Theory of Logic Circuits

Laboratory manual

Exercise 7

Dynamics of switching circuits

1. Introduction

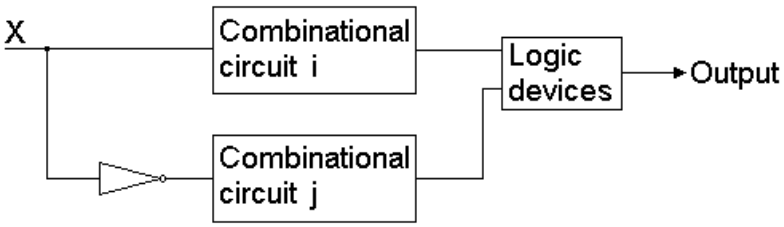
While thinking about real switching devices one has to take into consideration that unlike ideal circuits they posses the property of having two states: stable and unstable. What's more, all logical elements such as gates, flip-flops, require some time (called propagation time) to change its output state when appropriate change of the value on the input occurs. These properties, along with the fact that actually only one signal can change its value at a time (instantaneous and simultaneous change of multiple signals is not possible – they always change one after another, even if the difference in time is insignificant), cause some momentary errors on the output of a circuit or even result in its malfunction.

These problems with dynamic properties of switching circuits can be divided into three categories:

- hazards occurring in combinational circuits (or in a combinational part of a sequential circuit),
- races in sequential circuits,
- essential hazards in sequential circuits.

2. Hazards in combinational circuits

Hazards in combinational circuits occur as momentary errors on the output and are the result of delay time introduced by the use of some gates in implementation of a function, as shown in the picture below.

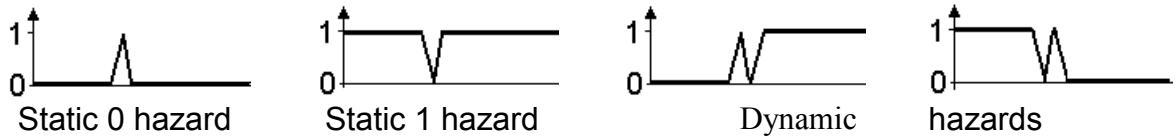


There are two main types of hazards: static and dynamic.

Static hazard occurs when the state on the output should stay at some level but for a moment it changes its value. When the output should be "1" but for a moment it goes to 0 (1→0→1) we call it a static 1 hazard HS1. When the output should be "0" but for a moment it goes to "1" (0→1→0) we name it a static 0 hazard HS0.

Dynamic hazard occurs when the state on the output is supposed to change the value once, but the change is multiplied (1→[0→1→]0 or 0→[1→0→]1).

The timing charts for all these cases are as follows.



Circuits are usually given in form of their logical diagrams. When analysing a circuit with respect to hazards, the first step is to obtain the logical expression corresponding to its diagram. In this process it is essential to remember that the theorems such as $x+\bar{x}=1$ and $x\cdot\bar{x}=0$ are no longer

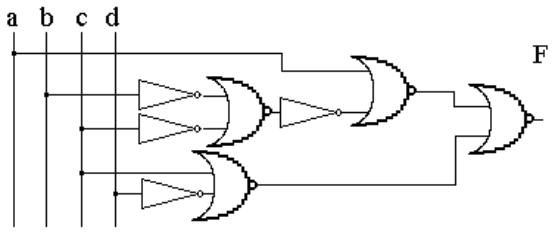
valid (they are true only for ideal elements), because for the propagation time of NOT gate in the first case we get zero and in the second case one when signal x changes its value.
That in turn leads to other theorems that need modification.

Instead of	we must use
$x+\overline{x}\cdot y = x+y$	$x+\overline{x}\cdot y = (x+\overline{x}+\overline{y})\cdot(x+y)$
$x\cdot(\overline{x}+y) = x\cdot y$	$x\cdot(\overline{x}+y) = x\cdot\overline{x}\cdot\overline{y}+x\cdot y$
$x\cdot z+\overline{x}\cdot y = (x+y)\cdot(z+\overline{x})\cdot(z+y)$	$x\cdot z+\overline{x}\cdot y = (x+\overline{x}+\overline{y}+\overline{z})\cdot(x+y)\cdot(z+\overline{x})\cdot(z+y)$
$(x+z)\cdot(\overline{x}+y) = x\cdot y+z\cdot\overline{x}+z\cdot y$	$(x+z)\cdot(\overline{x}+y) = x\cdot\overline{x}\cdot\overline{y}\cdot\overline{z}+x\cdot y+z\cdot\overline{x}+z\cdot y$

