

PNIEWSKI Roman, PNIEWSKA Julia

SYMULACJA UKŁADÓW Z LOGIKĄ REWERSYJNĄ W PROGRAMIE QUCS

Streszczenie

Szybkość systemów cyfrowych (w tym nowoczesnych komputerów) ograniczają zjawiska związane ze stratami energii i wydzielaniem ciepła. Rozwiązaniem alternatywnym jest wykorzystanie logiki rewersyjnej w syntezie systemów cyfrowych. W artykule przedstawiono podstawowe bramki rewersyjne i metody ich modelowania z wykorzystaniem symulatora QUCS.

WSTĘP

Konwencjonalne komputery wykorzystują dwuwartościową logikę Boole'a. Funkcje opisujące układ cyfrowy wykorzystują dwa operatory AND i OR. Te dwie operacje, posiadają kilka wejściowych i jeden bit wyjściowy, co powoduje zmniejszenie informacji na wyjściu. Gdy układ ma mniej dostępnych stanów, to jego entropia staje się mniejszą. Ponieważ druga zasada termodynamiki zabrania zmniejszenie entropii w zamkniętym układzie, to zmniejszenie entropii w jednym miejscu musi być skompensowane generacją entropii w innym miejscu. Generowana entropia wskutek skasowania bitu informacji wynosi:

$$\Delta S = k_B T \cdot \ln 2 \quad (1)$$

A zatem komputery pracujące na podstawie algebry Boole'a są zawsze urządzeniami rozpraszającymi energię nie mniejszą niż:

$$k_B T \cdot \ln 2 \quad (2)$$

Ta generacja ciepła w ciągu procesu obliczeniowego stanowi ograniczenie na możliwą szybkość komputera ze względu na ilość generowanego ciepła.

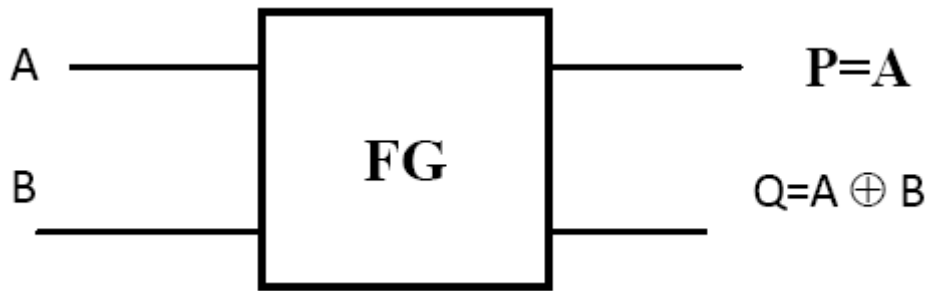
Fredkin i Toffoli udowodnili, że rewersyjne logiczne bramki mogą tworzyć podstawę dla komputera uniwersalnego. Zaproponowali trzybitową bramkę rewersyjną, która nosi nazwę bramki Fredkina. Bramka Fredkina może być zastosowana do realizacji rewersyjnej bramki AND.

1. LOGIKA REWERSYJNA

Bramki rewersyjne, wykorzystywane w syntezie układów cyfrowych, a szczególności przy przetwarzaniu kwantowym charakteryzują się „odwracalnością” logiczną tzn. połączenie szeregowe dwóch identycznych bramek generuje sygnał wyjściowy równoważny wejściowemu. W syntezie systemów cyfrowych najczęściej stosowane są bramki:

- Feynmana
- Toffoli'ego
- Fredkina

Bramka Feynmana



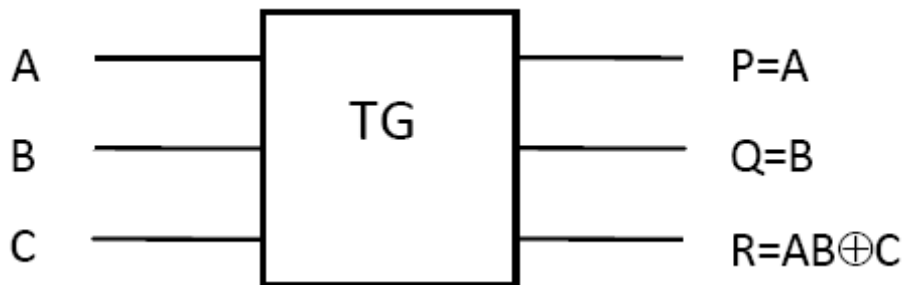
Rys. 1. Bramka Feynmana

Funkcję przejść bramki przedstawiono w tabeli Tab1.

