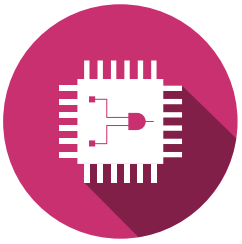

Programmable logic circuits (ex. PLD)

Exercises embedded systems

1 Programmable Logic Device (PLD)

1.1 Exclusive OR function in a PAL

Draw the fuses in the scheme of the following figure to generate the functions `out1 <= in1 xor in2;` and `out2 <= in2 xor in3 xor in4;`.

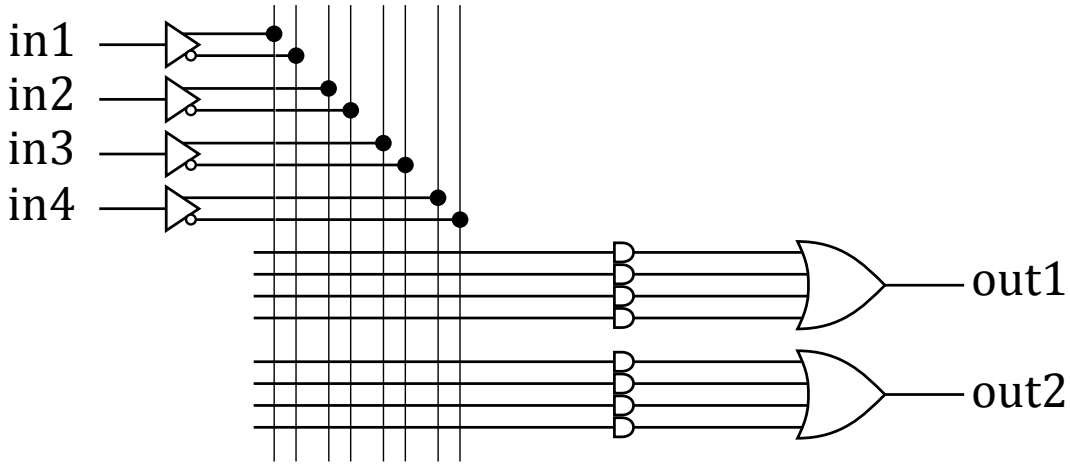

Figure 1: PAL4H2

Determine the number of necessary inputs of the OR gate to calculate the parity for an 8-bit word.

1.2 Counter in a PAL

Draw the fuses in the scheme of the following figure to create a 2-bit counter. The input `in1` serves as an "enable" control: when `in1 = '0'` the counter is stopped and when `in1 = '1'` then it counts.

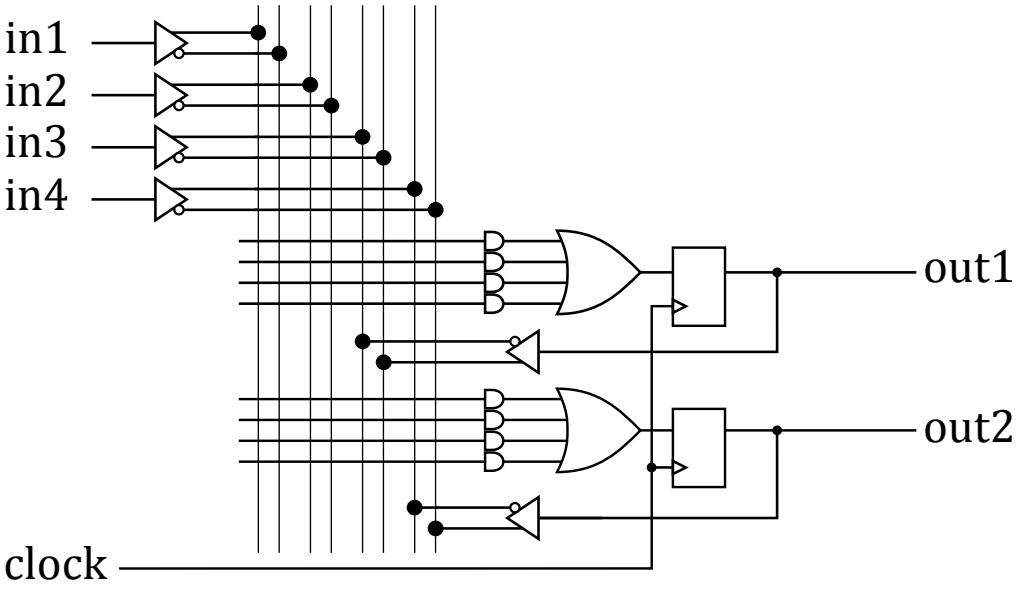

Figure 2: PAL4R2

Set the number of necessary inputs of the OR gate to create an 8-bit counter.

2 Complex Programmable Logic Device (CPLD)

2.1 Number comparison

For two numbers coded on 8 bits, determine the number of necessary inputs of the OR gate in a CPLD to make the following comparisons:

In a circuit, you need to determine if a regularly incrementing counter overflows a value stored in a register. Find a way to make this circuit as small as possible.

3 Field Programmable Gate Array (FPGA)

3.1 Multiplexer

Estimate the number of 8-input OR gates needed to create a multiplexer from 16 to 1.

Estimate the number of 4-input logic blocks needed to create the same multiplexer.

3.2 Counter

Based on the basic circuit of a FPGAs, as shown in the following figure, draw the scheme of a 6-bit counter, which counts continuously (without "Enable" control).

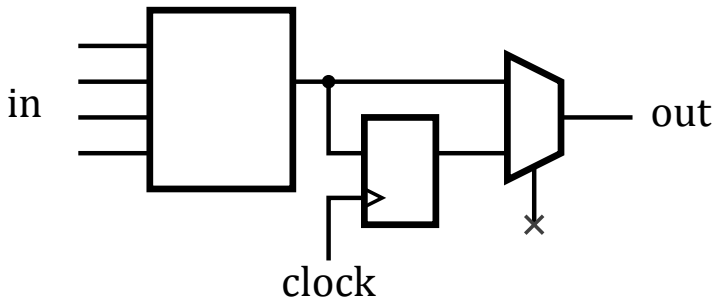

Figure 3: Base logic block of a FPGA

From this diagram, determine the number of basic blocks used to implement a 16-bit counter.

Acronyms

CPLD **C**omplex **P**rogrammable **L**ogic **D**evice. 2

FPGA **F**ield **P**rogrammable **G**ate **A**rray. 2, 3

PLD **P**rogrammable **L**ogic **D**evice. 1–3