Once we got the function in either Sum of Product terms form or Product of Sum terms form, we should establish whether this is a normal form of a function.
A function is called normal when there are no terms that would contain the same variable (a literal) in both uncomplemented and complemented forms (for example $x_1+\overline{x_1}+\ldots$ or $x_1\cdot\overline{x_1}\cdot\ldots$).
Such terms are the result of using the modified theorems in transformations. They cause problems because they cannot be interpreted as other terms as a group of 1s or 0s in a Karnaugh map. There is no such a cell in a Karnaugh map that would correspond to $x_1+\overline{x_1}+\ldots$ or $x_1\cdot\overline{x_1}\cdot\ldots$ and we have to analyse their meaning in another way.
When the function is normal, there is only one type of hazard that may occur and for SoP form it is a static 1 hazard, for PoS form it is a static 0 hazard. Static hazards are very easy to perceive in the Karnaugh map of a function so we simply need to draw it and fill it in accordingly to the logical expression, carefully recreating groups of 1s or 0s. Once the map is completed, we look through it for any logically adjacent 1s (for SoP form) or 0s (for PoS form) that are not covered by any common group. Each such a pair found corresponds to a condition for a static hazard. When transition from one to another happens as a result of the input change then a hazard occurs. To prevent it from happening we have to add some extra groups covering all such pairs. While creating these groups we should remember about the general rules for grouping, especially that groups should be of a maximum possible size.
When all adjacent 1s (or 0s) are covered by at least one common group, the function is hazard-free (i.e. free of the hazard corresponding to the form of a function).

2.1. Example

Analyse the function given by a logical diagram with respect to hazards. Propose a solution for all occurring problems.



First we obtain the logical expression for the function given.

$$F = \overline{a+\overline{\overline{b}+\overline{c}}+\overline{c+\overline{d}}}$$

Then we transform it, using DeMorgan's theorem.

$$F = (a+\overline{b}+\overline{c})\cdot(c+\overline{d})$$

What we get here, is a PoS form of the function (it's better not to change it into SoP - then we would have to use the modified theorems) so we don't need any more transformations. What's more, it is a normal form of the function as in none of terms there is a variable in both complemented and uncomplemented form. As stated before, it means that only one type of

hazard may occur and (the others would be the result of the term $x_1 + \overline{x_1} + \ldots$ if the function were not normal) for PoS form it is static 0 hazard. We draw the Karnaugh map for the function and recreate groups of 0s.

cd / ab	00	01	11	10
00		0		
01		0	0	0
11		0		
10		0		

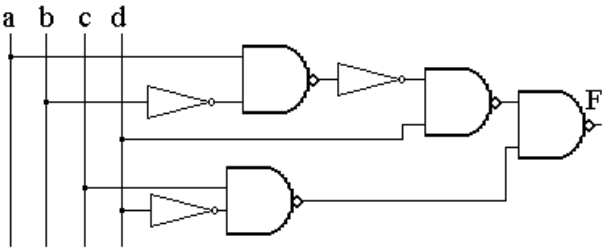
Now we look for all pairs of adjacent 0s that are not in any common group and we find that there is one such a pair, so when transition between the states $(0101 \leftrightarrow 0111)_{abcd}$ happens, a static 0 hazard occurs. To prevent it we have to add one extra group covering this pair. The biggest possible group to create is $(a + \overline{b} + \overline{d})$ so hazard-free function is as follows.

$F = (a + \overline{b} + \overline{c}) \cdot (c + \overline{d}) \cdot \underline{(a + \overline{b} + \overline{d})} \leftarrow$ anti-hazard group

At this stage our task is completed.

2.2. Example

Analyse the function given by a logical diagram with respect to hazards. Propose a solution for all occurring problems.



First we obtain the logical expression for the function given.

$$F = \overline{\overline{a \cdot \overline{b} \cdot d} \cdot \overline{c \cdot \overline{d}}}$$

Using DeMorgan's theorem we get the following.

$$F = a \cdot \overline{b} \cdot d + c \cdot \overline{d}$$

This is a SoP function (again it's better not to change it into PoS - then we would have to use the modified theorems). What's more, it is in its normal form (it doesn't have any $x_1 \cdot \overline{x_1} \cdot \ldots$ terms) so it may have only some HS1 (the other types of hazards for SoP function happen when it is not in a normal form and are the result of the term $x_1 \cdot \overline{x_1} \cdot \ldots$.). As before we create the Karnaugh map and fill it in.

