

Architektura Komputerów 2 - Projekt

Implementacja fizyczna układów cyfrowych

Prowadzący:

Dr inż. Piotr Patronik

Skład grupy projektowej:

Kacper Kucharczyk 248834

Artur Sobolewski 248913

17 czerwca 2020, Wrocław

Spis treści

1	Wstęp	3
2	Podstawy działania sumatora PPA	3
3	Cześć praktyczna	4
3.1	Przedstawienie problemu	4
3.2	Rozwiązanie problemu	4
3.3	Wyniki i dyskusja	6
3.4	Wnioski	7
4	Literatura	7

1 Wstęp

Centralne jednostki przetwarzające przeprowadzają różne złożone procesy używając podstawowych operacji arytmetycznych i logicznych. Celem działań projektantów układów cyfrowych było stworzenie systemów liczących, zoptymalizowanych pod względem czasowym i kosztowym. Jedną z opracowanych idei było równoległe wytwarzanie przeniesień (ang. Parallel Prefix Adder, PPA). Taka koncepcja znacznie skraca czas wykonywania operacji oraz przez zniesienie uzależnienia kolejnych bitów sumy od generowanych szeregowo przeniesień z wcześniejszych pozycji.

Wraz z dynamicznym rozwojem techniki cyfrowej wzrasta złożoność układów i proces ich projektowania staje się coraz trudniejszy. W celu ułatwienia tego procesu powstały języki opisu sprzętu (HDL – ang. Hardware Description Language), które pozwalają opisać działanie sprzętu zamiast całej struktury układu. Jednym z tych języków jest Verilog, opracowany w 1984. Istnieje również wiele narzędzi służących do syntezy logicznej i fizycznej zaprojektowanych za pomocą tego języka układów. W tym projekcie do utworzenia fizycznego obwodu i przeprowadzenia testów został wykorzystany program Qflow wraz z Yosys.

2 Podstawy działania sumatora PPA

W kontekście sumatorów PPA stosujemy dwa typy modeli:

- Sumatory seryjne - polegające na dodawaniu liczb bit po bicie. Przez swoją specyfikę pracy stosuje się je głównie w układach w których ważna jest dokładność działań, zaś czas ich wykonania jest mniej istotny.
- Sumatory równoległe - polegające na dodawaniu liczb równoległe, wszystkie bity w tym samym czasie.

Proces obliczenia sumy w sumatorach PPA może zostać podzielony na 3 części:

- Funkcja generacji przeniesienia, funkcja propagacji oraz tzw. pół-suma

$$g_i = x_i \text{ AND } y_i$$

$$p_i = x_i \text{ OR } y_i$$

$$h_i = x_i \text{ XOR } y_i$$

Na podstawie powyższych równań utworzony został moduł generacji i propagacji przeniesień (Rysunek 1(a))

