

Projektowanie układów logicznych

Logical circuits design

Zestaw zadań nr 5

Tasks' set 5

Demultiplekser

Wejścia:

- n wejść adresowych $A_0 \dots A_{n-1}$
- 1 wejście danych D_{in}

Wyjścia:

- 2^n wyjść danych $D_0 \dots D_{(2^n-1)}$

Działanie układu:

- Na wyjście D_x gdzie x to liczba binarna podana na wejścia adresowe $A_0 \dots A_{n-1}$ jest podawany stan wejścia D_{in}
- Na pozostałe wyjścia jest podawany stan 0

Zadania: zaprojektuj i wykonaj symulację demultiplekserów o 2, 4, 8 i 16 wyjściach.

Demultiplexer

Inputs:

- n address inputs $A_0 \dots A_{n-1}$
- 1 data input D_{in}

Outputs:

- 2^n data outputs $D_0 \dots D_{(2^n-1)}$

Circuit function:

- The output D_x where x is the binary number given to the address inputs $A_0 \dots A_{n-1}$ has state equal to the state of the input D_{in}
- Other outputs have state 0

Tasks: design and simulate demultiplexers that have 2, 4, 8 and 16 outputs.

1. Zaprojektuj bramki NAND i NOR jako złożenie bramek AND, OR i NOT, a następnie wykonaj ich symulację

Design NAND and NOR gates by composition of AND, OR and NOT gates. Then perform simulation of the designed gates.

2. Zaprojektuj bramkę XOR poprzez złożenie bramek AND, OR i NOT (wzór dostępny w prezentacji). Następnie wykonaj symulację, która pokaże czy przebiegi sygnałów zaprojektowanej bramki są identyczne z bramką XOR dostępną w Xilinx

Design XOR gate by composition of AND, OR and NOT gates (formula is in the presentation). Then perform a simulation which shows are waveforms of the designed gate and the XOR gate from Xilinx toolbox identical.

3. Zaprojektuj układy i wykonaj symulacje, które potwierdzą (lub nie ☺) poniższe prawa logiki:
 - a. Rozdzielność
 - b. Prawa De Morgana
 - c. Prawo wyłączonego środka (tertium non datur)
 - d. Syllogizm

Design circuits and perform simulations that prove (or not ☺) the following laws of logic:

- a. Distribution
- b. De Morgan's laws
- c. Law of excluded middle (tertium non datur)
- d. Syllogism