ab / cd	00	01	11	10
00				1
01				1
11				1
10		1	1	1

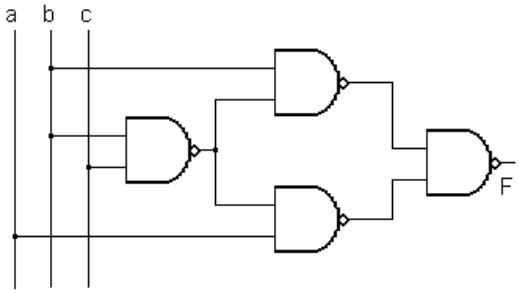
Then we check if all 1s that are logically adjacent are in at least one common group. If so, no hazards can happen. Otherwise we have some static 1 hazards for each such a pair.
For our function we have one such a pair here and that means that when the function reacts to the change in the input states between $(1011) \leftrightarrow (1010)_{abcd}$ the output instead of staying at 1 for a moment goes to zero. To prevent this we have to add an extra group covering this pair of 1s and it is $a \cdot \overline{b} \cdot c$, so the hazard-free function is as follows:

$F = a \cdot \bar{b} \cdot d + c \cdot \bar{d} + \underline{a \cdot \bar{b} \cdot c}$ ← anti-hazard group
Then our task is completed.

The case is more interesting when the function we get is not in its normal form, which will be shown in the two following examples.

2.3. Example

Analyse the function given by a logical diagram with respect to hazards. Propose a solution for all occurring problems.



First we obtain the logical expression for the function given.
$F = \overline{\overline{b \cdot \bar{b} \cdot \bar{c}} \cdot \overline{a \cdot \bar{b} \cdot c}}$
After applying DeMorgan's theorem, we got:
$F = b \cdot \overline{\bar{b} \cdot \bar{c}} + a \cdot \overline{\bar{b} \cdot \bar{c}} = b \cdot (\bar{b} + \bar{c}) + a \cdot (\bar{b} + \bar{c})$
Here we need to use the modified theorem and as a result of our transformations we get the function that is not in a normal form.
$F = b \cdot (\bar{b} + \bar{c}) + a \cdot (\bar{b} + \bar{c}) = a \cdot \bar{b} + a \cdot \bar{c} + b \cdot \bar{c} + b \cdot \bar{b} \cdot c$
The first three expressions we can reconstruct as normal groups of 1s in the Karnaugh map.

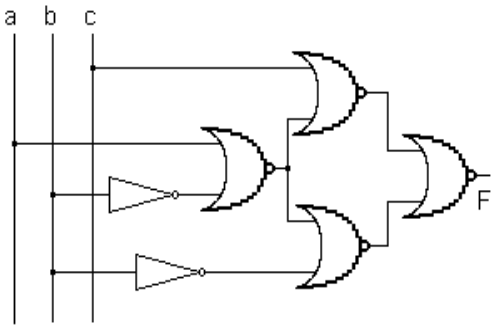
a \ bc	00	01	11	10
0				1
1	1	1		1

The term $b \cdot \bar{b} \cdot c$, however, is not that easy to deal with and we have to interpret it in another way. First, it is necessary to establish what it actually is - it is something that should be always zero, but it isn't. It is a momentary 1 when the b variable changes its value. As this momentary 1 is multiplied by c, the whole term equals to 1 when b changes and c is 1. So what we need to do is to find all these cases. We look in the Karnaugh map for such cells where c is 1 and between which b changes its value.
For transitions between such cells we draw a thick line to indicate this hidden 1 and then we consider where it happens. When two cells contain zeros, during transition from one to another the output of the function is supposed to stay at zero. For the case of going between the states $(001 \leftrightarrow 011)_{abc}$ on the way there is the thick line indicating that momentary one. So what happens is an error and such an error is called static 0 hazard. When one cell contains 0 and another 1, and transition between them has on its way the thick line such as in the case of the states $(101 \leftrightarrow 111)_{abc}$, then what we actually get is a change $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ or $0 \rightarrow 1 \rightarrow 0 \rightarrow 1$ and such glitches are called a dynamic hazard.
As to checking for the static 1 hazard, we have as usual to ensure that all pairs of adjacent 1s are in at least one common group.
In the example given we don't have any such a pair, which means that the function is free from static 1 hazard.
The next question is how to solve the problems i.e. how to remove HS0 and HD from the function. The answer is very easy - we need to leave out the term $x \cdot \bar{x}$. Then we obtain the function that is logically equivalent to the given in implementation, but hazard-free. $F = a \cdot \bar{b} + a \cdot \bar{c} + b \cdot \bar{c}$

Then the task is completed.

2.4. Example

Analyse the function given by a logical diagram with respect to hazards. Propose a solution for all occurring problems.



