HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts



Image by Qwen-Image 2.1

Algorithmic minimization of functions with many inputs and hazard analysis

DET-04 · Digital Electronics

GUIDING QUESTIONS

- Can a computer minimize a circuit for you – and how does it go about it?
- Why does a cheap robot vacuum drop its Wi-Fi connection in the middle of the night?
- Six signals decide whether a heat pump may start: what do you do when a Karnaugh map becomes too large?



Image by Qwen-Image 2.1

INTENDED LEARNING OUTCOMES

- You minimize simple functions algorithmically and explain how the algorithm deals with functions with many inputs.
- You detect hazards and eliminate those that are caused by the change of a single input.
- You distinguish combinational logic circuits from sequential circuits.



Image by Qwen-Image 2.1

AGENDA

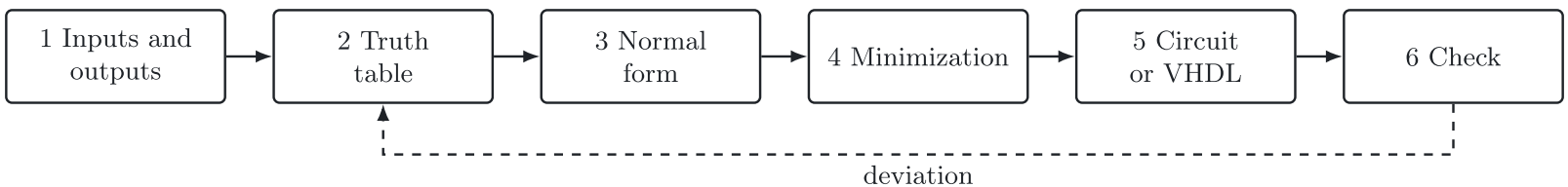
1. Prime implicants
2. The Quine–McCluskey algorithm
3. Functions with many inputs
4. Hazards
5. Combinational logic circuits and sequential circuits
6. What have you learnt?



Image by Qwen-Image 2.1

REVIEW OF DET-03

Synthesis: six steps from the description to the checked circuit



- Step 3, normal form: the OR of the minterms of all ones, the AND of the maxterms of all zeros
- Step 4, minimization: Boolean algebra and the Karnaugh map; don't cares enlarge the loops
- Step 6, check: the same truth table or the same minterm list means the same job

Today: step 4 by table and for many inputs – and short pulses that the check in step 6 does not see

HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts

1 Prime implicants

Step 4 of the synthesis (DET-03), the minimization: which loops does a minimal expression really need?

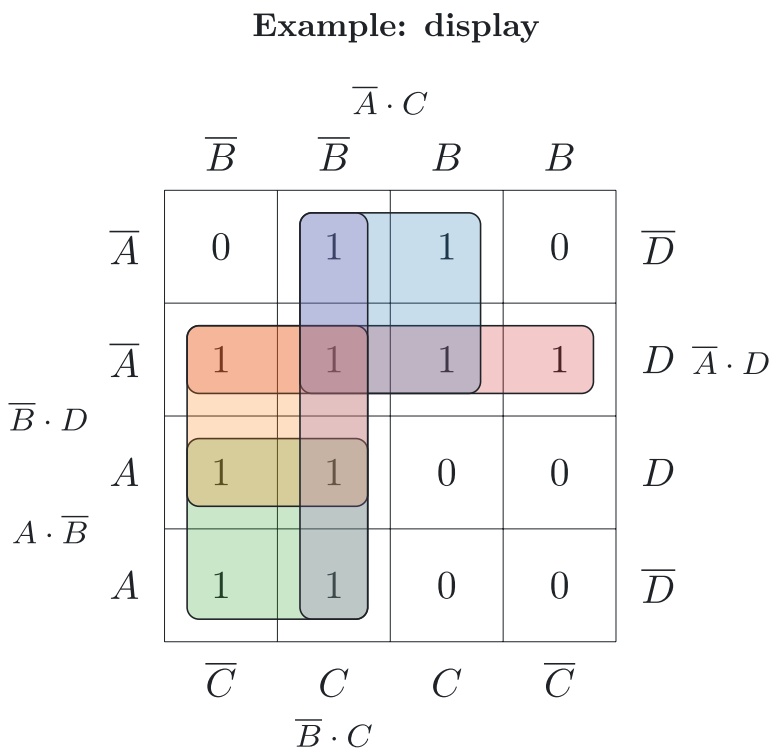
- Implicants and prime implicants
- Essential and redundant prime implicants
- When there is a choice

