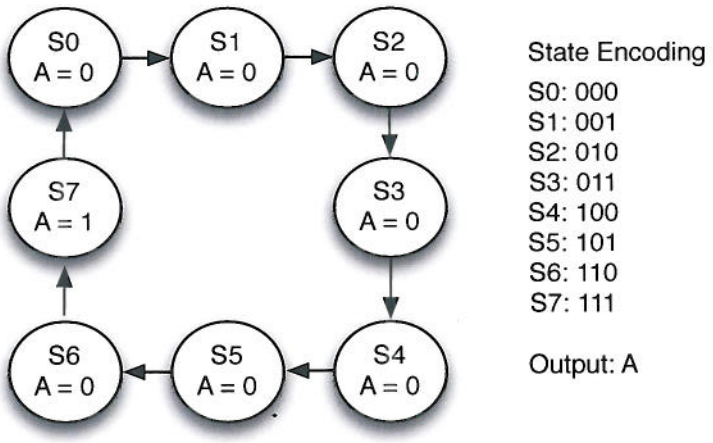


Figure 4.1 State diagram

5. a) A block diagram of a 1 bit full adder (FA) is given in Figure 5.1. The equations that describe its outputs are $s = x \oplus y \oplus c_{in}$, and $c = x \cdot y + c_{in} \cdot (x \oplus y)$.

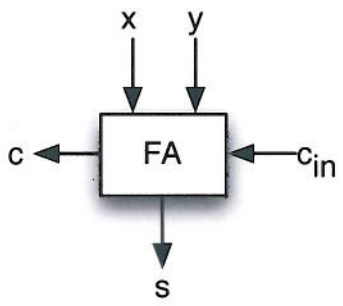


Figure 5.1 Full Adder

Assuming that only 4-LUTs are used when such a design is mapped on an FPGA, estimate the number of LUTs required for one 1 bit full adder.

[3]

b) Using the above FA as a building block, design an 8-bit ripple-carry adder that takes as input two 8-bit numbers and produces an 8 bit output (i.e. ignore the carry from the MSB stage).

[3]

c) State the critical path, and derive the maximum frequency of the design, assuming that all the logic is mapped to LUTs. Assume that the propagation delay of a LUT is $2ns$.

[4]

d) Design an 8-bit carry-select adder using as building blocks three 4-bit ripple-carry adders. You can ignore the carry from the MSB stage.

[3]

e) A 1-bit two to one multiplexer can be implemented using one 4-LUT. Estimate the total number of 4-LUTs required for the above 8-bit carry-select adder design.

[3]

f) State the maximum frequency of the above 8-bit carry-select adder design, assuming that the propagation delay of a LUT is $2ns$.

[4]

6. a) Figure 6.1 shows a circuit that performs FIR filtering according to the following equation

$$y(n) = c1 * x(n) + c2 * x(n-1) + c3 * x(n-2)$$

where $c1, c2, c3$ are coefficients that have integer values. The system produces one output per clock cycle. The propagation delays of the components of the system are given in Table 1. Assume that the wires do not exhibit any delay, and all registers are clocked with the same clock signal. State and justify the critical path in this design. What is the maximum frequency of the design?

[10]

b) Assume that you cannot change the adder and multiplier blocks. Explain how you can increase the maximum frequency of the design and state the resulting frequency. Comment on the resulting throughput and latency of the modified design.

[7]

[continued on the following page]

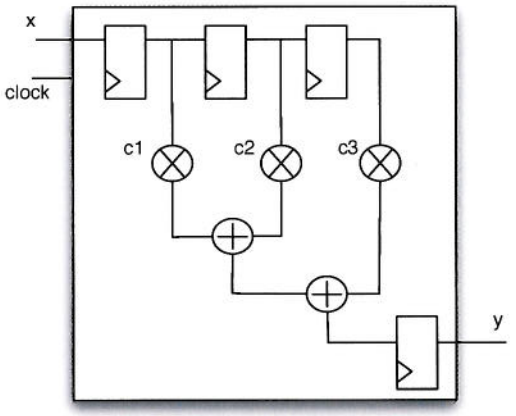


Figure 6.1 FIR system

Table 1 Propagation delay of the Components

symbol	value	description
T_p^{reg}	5ns	register propagation delay
T_{setup}^{reg}	2ns	register setup time
T_{hold}^{reg}	3ns	register hold time
T_p^{mul}	10ns	multiplier propagation time
T_p^{add}	8ns	adder propagation time

c) The design is used to filter an asynchronous signal. Explain the problems that may rise by feeding to the FIR design an asynchronous input. A synchronizer with a single register is added to the original design as shown in Figure 6.2. Calculate the MTBF when the design is clocked a 40 MHz clock. The rate of asynchronous transitions per second is 40, and $\tau = 0.5ns$ and $T_0 = 9.5x10^{12}sec.$