First we obtain the logical expression for the function given.

$$F = \overline{\overline{c + a + \overline{b}} + \overline{\overline{b} + a + \overline{b}}}$$

Once more we apply DeMorgan's theorem, which results in the following expression:

$$F = (c + \overline{a + \overline{b}}) \cdot (\overline{b} + \overline{a + \overline{b}}) = (c + \overline{a} \cdot b) \cdot (\overline{b} + \overline{a} \cdot b)$$

Here we need to use the modified theorem and as a result of our transformations we get the function that is not in its normal form.

$$F = (c + \overline{a} \cdot b) \cdot (\overline{b} + \overline{a} \cdot b) = (c + \overline{a}) \cdot (c + b) \cdot (\overline{b} + \overline{a}) \cdot (\overline{b} + b + a)$$

The first three expressions we can reconstruct as groups of 0s in the Karnaugh map.

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

The term $(b + \overline{b} + a)$ must be interpreted in another way. Again it is necessary to establish what it is - it is an expression that should be always one, but it isn't. It is a momentary 0 when the b variable changes its value. As this momentary 0 is added to a, the whole term equals to 0 when b changes and a is 0. So what we have to do is to find all these cases. We look in the Karnaugh map for such cells where a is 0 and between which b changes its value.
For transitions between such cells we draw a thick line to indicate this hidden 0 and then we consider where it happens. When two cells contain ones, during transition from one to another the output of the function is supposed to stay at one. For the case of going between the states $(001 \leftrightarrow 011)_{abc}$ on the way there is the thick line indicating that momentary zero. So what happens is an error and such an error is called static 1 hazard. When one cell contains 0 and another 1, and transition between them has on its way the thick line such as in the case of the states $(000 \leftrightarrow 010)_{abc}$, then what we actually get is a change $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ or $0 \rightarrow 1 \rightarrow 0 \rightarrow 1$ and as you have already learnt such glitches are called a dynamic hazard.
As for checking for the static 0 hazard, we have as usual to ensure that all pairs of adjacent 0s are in at least one common group.
In the example given we don't have any such a pair, which means that the function is free from static 0 hazard.
The next question is how to solve the problems i.e. how to remove HS1 and HD from the function. Again it is very easy - we need to leave out the term $x + \overline{x}$. Then we obtain the function that is logically equivalent to the given in implementation, but hazard-free. $F = (c + \overline{a}) \cdot (c + b) \cdot (\overline{b} + \overline{a})$
Then the task is completed.

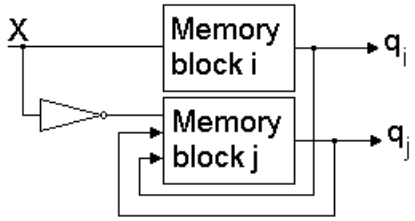
There is one thing that requires some special emphasis. Most people prefer to deal with SoP form instead of PoS form of a function. In the case like in Example 4, however, it is rather tricky, because to transform it into SoP form one must remember all modified theorems (more sophisticated that the ones used here) and usually it ends with mistakes. This, in turn, results in obtaining the function that has the dynamic properties different to the function given.

The another commonly made mistake is that when given a function in SoP form and asked about a static 0 hazard, as an answer the groups of zeros are created, then checked whether there are any unconnected adjacent ones. This approach is **totally wrong**, because while creating groups actually the new function is created instead of analysing the one that is given. As stated above, for SoP function the other hazards than a static 1 hazard can occur if and only if the form is not normal because of a term $x \cdot \bar{x}$.
Respectively, when given a function in PoS form and asked about a static 1 hazard, it is wrong to create groups of 1s in order to check for any unconnected adjacent 1s. Through creating groups the new function with new properties is created instead of analysing the one given. For PoS function the other hazards than a static 0 hazard can occur if and only if the form is not normal because of a term $x + \bar{x}$.

As in combinational circuits hazards are the reason of only some momentary errors on the output, for the whole circuit it usually doesn't matter. The different case is, however, for a combinational part of a sequential circuit. In a sequential circuit this momentary error may influence other elements and result even in performing of an action contradictory to the circuit programme. And that is why, while implementing a sequential circuit with feedback loops, it is essential to create the logical expressions describing the circuit ensuring that these expressions are hazard-free. The methods and rules for creating input excitation functions for implementation of sequential circuits with flip-flops are described in material concerning Exercise 3 (Basic sequential switching circuits) and 4 (Asynchronous sequential switching circuits).

