|  |  |
| --- | --- |
|  | Architektury komputerów  Wyższa Szkoła Zarządzania i Bankowości w Krakowie Informatyka II rok 2023 |

Ćwiczenie nr 1 – Bramki logiczne – Protokół obserwacji i pomiarów

|  |  |  |
| --- | --- | --- |
| Imię i Nazwisko | Grupa, termin zajęć | Ocena |
| Mazurok Oleksandr | 1K211, 05.03.2025 |  |
|  |  |  |

Zadanie 2.1 Bramka NAND w układach cyfrowych

**Tablica stanów logicznych bramki NAND**

Należy uzupełnić tablicę prawdy dla bramki NAND. Zakładamy, że wejścia bramki oznaczone są jako A i B, natomiast jej wyjście ma oznaczenie Y.

|  |  |  |
| --- | --- | --- |
| Wejście  A | Wejście  B | Wyjście  Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Y=~(A\*B)**

**2.2 Trójwejściowej bramki NAND**

Należy uzupełnić tablicę prawdy dla bramki NAND. Zakładamy, że wejścia bramki oznaczone są jako A, B i C, natomiast jej wyjście ma oznaczenie Y.

|  |  |  |  |
| --- | --- | --- | --- |
| Wejście  A | Wejście  B | Wejście  C | Wyjście  Y |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Zapis algebraiczny funkcji realizowanej przez bramkę:**

**Y=~(A\*B\*C)**

**2.3 Inne funktory logiczne**

Na podstawie obserwacji uzupełnić poniższą tabelę opisująca funkcje logiczne realizowane przez inne bramki

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | NOT | AND | OR | NOR | XOR | XNOR |
| Symbol graficzny | |  |  |  |  |  |  |
| We A | We B |  |  |  |  |  |  |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| Funkcja logiczna | | ~A | A\*B | A+B | ~(A+B) | ~(A⬄B)/A⊕B | A⬄B/A⊙B |

**2.4 Realizacja funkcji logicznych NOT, AND, OR i NOR, XOR z samych bramek typu NAND**

Należy zamieścić odpowiednie fragmenty zrzutów ekranu wykonane podczas zajęć, szkice schematów, skany szkiców lub samodzielnie wykonane rysunki schematów w postaci cyfrowej.

1. Funkcja NOT

**A\*A=A**

**NAND(A,B = A) = ~(A\*A) = ~A = NOT(A)**



1. Funkcja AND

**~~A= A**

**~NAND(A, B) = ~~(A\*B) = A \* B**





1. Funkcja OR

**~~A = A**

**~A = ~(A\*A) = NAND(A, A)**

**OR(A, B) = A + B = ~~(A + B) = ~(~A \* ~B) = NAND(~A,~B) = NAND(P,Q)**

**P = ~A = NAND(A,A)**

**Q = ~B = NAND(B,B)**



1. Funkcja NOR

**NOR(A,B) = ~(A+B) = ~(~[~(A+B)]) = ~(~[~A\*~B)]) = ~NAND(P, Q) = NAND(NAND(P, Q), NAND(P, Q)) = NAND(G, G)**

**P = ~A = NAND(A, A)**

**Q = ~B = NAND(B, B)**

**G = NAND(P, Q)**



1. Funkcja XOR

|  |  |  |  |
| --- | --- | --- | --- |
| A\B | 0 | 1 |  |
| 0 | 0 | 1 |  |
| 1 | 1 | 0 |  |

**A \* ~B + ~A \* B = ~[~(A \* ~B + ~A \* B)] = ~[~(A \* ~B) \* ~(~A \* B)] = NAND(P, Q)**

**P = ~(A \* ~B) = NAND(A, ~B)**

**Q = ~A \* B = NAND(~A, B)**



**2.5 Realizacja funkcji logicznych NOT, OR , AND, NAND i XOR z samych bramek typu NOR**

Należy zamieścić odpowiednie fragmenty zrzutów ekranu wykonane podczas zajęć, szkice schematów, skany szkiców lub samodzielnie wykonane rysunki schematów w postaci cyfrowej.