IMPLICANTS AND PRIME IMPLICANTS

- Implicant: a product term that implies the function – wherever the term is 1, the function is 1
- Prime implicant: an implicant that no larger implicant contains – a loop as large as possible

Example: the display of a smartphone

- A: call active, B: proximity sensor covered
- C: power button pressed, D: phone lifted
- Y: display on – during a call while the sensor is free, outside a call with the button or the lift
- Five loops as large as possible: the five prime implicants of the display

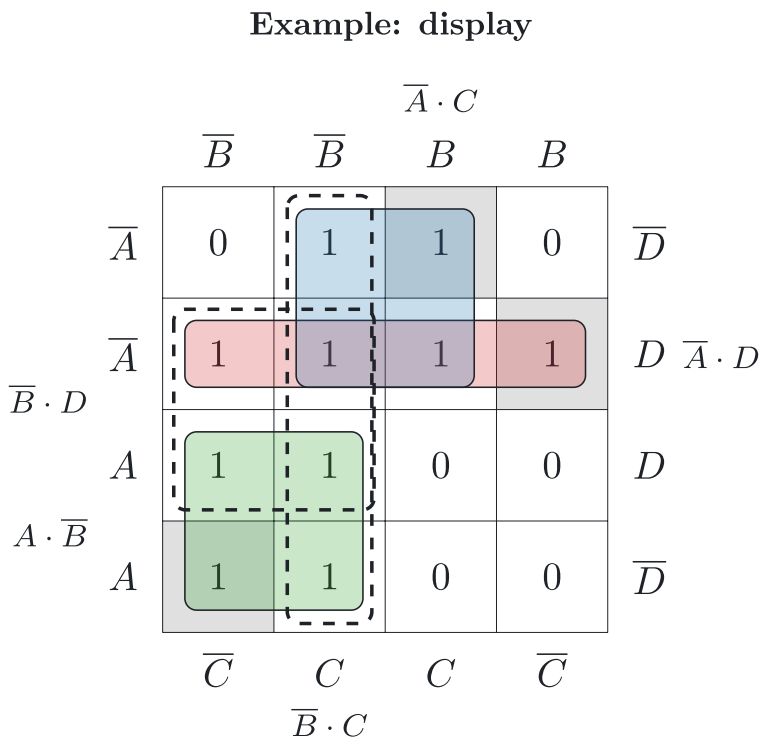


ESSENTIAL AND REDUNDANT PRIME IMPLICANTS

- Essential prime implicant: covers a one that no other prime implicant covers – in every minimal expression
  - Display: minterms 5, 6 and 8, shaded – $\overline{A} \cdot D$, $\overline{A} \cdot C$ and $A \cdot \overline{B}$
- Redundant prime implicant: all its ones are covered by the essential ones – leave it out
  - Display: $\overline{B} \cdot D$ and $\overline{B} \cdot C$, dashed

Minimal disjunctive normal form (DNF):

$$Y = \overline{A} \cdot C + \overline{A} \cdot D + A \cdot \overline{B}$$



WHEN THERE IS A CHOICE

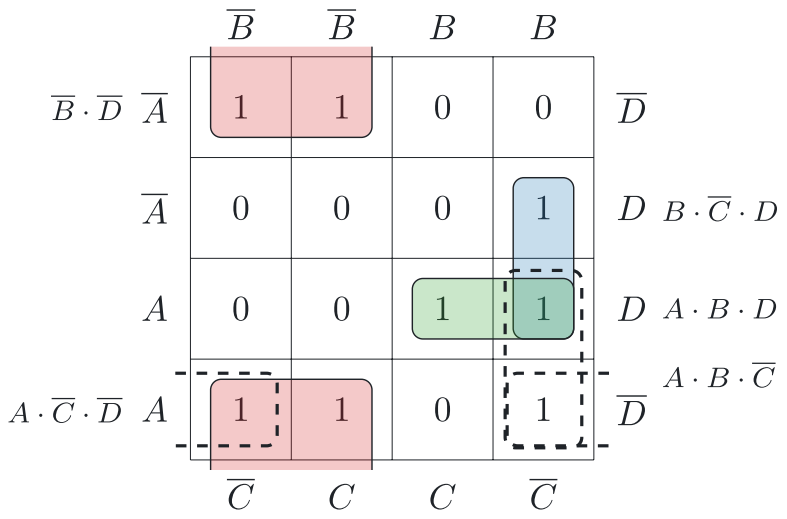
- If the essential prime implicants leave ones uncovered, as few other prime implicants as possible cover them