3. Races in sequential circuits

Races in sequential switching circuits occur when the circuit programme requires the transition between two states, the codes of which are not logically adjacent – i.e. they differ in more than one bit. It means that a race may happen only when for describing internal states there is needed more than one state variable.
Races occur either as momentary or permanent errors in the state variables values and are the result of a delay time caused by different paths through which a signal and its complementation control at least two memory blocks (devices), as shown (in the simplified form) in the picture below.

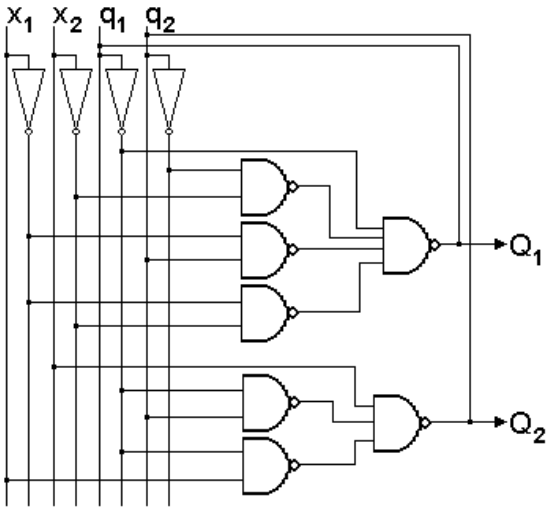


When a race results only in some momentary errors, but a circuit eventually reaches an expected state, such a race is called non-critical (NCR).
If a race causes permanent errors – i.e. a circuit does not react accordingly to its programme and goes to the wrong state, such a race is called critical (CR).

The third type of race, so-called controlled race (CTR), often offers a solution for encountered problems. The controlled race happens when a designer consciously changes some unstable states in such a way that the possibility of requiring not adjacent transition is avoided.
As no-critical race results only in momentary errors, then when a circuit works independently and these errors cannot influence the behaviour of any other elements, it usually is left as it is.
The problem of critical race, however, has to be solved always if we want to ensure the correct work of the whole circuit.
In general, the ways of avoiding the critical race are two (but not always both are possible to use): the change of encoding of internal states and introducing a controlled race.
The special case of the change of encoding is introducing an additional state variable when the solution for the given number of variables is not possible to achieve.
All these cases will be shown in examples.

3.1. Example

Analyse the circuit given by a logical diagram with respect to races. Propose a solution for the critical-race-free circuit.



The first step in such analysis is to read the diagram and change it into logical expressions describing the feedback loops.

$$Q_1 = \overline{\overline{q_1 \cdot q_2} \cdot \overline{x_2 \cdot q_2} \cdot \overline{x_1 \cdot x_2}}$$

$$Q_2 = \overline{x_2 \cdot \overline{q_1 \cdot q_2} \cdot \overline{x_1 \cdot q_1}}$$

After applying DeMorgan's theorem we receive the following:

$$Q_1 = q_1 + \overline{q_2} \cdot \overline{x_2} + q_2 \cdot \overline{x_1} + \overline{x_1} \cdot \overline{x_2}$$

$$Q_2 = \overline{x_2} + \overline{q_1} \cdot q_2 + x_1 \cdot \overline{q_1}$$

Then we recreate the transition maps for Q_1 and Q_2, and a composite map.

In the process we should notice that for Q_1 the group $\overline{x_1} \cdot \overline{x_2}$ is an anti-hazard group.

Q_1

q_1q_2 \ x_1x_2	00	01	11	10
00	1	0	0	1
01	1	1	0	0
11	1	1	1	1
10	1	1	1	1

Q_2

q_1q_2 \ x_1x_2	00	01	11	10
00	1	0	1	1
01	1	1	1	1
11	1	0	0	1
10	1	0	0	1

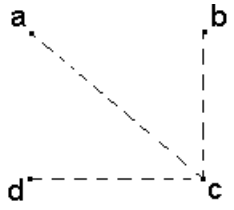
$Q_1\ Q_2$

q_1q_2 \ x_1x_2	00	01	11	10
00	11	00	01	11
01	11	11	01	01
11	11	10	10	11
10	11	10	10	11

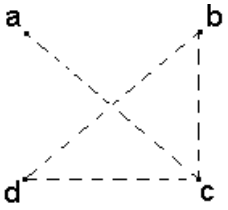
Next, there is a time for the actual analysis. To make it easier, we may encircle the values corresponding to stable states.

q_1q_2 \ x_1x_2	00	01	11	10	
00	11	(00)	01	11	
01	11	11	(01)	(01)	
11	(11)	10	10	(11)	
10	11	(10)	(10)	11	$Q_1\ Q_2$
	NCR	CTR		CR	