1. Funkcja NOT

**A + A = A**

**NOR(A, A) = ~(A+A) = ~A**



1. Funkcja OR

**~~A = A**

**A + B = ~~(A + B) = ~(NOR(A, B)) = NOR(NOR(A, B), NOR(A, B)) = NOR(P,P)**

**P = NOR(A, B)**



1. Funkcja AND

**A \* B = ~[~(A\*B)] = ~[~A+~B] = NOR(P, Q)**

**P = ~A = NOR(A, A)**

**Q = ~B = NOR(B, B)**



1. Funkcja NAND

**NAND(A, B) = ~(A \* B) = ~A + ~B = ~[~(~A + ~B)] = ~[NOR(P, Q)] = NOR(NOR(P, Q), NOR(P, Q)) = NOR(K, K)**

**P = ~A = NOR(A, A)**

**Q = ~B = NOR(B, B)**

**K = NOR(P, Q)**



1. Funkcja XOR

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

**(A \* ~B) + (~A \* B) = ~(~[(A \* ~B) + (~A \* B)]) = ~(NOR(M, N)) = NOR(K, K)**

**P = ~A = NOR(A,  A)**

**Q = ~B = NOR(B,  B)**

**M = A \* ~B = ~~(A \* ~B) = ~[~A + B] = NOR(P, B)**

**N = ~A \* B = ~~(~A \* B) = ~[A + ~B] = NOR(A, Q)**

**K = NOR(M, N)**

**G = NOR(K, K)**



2.6 Półsumator

a) Tabele prawdy dla wyjść półsumatora

**S - suma**

|  |  |  |
| --- | --- | --- |
| A/B | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

**C - przeniesienie**

|  |  |  |
| --- | --- | --- |
| A/B | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |

b) Funkcje realizowane przez poszczególne wejścia można zapisać w postaci kanonicznej:

S=.................XOR(A, B)............................ C=..................AND(A, B)..................

c) Schemat układu półsumatora (zrzut fragmentu ekranu Multisim-a) :



2.7 Układ odejmujący

a) Tabele prawdy dla wyjść układu odejmującego

R - różnica

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

B - pożyczka

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |



b) Funkcje realizowane przez poszczególne wejścia można zapisać w postaci kanonicznej:

**R** = .........**XOR(A, B)**................ **B** =……..**~A \* B**……

Schemat zbudowanego układu (zrzut fragmentu ekranu Multisim-a):

**2.8 Komparator binarny**

**S**

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

S = XNOR (A,B)

Schemat zbudowanego komparatora dla dwóch liczb dwubitowych



Czy można zbudować układ komparatora, wykorzystując funktor EXOR?  
 Jeśli tak, poniżej należy zamiesić schemat odpowiedniego układu.

**XNOR zwraca nam jedynkę tylko w tedy, gdy ma dwie jednakowe wartości na wejściach. Tymczasem XOR zwraca jedynkę odwrotnie, w tedy, gdy ma dwie różne wartości na wejściach. W taki sposób, z XOR-em jeszcze muszę być NOT, który będzie zmieniać wynik XOR na odwrotny i dawać wartość XNOR z odpowiednimi wartościami na wejściu. Czyli wykonać porównania używając tylko bramki XOR, nie uda się otrzymać na wyjściu poprawny wynik porównania.**

Objaśnić jak można rozszerzyć układ do dowolnej ilości bitów.

Używając **zasady przechodniości:** jeśli **a** = **b**,a **b =** **c**, to **a** = **c**, możemy rozszerzać układ do dowolnej ilości bitów za pomocą bramek **XNOR** oraz **AND**



2.9 Bramka NAND TTL jako układ analogowy

**2.9.1. Charakterystyka przejściowa bramki (zrzut ekranu lub prawidłowo opracowany wykres)**

Wartości zmierzone:

a) Napięcie wyjściowe w stanie wysokim: VOH = …...........

b) Napięcie wyjściowe w stanie niskim: VOL = ................

c) Maksymalne napięcie wejściowe, przy którym jeszcze nie zmienia się napięcie wyjściowe:

VIL =..............

d) Minimalne napięcie wejściowe, przy którym jeszcze nie zmienia się napięcie wyjściowe:

VIH = ….......

**2.9.2. Prąd pobierany przez wejście bramki**

Wejście bramki w stanie logicznym wysokim pobiera prąd IIH = ….......

Wejście bramki w stanie logicznym niskim pobiera prąd IIL = ….........

**2.9.3. Pomiar charakterystyki wyjściowej (wyznaczanie obciążalności wyjścia bramki)**

W tym punkcie mierzymy prąd IOHmax, który płynie, gdy wyjście bramki obciążone jest taką wartością rezystancji, że napięcie wyjściowe przestaje być „rozumiane” przez bramkę jako stan wysoki (tzn. napięcie wyjściowe jest bliskie przekroczenia wartości VIH).

Ten prąd wynosi IOHmax = ….......

Biorąc pod uwagę zmierzony tutaj prąd IOHmax oraz prąd IIH pobierany przez wejście bramki można obliczyć ilość wejść bramek, które można dołączyć do jednego wyjścia bramki. Ta ilość to:

N = ….......