Example: a function with four inputs

$$Y(A,B,C,D) = \sum m(0,2,5,8,10,12,13,15)$$

- Essential, solid loops: $\overline{B} \cdot \overline{D}$, $B \cdot \overline{C} \cdot D$ and $A \cdot B \cdot D$
- Left uncovered: minterm 12, bottom right
- Two dashed loops can cover it: $A \cdot B \cdot \overline{C}$ or $A \cdot \overline{C} \cdot \overline{D}$
- So there are two minimal DNFs of the same size – unlike the canonical DNF, the minimal DNF is not always unique

Example: two minimal forms



2 The Quine–McCluskey algorithm

Step 4 of the synthesis (DET-03) by table instead of by eye: an algorithm in two parts that a computer can follow

- From the map to the table
- Part 1: finding all prime implicants
- Part 2: selecting with the prime implicant chart

FROM THE MAP TO THE TABLE

- Rules for a computer instead of the eye: like the map, the algorithm unites terms that differ in only one variable

Example: the ten ones of the display (A call active, B sensor covered, C button pressed, D phone lifted)

$$Y(A,B,C,D) = \sum m(1,2,3,5,6,7,8,9,10,11)$$

- Minterm number = row of the truth table, A to D read as a binary number: row 3 is 0011
- Code: 1 for a variable as it is, 0 for a negated one – minterm 3, $\overline{A} \cdot \overline{B} \cdot C \cdot D$, has the code 0011
- Group = number of ones in the code: terms that can unite lie in consecutive groups, so only these are compared

Table 1: the ten minterms of the display in their groups

Group	Minterm	A	B	C	D
1	1	0	0	0	1
1	2	0	0	1	0
1	8	1	0	0	0
2	3	0	0	1	1
2	5	0	1	0	1
2	6	0	1	1	0
2	9	1	0	0	1
2	10	1	0	1	0
3	7	0	1	1	1
3	11	1	0	1	1

PART 1: FINDING ALL PRIME IMPLICANTS

- Part 1 builds one table after another, starting from table 1 with the minterms in their groups
- Next table: compare each term with each term of the next group
  - Codes differ in exactly one position: write the united term with a dash there into the next table and tick both
  - Table 2: minterm 1 (0001) and minterm 3 (0011) unite into 00-1, which covers minterms 1 and 3
  - From table 3 on, the dashes must also stand in the same position: 00-1 (1, 3) and 01-1 (5, 7) unite into 0--1 (1, 3, 5, 7)
- Part 1 ends when no terms can be united any more; the terms without a tick are the prime implicants

Display, table 3: five terms that cover four minterms each – the five loops of 1.1

Minterms	A	B	C	D	Term
1, 3, 5, 7	0	-	-	1	$\overline{A} \cdot D$
1, 3, 9, 11	-	0	-	1	$\overline{B} \cdot D$
2, 3, 6, 7	0	-	1	-	$\overline{A} \cdot C$
2, 3, 10, 11	-	0	1	-	$\overline{B} \cdot C$
8, 9, 10, 11	1	0	-	-	$A \cdot \overline{B}$

PART 2: SELECTING WITH THE PRIME IMPLICANT CHART

- Prime implicant chart: one more table, with a row for each prime implicant from part 1 and a column for each minterm of the function
- Display (A call active, B sensor covered, C button pressed, D phone lifted): the columns are the minterms of its ten ones; ● marks where a prime implicant covers a minterm