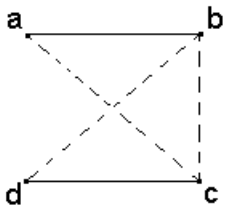
The map is analysed column after column as follows.
For $x_1x_2 = 00$, there is only one stable state, which means that a critical race cannot occur (for this at least two stable states are needed). All unstable states here require the state variables to change to "11", so even from not logically adjacent state of "00" the circuit will reach the proper state, no matter which state variable wins the race. Such situation is called a non-critical race, and as we need only critical-race-free solution, we may leave it as it is.
In the second column, where $x_1x_2 = 01$, we have two unstable states, both with logically adjacent transitions, and what's more, one leads to the second while the second ends in a stable state. Such a case is named a controlled race because it offers the possibility of safe transition from the current state $q_1q_2 = 01$ into "10" that is not logically adjacent, avoiding a critical race, which would happen otherwise.
The column of x_1x_2 equal to "11" shows only logically adjacent transitions.
The last column of $x_1x_2 = 10$ contains two stable states and requires a transition from the current state $q_1q_2 = 00$ to the state "11", which means that both variables should change at the same moment. As this is not possible and if q_2 reacts faster, then a circuit will reach the incorrect state "01". This situation is a critical race that must be avoided.
To choose the way to solve the problem it is best to draw a transition diagram that will show which states should, which have to be logically adjacent in order to obtain a critical-race-free solution for the given number of the state variables.
As we would not consider the current encoding of states, we should assign to each row of the map some symbols, for example $q_1q_2 = 00 \equiv a$, $q_1q_2 = 01 \equiv b$, $q_1q_2 = 11 \equiv c$, $q_1q_2 = 10 \equiv d$.
In this diagram we draw a continuous line if without logically adjacent encoding of two states connected by the line (for the given number of state variables) there may occur a critical race. For possible non-critical and controlled races we use a dashed line.



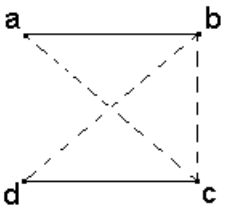
After analysing the 1st column



After analysing the 1st and 2nd column



After analysing the 1st, 2nd and 3rd column



The final diagram

First column contains the non-critical race, so each of other states should be connected by dashed lines with c state.

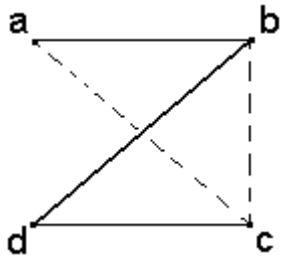
The second column introduces the controlled race, which requires the connection of states b and c with d state, both with a dashed line. The connection between c and d is already done, so we only add the one for b and d.
The third column requires the logical adjacency of a and b, and c with d. If we changed the current encoding without keeping this adjacency, we would cause a critical race. That is why we need to use a continuous line to indicate this. As c and d are already connected with the dashed line, we have to exchange it with continuous that prevails.
The last column contains the transition to the state c but there are two unstable states in this column. That offers us a possibility of introducing a controlled race, so we should have connections between b and d with c by a dashed line, despite the fact that currently there is the critical race. As all these connections are already done in the previous stages of the analysis process, the diagram is finished.
Now we may look back at the current encoding and check whether with keeping it (this is usually the fastest way) we are able to avoid critical races. For the example given it is possible as all states connected by continuous lines are logically adjacent. The only action actually needed is introducing a controlled race in the last column to prevent the critical race. So the critical-race-free solution, presented in the form of a map, looks as follows.
As the solution is supposed to be only critical-race-free and not race-free, we leave the non-critical race as it is.

	q_1q_2 \ x_1x_2	00	01	11	10	
a	00	11	(00)	01	10	
b	01	11	11	(01)	(01)	
c	11	(11)	10	10	(11)	
d	10	11	(10)	(10)	11	$Q_1\ Q_2$
		NCR	CTR		CTR	

3.2. Example

Analyse the circuit given by a composite map below with respect to races. Propose a solution for the race-free circuit.

	q_1q_2 \ x_1x_2	00	01	11	10	
a	00	11	(00)	01	11	
b	01	11	10	(01)	(01)	
c	11	(11)	(11)	10	(11)	
d	10	11	(10)	(10)	11	$Q_1\ Q_2$
		NCR	CR		CR	



The first column contains a non-critical race, the second and the last critical races.
In order to avoid critical races we have to keep all states connected with continuous lines in the created transition diagram as logically adjacent. It means that we have to change encoding – currently the states b and d differ on both bits. There are several ways of doing the new encoding, but usually we choose the one that requires the fewest possible changes. Then we should simply exchange codes between c and d.