Tab. 1. Funkcja przejść bramki Feynmana

WEJŚCIA		WYJŚCIA	
A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Bramka Toffoli'ego



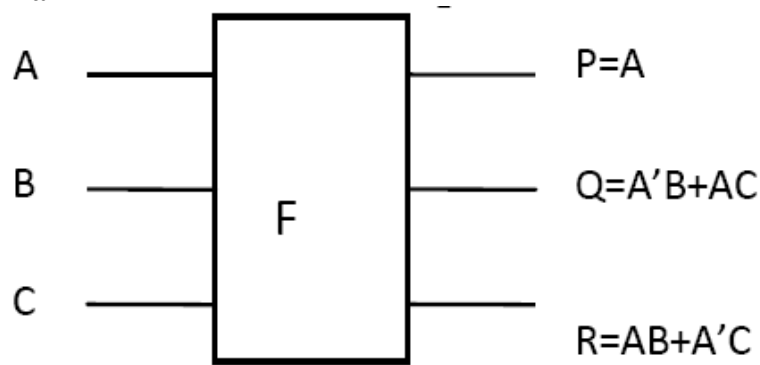
Rys. 2. Bramka Toffoli'ego

Właściwości bramki charakteryzuje funkcje przejść przedstawiona w tabeli Tab.2.

Tab. 2. Funkcja przejść bramki Toffoli'ego

WEJŚCIA			WYJŚCIA		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Bramka Fredkina



Rys. 3. Bramka Fredkina

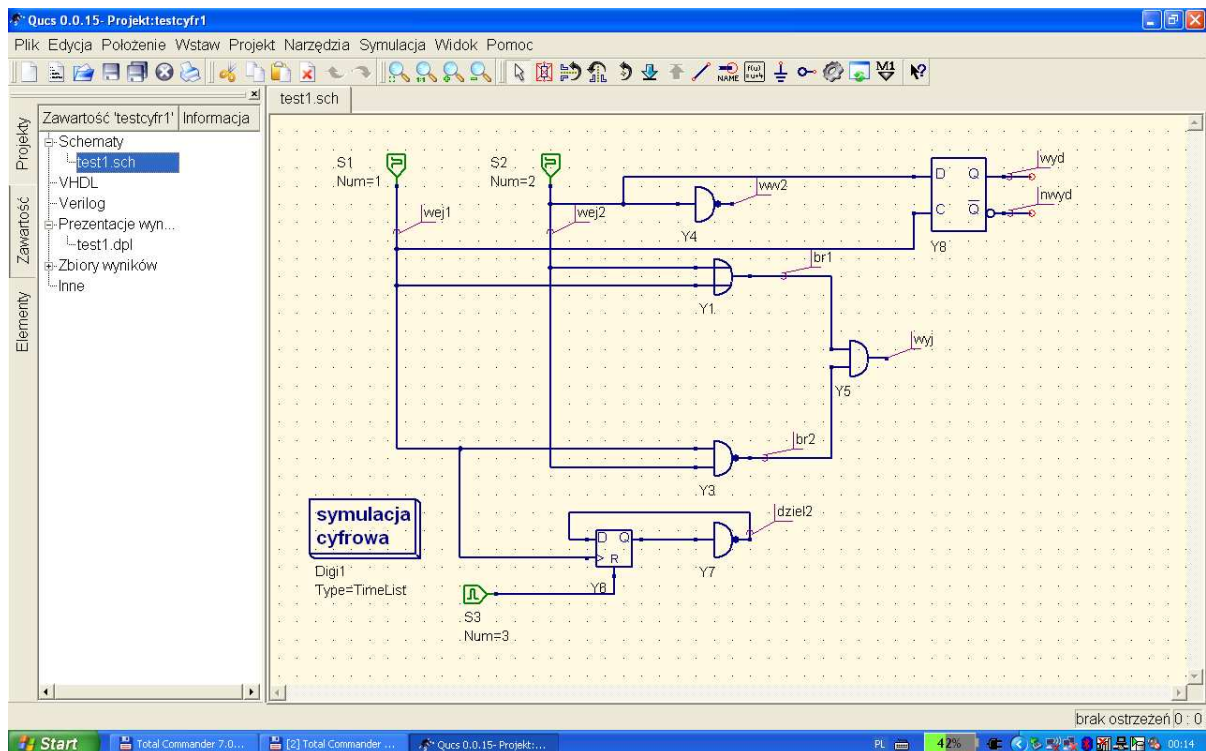
W odróżnieniu od bramki Torffoli, która ma dwa kontrolne bity i jeden celowy bit, bramka Fredkina ma jeden kontrolny kubit i dwa bity celowe. Celowe bity wymieniają się jeżeli kontrolny bit jest równy 1, w przeciwnym wypadku one pozostają bez zmian. W tablicy Tab.3. pokazano funkcję przejściową bramki