- Obliczenie przeniesienia na następną pozycję

$$P_{k:i} = P_{k:j+1} \text{ AND } P_{j:i}$$

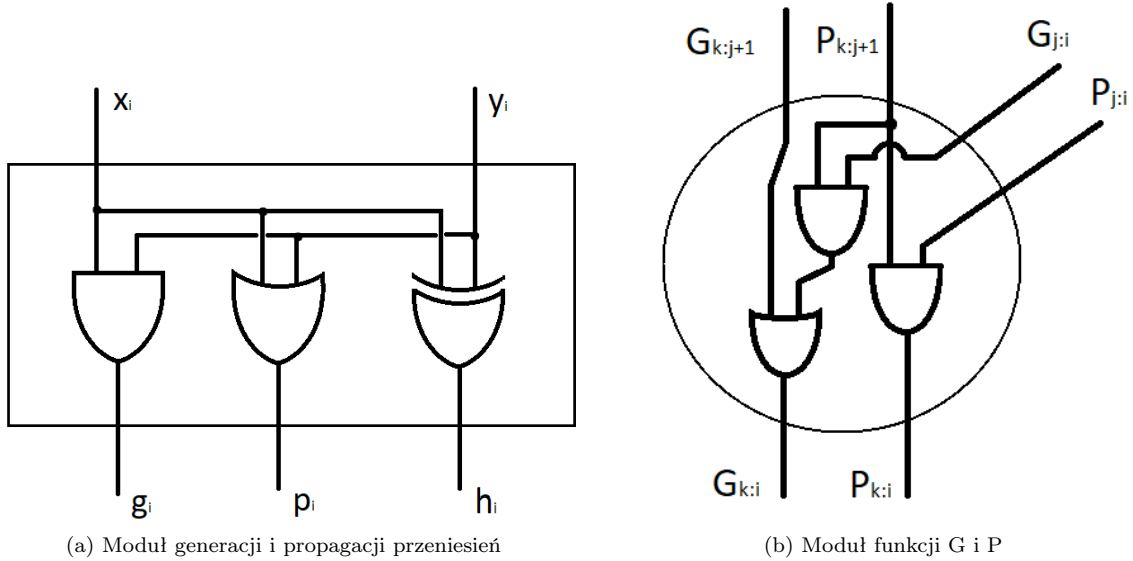
$$G_{k:i} = G_{k:j+1} \text{ OR } (P_{k:j+1} \text{ AND } G_{j:i})$$

Na podstawie powyższych równań utworzony został moduł funkcji G i P (Rysunek 1(b))

- Obliczenie sumy końcowej oraz końcowego przeniesienia

$$S_i = P_i \text{ XOR } C_{i-1}$$

$$c_{i+1} = G_{i:0} \text{ , } i - \text{pozycja najwyższego bitu}$$



Rysunek 1: Moduły użyte w schemacie sumatora.

3 Część praktyczna

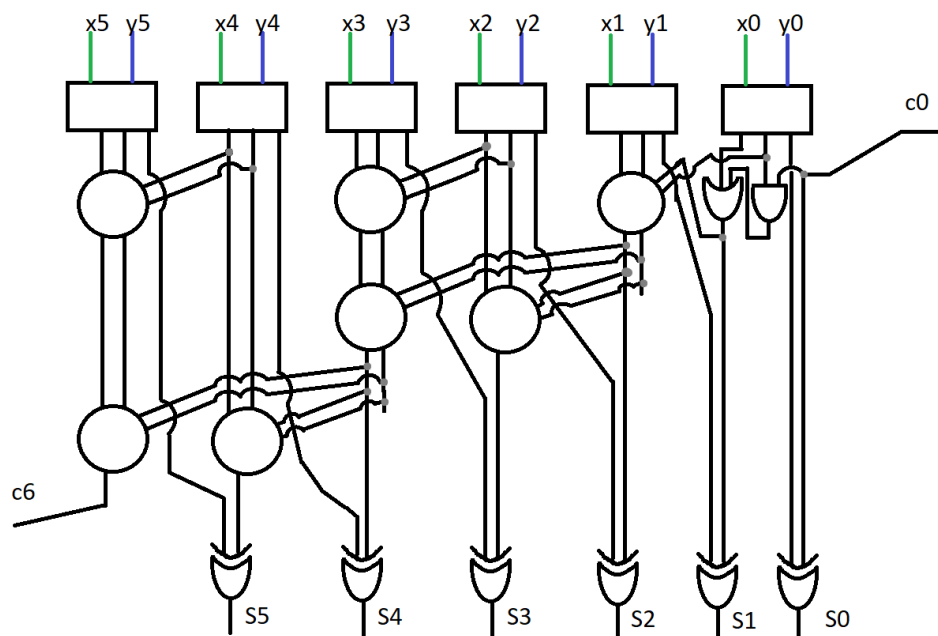
3.1 Przedstawienie problemu

Celem naszego zadania było zaprojektowanie 6-bitowego sumatora prefiksowego oraz jego implementacja w języku opisu sprzętu HDL, jakim jest Verilog. Kolejnym etapem jest dokonanie syntezy logicznej oraz fizycznej zaprojektowanego układu, te procesy wykonamy za pomocą narzędzi Yosys oraz Qflow. Do układu należało również wykonać testbench, który będzie testował naszą implementację sumatora pod kątem wszelkich przeoczeń w projekcie poprzez wprowadzanie do układu różnych sygnałów wejściowych oraz kontrolowanie wyjść układu. Testy powinny wyczerpywać wszelkie szczególne przypadki jakie mogą wystąpić w procesie faktycznego użytkowania.