[3]

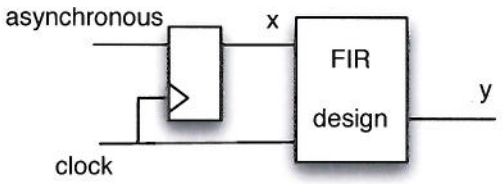


Figure 6.2 FIR system synchronizer

SOLUTIONS - DIGITAL SYSTEM DESIGN 2010

EJ.05
Ex 3.19

1. [Background and application to a new example.]

a) The iterative equations are: 1/12

$$x_{i+1} = x_i - d_i \cdot y_i \cdot 2^{-i}$$
$$y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i}$$
$$z_{i+1} = z_i - d_i \cdot \alpha_i$$

where $d_i \in \{-1, 1\}$ as determined by some criterion

[10]

b) The selection of those angles is because their corresponding tan values are powers of two.

The maximum angle is the sum of all the angles:

$$\theta = 45.000 + 26.565 + \cdots + 0.895 = 98,987^\circ$$

[4]

c) $\tan(\alpha) = \dfrac{\sin(\alpha)}{\cos(\alpha)}$

1. Set $z = \alpha$
2. $x = 1/K = 0.607252935$
3. $y = 0$
4. Iterate with $d_i = \text{sign}(z_i)$

After n rotations we get: $x_n = \cos(\alpha)$

$y_n = \sin(\alpha)$

$z_n = 0$

$$\tan(\alpha) = \frac{y_n}{x_n}$$

[3]

d) In order to evaluate $\cos(\alpha)$, we need to set $z = \alpha$, and iterate with $d_i = \text{sign}(z_i)$

$z_0 = 18,4350$, $d_0 = +1$

iteration.

1. $z_1 = 18,4350 - 45,000 = -26,565$, $d_1 = -1$

2. $z_2 = -26,562 + 26,565 = 0$

We need two iterations.

The given architecture performs one iteration per clock cycle, so we need two clock cycles.

[3]

2. [Bookwork]

3

a)

- PLL aligns the rising edge of reference input clock to feedback clock using PFD.
- PFD detects difference in phase and freq between reference clock and feedback clock and generates "up" and "down" control signals based on whether the feedback freq. is lagging or leading the ref. freq.
- The loop filter converts the "up" and "down" signals to control the oscillation freq of the VCO
- feedback loop counter is used to increase VCO freq above input ref. freq.
- Pre-scale counter (N) is used to produce the ref freq from Fin
- The post-scale counter (C) allow to generate other freq. from Fvco.

[5]

b)

DLL	PLL
· first order loop	· second/third order loop
· freq. synthesis is diff.	· stability can be an issue
· ref clock jitter passed to outp.	· phase error accumulation
· no phase error accum.	

[5]

c) $f_{REF} = f_{IN}/N$

$f_{VCO} = f_{REF} \times M = f_{IN} \times M/N$

$f_{OUT} = f_{VCO}/C = f_{IN} \times M/(N \times C)$ [5]

d) [redacted]. [5]

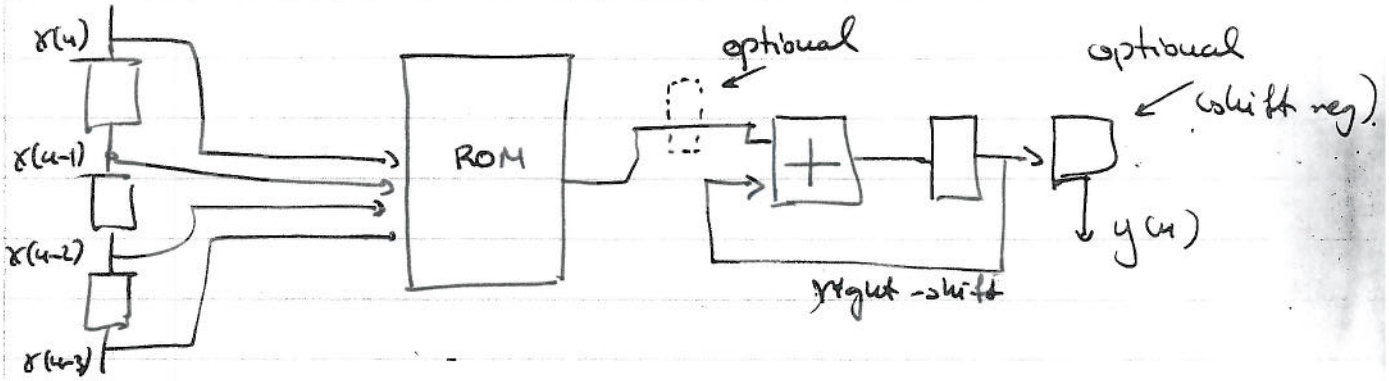
F_{VCO} should be 150MHz.

