

Projekt - Układ kombinacyjny

Opis zadania: Należy napisać program, który na podstawie opisu połączeń bramek logicznych (NAND, NOR, AND, OR, NOT) dla zadanego pliku z wartościami na wejściach generuje plik opisujący stany wyjściowe. Dodatkową wyznaczaną informacją jest czas propagacji sygnałów.

Program, na podstawie zewnętrznych plików wejściowych, które zawierają opis struktury układu oraz wartości sygnałów wejściowych, ma generować plik wynikowy prezentujący stany danych wyjść oraz odpowiedni dla nich czas propagacji.

Format danych wejściowych

Plik opisujący układ logiczny:

Układ jest definiowany poprzez listę bramek logicznych.

Każda bramka w pliku jest zapisana w formie:

Typ_bramki wejście1 wejście2 wyjście czas_propagacji

Typ_bramki - określa rodzaj bramki(NAND, AND, NOR, OR, XOR, NOT etc.)

wejście1, wejście2 - sygnały wejściowe bramki (np. a,b,c)

wyjście - nazwa sygnału wyjściowego danej bramki (np. d,e,f)

czas_propagacji - czas propagacji sygnału przez bramkę (np. 10)

Układ może zawierać wiele bramek korzystających z tych samych sygnałów oraz dowolną liczbę sygnałów pośrednich.

System zakłada, że opis bramek jest poprawny. W razie błędu, program zakończy działanie.

Dane wejściowe:

Drugi plik wejściowy zawiera zestawy wartości logicznych przypisanych do sygnałów wejściowych układu.

Pierwsza linia określa nazwy wejść(np. a,b)

W kolejnych liniach znajdują się ich wartości dla poszczególnych symulacji.

Przykład:

a b

0 0

0 1

1 0

1 1

Dla każdej linii program przeprowadza osobną symulację działania całego układu logicznego.

Interpretacja układu przez program i przetwarzanie danych

Program powinien rozpocząć działanie od wczytania pliku opisującego układ logiczny.

Każda bramka jest zapisywana jako instancja specjalnie przygotowanej klasy, która przechowuje:

- jej typ,
- nazwy sygnałów wejściowych,
- nazwa sygnału wyjściowego,
- przypisany czas propagacji,
- bieżąca wartość wyjścia,
- skumulowany czas dotarcia sygnału do wyjścia.

Podczas wczytywania każda bramka jest zapisywana w np. wektorze jako samodzielny obiekt posiadający wszystkie potrzebne informacje do późniejszego przetwarzania.

Przechowywanie bramek w ten sposób umożliwi, bądź ułatwi, iterowanie po całym układzie oraz dostęp do sygnałów pośrednich.

Następnie, wczyta i zapisze w kolejnej tablicy/wektorze drugi plik wejściowy zawierający wartości wejść.

Obliczenia

Dla każdego zestawu wejść program przypisuje początkowe wejścia do odpowiednich wejść bramek i następnie wielokrotnie iteruje po liście bramek próbując obliczyć wyjścia tych bramek, których wejścia są już znane.

Gdy wartości obu wejść bramki są znane, program oblicza wartość wyjściową zgodnie z typem bramki i następnie ustala czas propagacji.

Iterowanie po układzie kończy się, gdy program nie może już obliczyć żadnej innej wartości bramki.

Wynik działania programu

Dla każdego zestawu wartości wejściowych (z 2. pliku wejściowego) program powinien wyprowadzić wartości wybranych sygnałów wyjściowych oraz ich czas propagacji.

Wyniki będą zarówno wypisywane na ekran jak i zapisywane do pliku wynikowego.

Założone cechy i ograniczenia

Układ nie zawiera sprzężeń zwrotnych

Jedynie operacje bramek wpływają na czas propagacji

Kolejność bramek w pliku nie musi odpowiadać kolejności tych bramek w układzie - program sam ustali kolejność obliczeń

Format danych wejściowych zakłada poprawność strukturalną

Każdy zestaw wartości wejściowych jest traktowany jako niezależna symulacja, wykonywana na tej samej strukturze układu.

Program ma być prosty do rozbudowy, by mógł w przyszłości pracować na rozszerzonych danych, np. o bramki z ilością wejść większą niż 2 lub z większą ilością znanych wejść początkowych