The map looks as follows.

	q_1q_2	x_1x_2 00	01	11	10	
a	00	10	(00)	01	10	
b	01	10	11	(01)	(01)	
d	11	10	(11)	(11)	10	
c	10	(10)	(10)	11	(10)	$Q_1\ Q_2$
		NCR				

In the first column there still is a non-critical race and as this time the solution is supposed to be race-free, we should introduce a controlled race and there are two following possibilities.

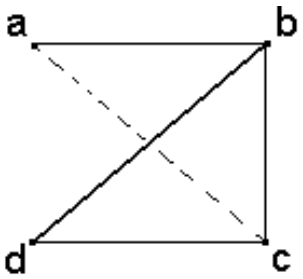
	q_1q_2	x_1x_2 00	01	11	10	
a	00	10	(00)	01	10	
b	01	00	11	(01)	(01)	
d	11	10	(11)	(11)	10	
c	10	(10)	(10)	11	(10)	$Q_1\ Q_2$
		CT R				

	q_1q_2	x_1x_2 00	01	11	10	
a	00	10	(00)	01	10	
b	01	11	11	(01)	(01)	
d	11	10	(11)	(11)	10	
c	10	(10)	(10)	11	(10)	$Q_1\ Q_2$
		CT R				

3.3. Example

Analyse the circuit given by a composite map below with respect to races. Propose a solution for the race-free circuit.

	q_1q_2	x_1x_2 00	01	11	10	
a	00	(00)	11	(00)	(00)	
b	01	00	(01)	10	11	
c	11	10	(11)	(11)	(11)	
d	10	(10)	11	(10)	(10)	$Q_1\ Q_2$
			CR	CR		



The two middle columns have critical races. The transition diagram shows that to avoid critical races the state b should be logically adjacent with all other states. For encoding by two bits it is not possible to achieve, so we need to add one more state variable. One of the possible ways of encoding is as follows:

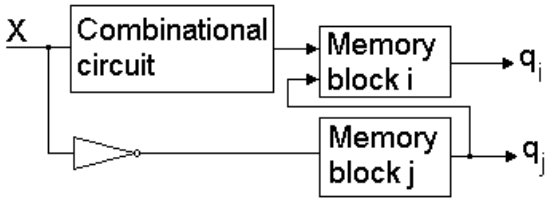
State	$q_1q_2q_3$
a	000
b	001
c	011
d	101

We also have to introduce some controlled races. Again, one of the possible solutions shows the map below.

	$q_1q_2q_3$	x_1x_2 00	01	11	10
a	000	(000)	010	(000)	(000)
b	001	000	(001)	101	011
c	011	111	(011)	(011)	(011)
	010		011		
	110				
	111	101	011		
d	101	(101)	111	(101)	(101)
	100				
		CTR	CTR		$Q_1\,Q_2\,Q_3$

4. Essential hazards in sequential circuits

Essential hazards are a result of unequal signal delay paths that exist in a circuit and they cause the transition to the stable state unpredicted by the circuit programme. They may occur when a signal and its complementation drive a memory block (device) by such paths that one leads through another memory block while the other only a combinational circuit, as shown in the picture below.



The correct work of a circuit is that when there is a stable state, the circuit stays in it until some change on the inputs occurs. When such a change happens, the circuit should “notice” it by changing first the column of a map, then, according to the state it encounters, if it is an unstable state the circuit should proceed with transition to the appropriate stable state – i.e. change the row in the map.
Sometimes, however, due to these different paths in a circuit, one memory block (device) may react faster than the others, which results in a change of a current state (i.e. the row within the column). If on the way there are some unstable states that cause further changes, then when eventually the other memory block “sees” the change on the input, transition leads to the wrong state and this is essential hazard. The example of such a case is shown below.

4.1. Example

Analyse the circuit given by a composite map below with respect to essential hazards.

q_1q_2 \ x_1x_2	00	01	11	10
00	00	00	10	-
01	01	00	01	11
11	10	11	01	11
10	10	10	10	10

$Q_1\ Q_2$

As with analysing races first it is best to mark stable states, but also instead of using binary codes to refer to states we will assign them some decimal numbers.

q_1q_2 \ x_1x_2	00	01	11	10
00	(1)	(4)	8	-
01	(2)	4	(7)	9
11	3	(5)	7	(9)
10	(3)	(6)	(8)	(10)