Prime implicant	1	2	3	5	6	7	8	9	10	11
$\overline{A} \cdot D$	●		●	●		●				
$\overline{A} \cdot C$		●	●		●	●				
$A \cdot \overline{B}$							●	●	●	●
$\overline{B} \cdot D$	●		●					●		●
$\overline{B} \cdot C$		●	●						●	●

- A column with a single ●: only one prime implicant covers this minterm – it is essential
  - Columns 5, 6 and 8: $\overline{A} \cdot D$, $\overline{A} \cdot C$ and $A \cdot \overline{B}$ – together they cover all ten minterms
- Minterms left over: as few other prime implicants as possible cover them, as in 1.3

3 Functions with many inputs

Step 4 of the synthesis (DET-03) for the heat pump: six inputs and 64 cells

- When the map becomes too large
- The algorithm with many inputs
- Logic minimization software: exact and heuristic

WHEN THE MAP BECOMES TOO LARGE

Example: the release of a heat pump

- A: heating demand, B: hot-water demand
- C: summer mode, above 19 °C
- D: block by the energy supplier
- E: below the bivalence temperature of −7 °C
- F: frost protection
- Y: the heat pump may start – frost protection always; without a block, hot water in every season and heating outside summer mode and above −7 °C
- Six inputs: 64 cells in four maps, one for each combination of A and B; X where C and E would both be 1, which never occurs
- By eye, error-prone, because neighbours also lie in the other maps – the map suits the eye up to five inputs

Example: heat pump with six inputs

$\overline{A} \cdot \overline{B}$

	$\overline{D}$	$\overline{D}$	D	D
$\overline{C}$	0	0	0	0
$\overline{C}$	1	1	1	1
C	1	X	X	1
C	0	X	X	0

$\overline{A} \cdot B$

	$\overline{D}$	$\overline{D}$	D	D	
$\overline{C}$	1	1	0	0	$\overline{F}$
$\overline{C}$	1	1	1	1	F
C	1	X	X	1	F
C	1	X	X	0	$\overline{F}$

$A \cdot \overline{B}$

	$\overline{E}$	E	E	$\overline{E}$
$\overline{C}$	1	0	0	0
$\overline{C}$	1	1	1	1
C	1	X	X	1
C	0	X	X	0

$A \cdot B$

	$\overline{E}$	E	E	$\overline{E}$	
$\overline{C}$	1	1	0	0	$\overline{F}$
$\overline{C}$	1	1	1	1	F
C	1	X	X	1	F
C	1	X	X	0	$\overline{F}$

THE ALGORITHM WITH MANY INPUTS

- The same two parts, only the tables grow
- Don't cares count as ones while uniting and get no column in the chart
- Heat pump: 31 ones and 16 don't cares – part 1 needs six tables and finds four prime implicants
  - $C \cdot E$ contains only don't cares; the other three are essential
- The effort grows exponentially with the number of inputs – realistic for 15 to 20 inputs at most

Table	Terms	Minterms per term	Prime implicants found
1	47	1	–
2	119	2	–
3	116	4	$A \cdot \overline{C} \cdot \overline{D} \cdot \overline{E}$
4	54	8	–
5	12	16	$B \cdot \overline{D}$ and $C \cdot E$
6	1	32	F

Minimal DNF of the heat pump release:

$$Y = F + B \cdot \overline{D} + A \cdot \overline{C} \cdot \overline{D} \cdot \overline{E}$$

LOGIC MINIMIZATION SOFTWARE: EXACT AND HEURISTIC

Two kinds of algorithm for two-level minimization, the search for a minimal DNF or minimal conjunctive normal form (CNF):

1. Exact: the Quine–McCluskey algorithm finds the minimal form for sure – realistic for 15 to 20 inputs at most
2. Heuristic: the Espresso heuristic logic minimizer works with simplifying assumptions – as a rule the same result, in individual cases slightly off the optimum, but with much lower effort

Synthesis tools, which turn VHDL code into a circuit, minimize it on the way (DET-03, 6.3): a computer can minimize for you – but the minimal circuit is not always the best one (chapter 4)

4 Hazards

Beyond step 6 of the synthesis (DET-03): the check of the truth table does not see short pulses