Tab. 3. Funkcja przejść bramki Fredkina

WEJŚCIA			WYJŚCIA		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

2. SYMULATOR QUCS

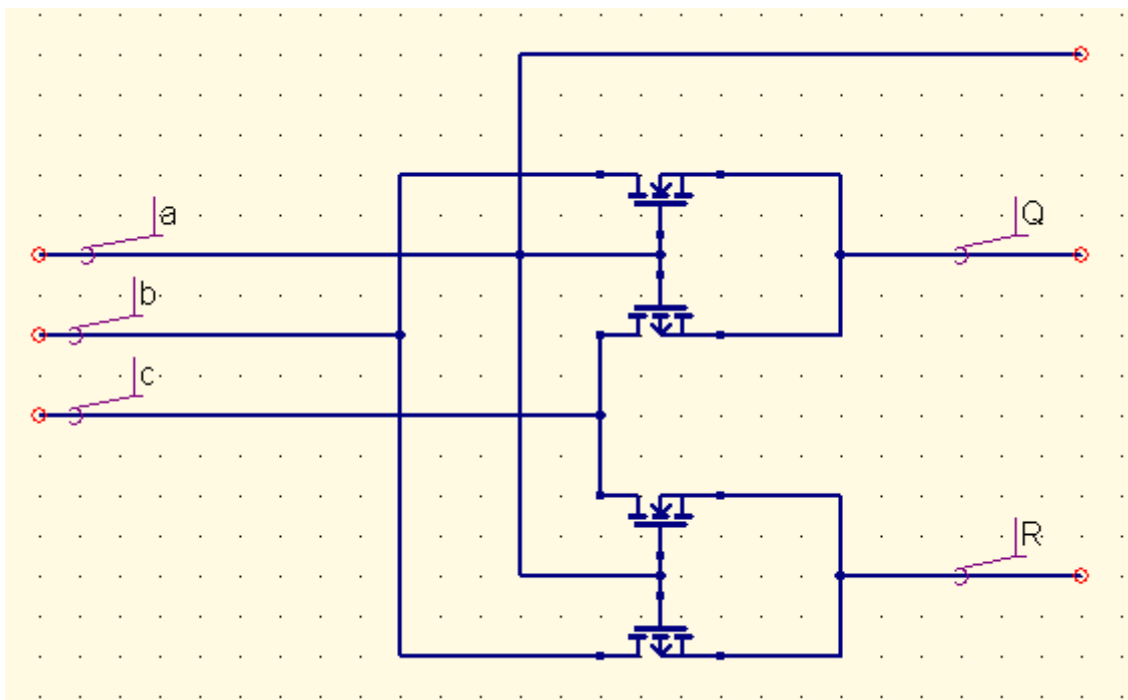
Program QUCS (Quite Universal Circuit Simulator) jest darmowym symulatorem układów elektronicznych. Pierwsza wersja programu była przeznaczona dla systemu Linux. Aktualne wersje programu pracują w systemach operacyjnych Linux i Windows. Oprogramowanie jest w pełni darmowe, dostępne są kody źródłowe programu, co umożliwia jego modyfikację. Program umożliwia symulację układów analogowych i cyfrowych. Do symulacji analogowej wykorzystano algorytm SPICE, natomiast symulacja cyfrowa układów przebiega wieloetapowo. Na podstawie schematu ideowego program generuje listę połączeń i zapisuje w pliku netlist.txt. Lista zapisana jest w języku VHDL (VHSIC Hardware Description Language). Następnie dokonywana jest konwersja na język C (przy wykorzystaniu środowiska FreHDL). Do kompilacji otrzymanego kodu źródłowego zastosowano kompilator Mingw. Procesem symulacji w QUCS „steruje” plik wsadowy „qucsdigi.bat”. Modyfikacja tego pliku pozwala na dowolne sterowanie procesem symulacji. Dzięki takiemu rozwiązaniu jest możliwe dołączenie do symulatora własnych programów działających „wsadowo”.