$N = 1$, $M = 5$, $C = 2$.

5

3. [Tests student's ability to design a system based on Distributed arithmetic]

a)



[12]

c) x has 8 bits. So, 1 result per 8 clock cycles.

[1]

d)

$x^i(n)$	$x^i(n-1)$	$x^i(n-2)$	$x^i(n-3)$	Value
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	3
0	1	0	1	4
0	1	1	0	5
0	1	1	1	6
1	0	0	0	5
1	0	0	1	6
1	0	1	0	7
1	0	1	1	8
1	1	0	0	9
1	1	0	1	9
1	1	1	0	10
1	1	1	1	11

size
16 x 4

[2]

b@) 4-bit adder

[2]

4. [Tests students' ability to map an FSM to an FPGA using LUTs]

a)

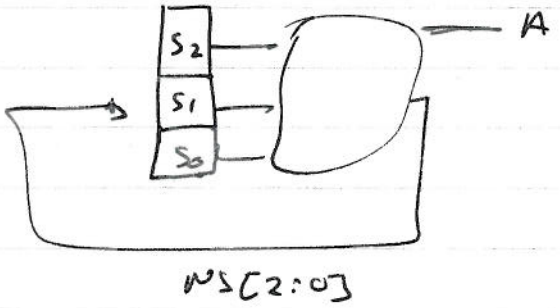
S_2	S_1	S_0	NS_2	NS_1	NS_0	A
0	0	0	0	0	1	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	1	1	1	0
1	1	1	0	0	0	1

$$NS_0 = \bar{S_2}\bar{S_1}\bar{S_0} + \bar{S_2}S_1\bar{S_0} + S_2\bar{S_1}\bar{S_0} + S_2S_1\bar{S_0}$$

$$NS_1 = \bar{S_2}\bar{S_1}S_0 + \bar{S_2}S_1\bar{S_0} + S_2\bar{S_1}S_0 + S_2S_1\bar{S_0}$$

$$NS_2 = \bar{S_2}S_1S_0 + S_2\bar{S_1}\bar{S_0} + S_2\bar{S_1}S_0 + S_2S_1\bar{S_0}$$

$$A = S_2S_1S_0$$



[20]

b) A 4-LUT implements a 4-input boolean function.

$NS_0 \rightarrow 4 + 1 = 5$ LUTs
$NS_1 \rightarrow 4 + 1 = 5$ LUTs
$NS_2 \rightarrow 4 + 1 = 5$ LUTs
$A \rightarrow 1$ LUT.

Total: 16 LUTs. [5]

c) $t_p + t_{Logic} + t_s < T \Rightarrow T = 3 + 2 \cdot 2 + 2 = 9$ use

↑
2 levels of LUTs

$t_p + t_p > t_h \Rightarrow$ always holds.

Max freq: $\frac{1}{9\,use}$

[3]

d) No, because of the feedback loop.

[2]

5. [Tests students' ability to handle timing + Bookwork]

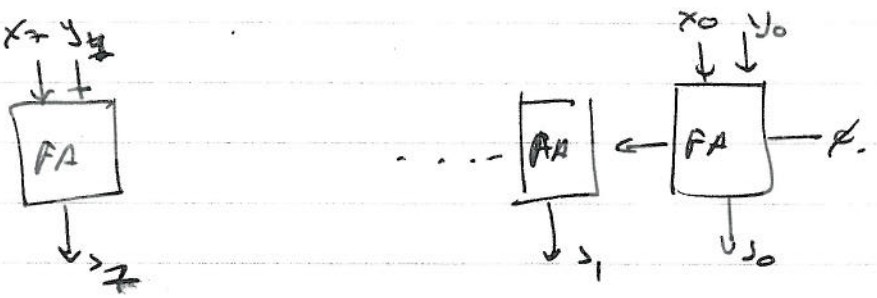
a)

A 4-LUT implements any boolean function of 4 inputs.

So, for one FA, 2 LUTs are needed.