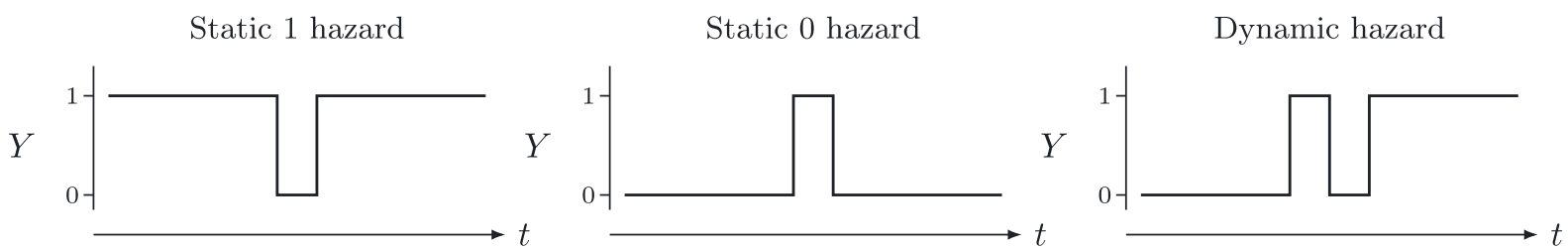
- Static and dynamic hazards
- One input, two paths
- Hazard analysis: the redundant prime implicant returns
- Two inputs at once: the function hazard

STATIC AND DYNAMIC HAZARDS

- Hazard: different paths have different delays, and while the inputs change, the output can show short, unwanted pulses

Named after what the output should do:

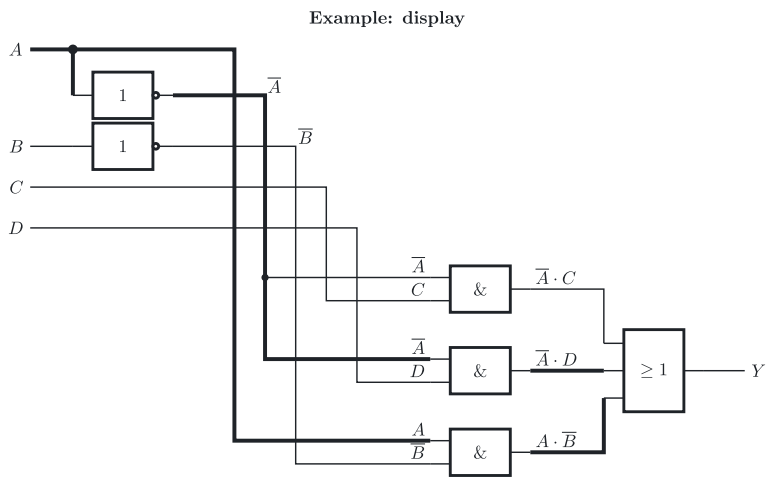
- Static 1 hazard: should stay 1, drops to 0 for a moment
- Static 0 hazard: should stay 0, rises to 1 for a moment
- Dynamic hazard: should change once, changes several times



- Logic hazard: caused by the change of a single input; function hazard: by several inputs at once

ONE INPUT, TWO PATHS

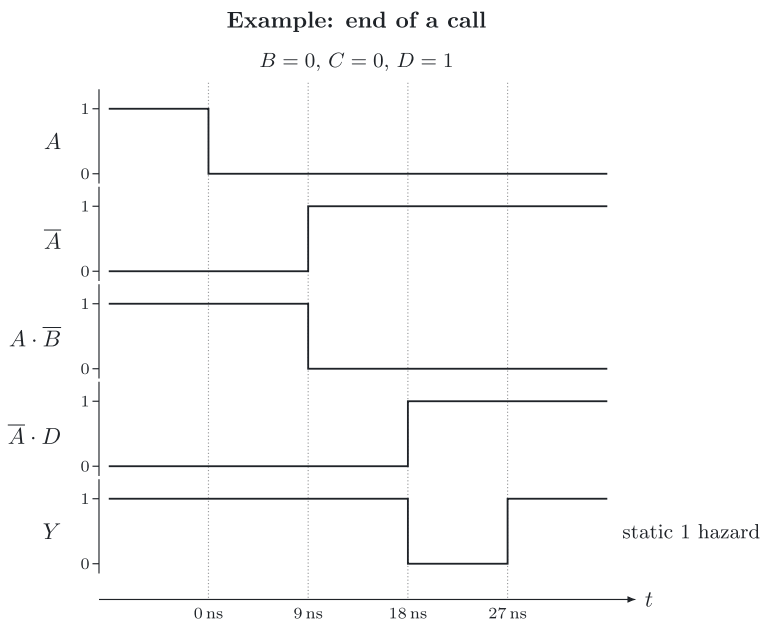
Example: the display, $Y = \overline{A} \cdot C + \overline{A} \cdot D + A \cdot \overline{B}$, with A call active, B sensor covered, C button pressed, D phone lifted