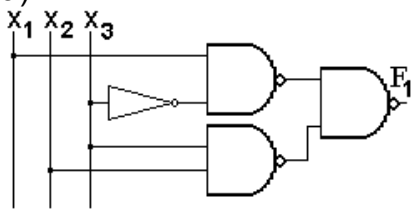
$Q_1\ Q_2$

Then the analysis with respect to essential hazards is as follows: we have to check the transition from each stable state when the input change occurs. If, by proceeding within the column first (which corresponds to the reaction in one memory block) and then in the row the circuit achieves the proper (i.e. predicted by the programme) state there is no problem. Otherwise, we have essential hazard.
For the state (1) the answer for the change on x_1 is unspecified (don't care condition). When x_2 changes, the values of state variables should be the same, so no problems may occur.
For (2) state when x_1 becomes 1, the transition is supposed to go to the (9) state, which corresponds to the change of the current state q_1q_2=01 into q_1q_2=11. If q_1 reacts before q_2, then the circuit proceeds through the unstable state 3 to the stable one, then the answer for the input change results in reaching the wrong state of (10) .
For (3) state the transition to (10) state due to x_1=1 has to be correct as not requiring the change of the state variables as well as the reaction to x_2=1.
The analysis of all other states and transitions results in the statement that for the circuit given there are three essential hazards. When going from the state (2) to (9) (as described above), the circuit may wrongly end in the (10) state; while proceeding from the state (5) to (7) the wrong state (8) may be achieved and from (7) state when the transition to (4) is required it incorrectly reaches (6) state.
Essential hazards cannot be fixed by simply adding additional terms in the next state output or excitation input equations. To fix an essential hazard one must first determine that such a hazard can exist. Theoretical analysis gives only hints where problems may occur but the answer whether they occur at all is given only by implementation on real elements. If such a hazard exists, it is necessary to ensure that the feedback delays for the state variables are adequate so that the essential hazard does not occur.

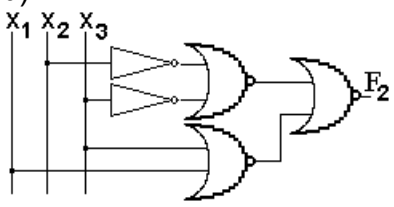
5. Tasks to be performed during laboratory

1. Analyse the circuits given with respect to hazards.

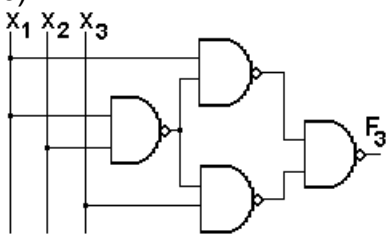
a)



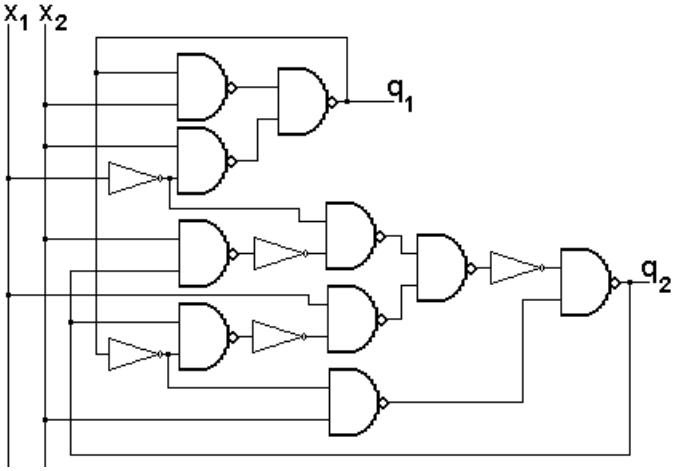
b)



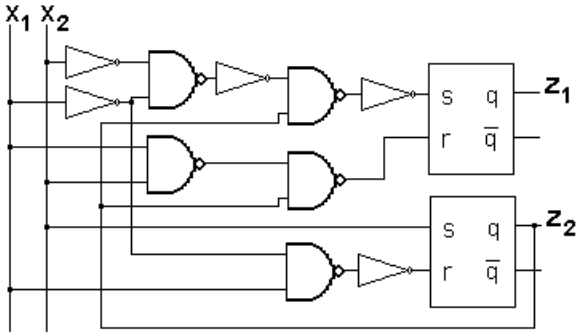
c)



2. Analyse the circuit given with respect to races.



3. Analyse the circuits given with respect to essential hazard.



6. Instructions to follow

1. Solve all tasks before the exercise.
2. Implement all circuits (using given elements) and compare their behaviour to the result of the theoretical analysis.
3. Perform experiments with several delay times for different parts of circuits observing the reaction of each circuit.