[3]

b)



[3]

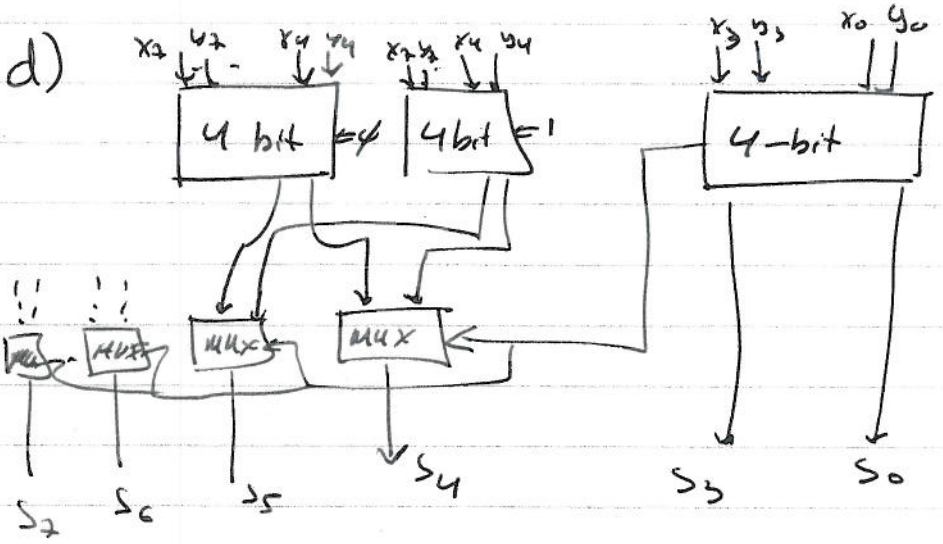
c) The critical path is from $x_0, y_0 \rightarrow s_7$

The logic goes through 8 LUTs $\Rightarrow$ 16 nsec

$$\text{max freq} = \frac{1}{16 \text{ nsec}}$$

[4]

d)



[3]

e) Each 4-bit req. 8 LUTs.

Also 4 2-to-1 muxes are needed.

Thus $3 \times 8 + 4 \times 1 = 28$ LUTs.

[3]

f) Critical path is through 4-bit ripple carry + mux

So $4 \times 2 + 1 \times 2 = 10$ nsec.

max freq: $\frac{1}{10 \text{ nsec}}$

[4]

6. [Tests student's ability on timing]

a) The critical path is between the output of the first register and the output register.

$$T_{cp}^{reg} + T_p^{mul} + 2 \cdot T_p^{add} + T_{set}^{reg} < T \Rightarrow$$

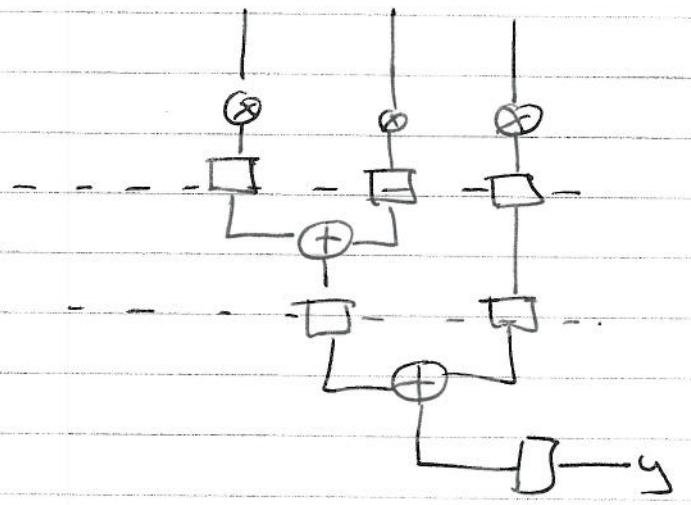
$$T > 5 + 10 + 2 \cdot 8 + 2 = 33 \text{ nsec}$$

Hold ineq: $T_{cp}^{reg} + t_p^{mul} + 2T_p^{add} < t_h$ ok

Max freq: $1 / 33 \text{ nsec}$

[10]

b) By adding two pipeline register stages



Throughput is 1 result per clock cycle,
Latency: 3 clock cycles

[7]

c) The first register may sample the signal when it changes value. (a change within setup-hold window)
This creates an oscilation to the output of the first ~~flip-flop~~ register.

12/12

$$MTBF(t_r) = \frac{\exp\left(\frac{t_r}{\tau}\right)}{T_0 \cdot f \cdot \alpha}$$

$$t_r = \frac{1}{40 \cdot 10^6} - t_{sett}^{neg} = 25 \cdot 10^{-3} \cdot 10^{-6} - 2 \cdot 10^{-9}$$
$$= 23 \cdot 10^{-9}$$

$$MTBF = \frac{e^{\frac{23 \times 10^{-9}}{0.5 \times 10^{-9}}}}{9.5 \times 10^{12} \cdot 40 \cdot 10^6 \cdot 40} =$$

$$= \frac{e^{46}}{15{,}200 \times 10^{18}} = \frac{9{,}49 \cdot 10^{19}}{15\,200\ 10^{18}} = 6{,}24 \times 10^{-3}$$

$\Rightarrow$ MTBF = 6,24 msec.