- Race: A reaches the OR gate on two paths (thick lines), and the path through the inverter is slower

Open in the simulator

End of a call: A falls, and Y should stay 1



- $A \cdot \overline{B}$ falls after 9 ns and $\overline{A} \cdot D$ rises only after 18 ns: a static 1 hazard from 18 ns to 27 ns

HAZARD ANALYSIS

- Mark of a logic hazard in a two-level circuit: two neighbouring ones in different loops

Procedure for a two-level circuit:

1. Set up the minimal DNF or CNF
2. Determine whether a single input leads from one prime implicant to a separate one
3. Add redundant prime implicants that unite them

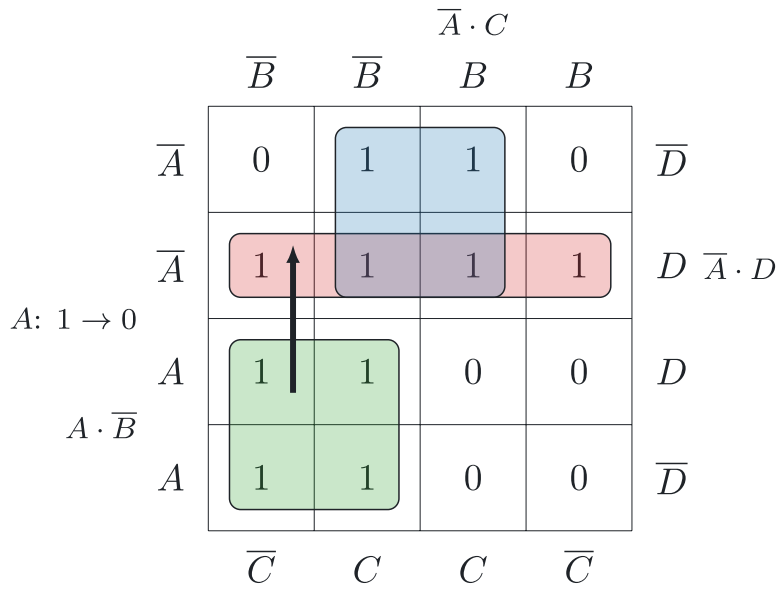
- Display, with $B = 0$, a change of A leads from $A \cdot \overline{B}$
  - to $\overline{A} \cdot D$ if $D = 1$: add $\overline{B} \cdot D$
  - to $\overline{A} \cdot C$ if $C = 1$: add $\overline{B} \cdot C$

Hazard-free, with two more AND gates – no longer minimal:

$$Y = \overline{A} \cdot C + \overline{A} \cdot D + A \cdot \overline{B} + \overline{B} \cdot C + \overline{B} \cdot D$$