3.2 Rozwiązanie problemu

Do wykonania naszego projektu wykorzystaliśmy architekturę Ladnera-Fischera (Rysunek 2.) w implementacji PPA. Zastosowana architektura pozwala na zminimalizowanie poziomów logicznych, tzn. minimum elementów GP układu. Dzięki temu można znacznie zmniejszyć niezbędną przestrzeń chipa oraz przyspieszyć działanie sumatora.

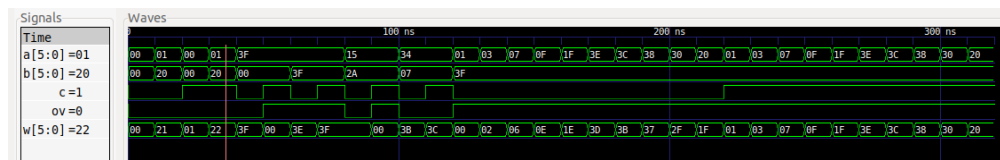
Układ uwzględnia możliwość istnienia przeniesienia początkowego (na Rysunku 2 jako C_0), które w znacznym stopniu może wpływać na otrzymywane wyniki. Wyróżnione jest również przeniesienie końcowe tzw. overflow (na Rysunku 2 jako C_6).



Rysunek 2: Schemat sumatora PPA w architekturze Landera-Fischera.

Następnie powyższy schemat (Rysunek 2) wykorzystaliśmy do zaprojektowania symulatora w syntezowalnym kodzie Verilog. W celu przetestowania go stworzyliśmy adekwatny testbench, który miał za zadanie zasymulować zwykłe przypadki użycia naszego układu jak i te skrajne, najczęściej wywołujące błędy obliczeniowe. Napisany przez nas test można uznać za wyczerpujący, gdyż rozważone są wszystkie sytuacje generujące przekłamania ostatecznego wyniku.

Kod naszej symulacji kompilowany był przy pomocy aplikacji Icarus, która pozwala zbadać zależności czasowe poszczególnych zmiennych (Rysunek 3.). Zmienna a i b oznaczają składniki sumy podane do sumatora, c - przeniesienie zerowe, ov - przeniesienie wyjściowe układu, w - wyjście układu (suma).



3.4 Wnioski

Układ który zaprojektowaliśmy, choć w teorii trywialny, przysporzył nam kilku problemów, głównie związanych ze specyfiką używanych narzędzi. Nauczyło nas to jednak pracy na mniej popularnych przyrządach programistycznych oraz obchodzenia się z dość surowymi opisami ich działania.

Narzędzia tj. Yosys i Qflow umożliwiają projektantom nowych układów cyfrowych dokonanie wstępnej syntezy logicznej, w celu przetestowania poprawności ich działania przed fazą tworzenia układu fizycznego. Pozwala to na wcześniejsze wprowadzanie poprawek, zanim implementacja zostanie wdrożona, dzięki czemu można zaoszczędzić wiele środków. Ponad to narzędzie Yosys pozwala w znacznym stopniu zoptymalizować układ, stosując moduły z własnych bibliotek.

W fazie implementacji fizycznej układu program Qflow przeprowadził szereg czynności służących do przygotowania zaprojektowanego układu do jego fizycznego utworzenia.

4 Literatura

- "Szybkie Sumatory" Prof. Janusz Biernat
- "Design and Implementation of Parallel Prefix Adder for Improving the Performance of Carry Lookahead Adder" R. Payal, M. Goel, P. Manglik
- Qflow 1.3 Digital Synthesis Flow Tutorial