Open in the simulator

Example: end of a call

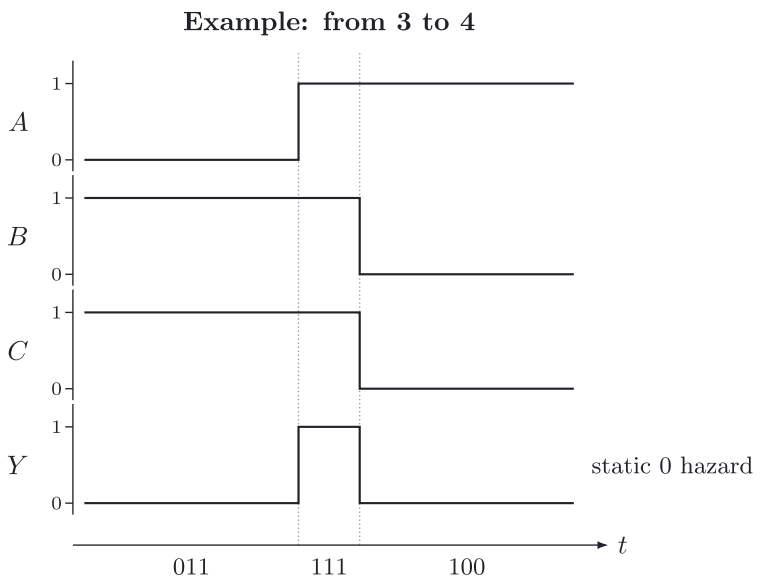


TWO INPUTS AT ONCE: THE FUNCTION HAZARD

- Several inputs change at once: depending on which one acts first, the output can briefly take a wrong value

Example: a counter with the bits A, B and C goes from 3 (011) to 4 (100)

- $Y = A \cdot B$ reports 6 or more: $Y = 0$ at 3 and at 4
- A rises before B falls: the gate sees 111 or 110 – a short 1, a static 0 hazard
- No additional term helps: the function itself demands $Y = 1$ for 111 and 110
- Remedy: read the output only when all changes have settled, with a clock (DET-06), or make sure that the inputs never change at the same time



HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts

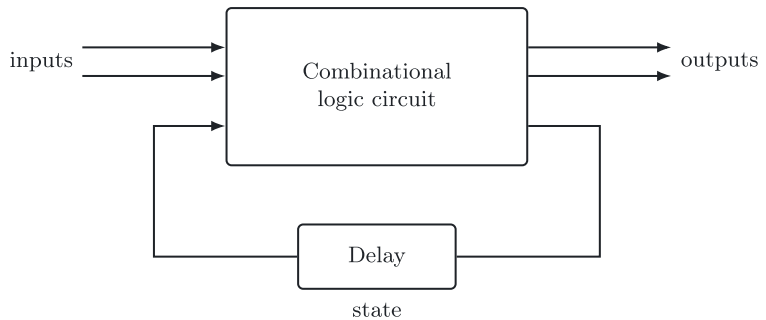
5 Combinational logic circuits and sequential circuits

Where a short pulse does harm: in circuits that remember

- Without memory and with memory
- When a short pulse leaves a mark

WITHOUT MEMORY AND WITH MEMORY

- Combinational logic circuit: the output depends only on the present inputs – a truth table describes it completely
  - Smartphone: whether the display is on, whether the phone rings
  - Heat pump: whether the compressor may start, the collective fault
- Sequential circuit: the output also depends on past inputs – it stores information as its state
  - Smartphone: the notice of a missed call, kept until you look at it; the step counter
  - Heat pump: the counter of the compressor starts; a fault message, kept until acknowledged



Structure of a sequential circuit: a combinational logic circuit with feedback through a delay

WHEN A SHORT PULSE LEAVES A MARK

Why does the robot vacuum drop its Wi-Fi connection at night?

- A short pulse reaches the reset input of its Wi-Fi module – the module restarts and has to reconnect

Whether a short pulse does harm depends on what it reaches:

- A slow load does not react: nobody sees 9 ns on the display, and a relay needs milliseconds
- A sequential circuit, such as the Wi-Fi module, takes in the pulse and changes its state
  - Heat pump: a counter that counts every rise of the release counts a compressor start that never happened

Remedy: remove the logic hazards (4.3), or read the signals only at fixed moments with a clock (DET-06)



Image by Qwen-Image 2.1

6 What have you learnt?

- The building blocks of knowledge
- Answers to the guiding questions

WHAT YOU HAVE LEARNT

- Prime implicants
- Quine–McCluskey algorithm
- Hazard analysis
- Combinational logic circuits and sequential circuits

ANSWERS TO THE GUIDING QUESTIONS

- Yes: the Quine–McCluskey algorithm finds all prime implicants table by table and selects with the prime implicant chart; large functions need heuristic software such as Espresso
- Robot vacuum: a hazard – a short pulse at the reset of its Wi-Fi module restarts the module
- Heat pump: leave it to the algorithm – three product terms: frost protection; hot water without a block; heating without a block, outside summer mode and above the bivalence temperature



Image by Qwen-Image 2.1

Your turn: first on your own, then together.