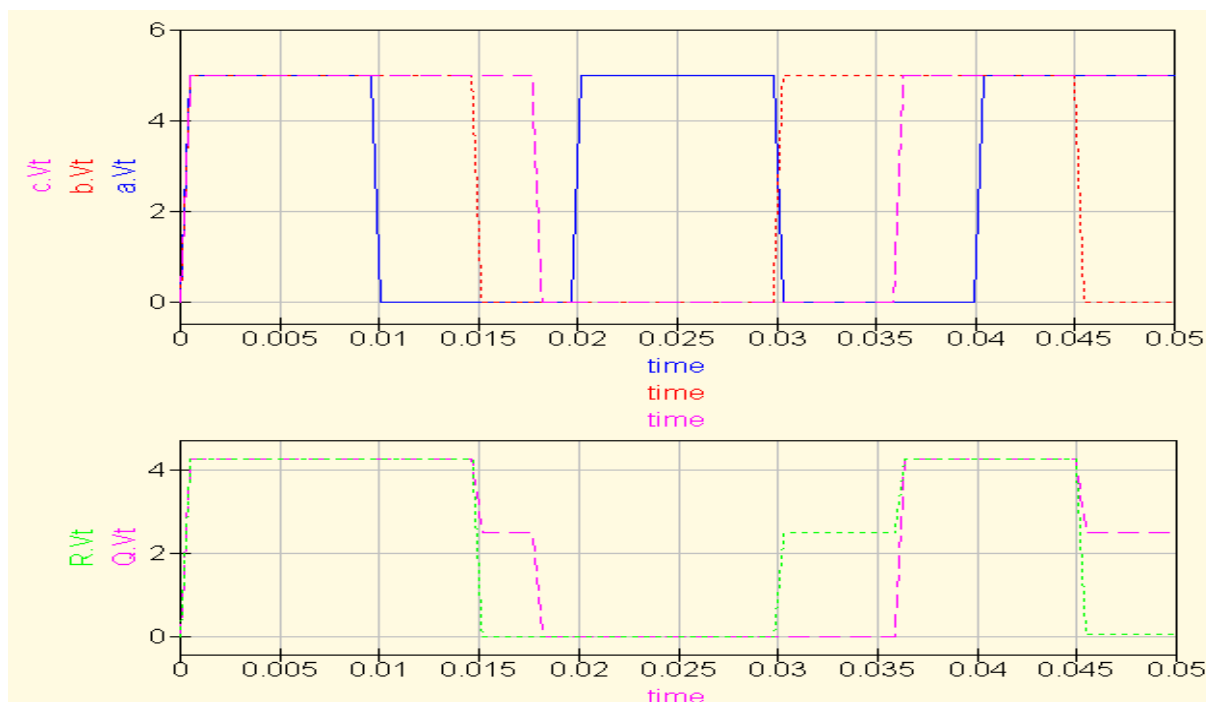
Rys. 4. Okno programu QUCS

3. MODELOWANIE UKŁADÓW Z LOGIKĄ REWERSYJNĄ

Układy logiki rewersyjnej można zasymulować w programie QUCS zarówno w technice analogowej jak i cyfrowej. Analogowe modelowanie układów rewersyjnych jest korzystne przy analizie wpływu uszkodzeń poszczególnych elementów na działanie systemu.



Rys. 5. Bramka Fredkina zbudowana z tranzystorów MOS



Rys. 6. Wyniki symulacji w programie QUCS

Na rysunku 5 przedstawiono model bramki Fredkina zbudowany w oparciu o przełączniki zrealizowane w technice MOSFET. Rys.6 przedstawia przykładowe przebiegi napięć uzyskane w wyniku symulacji.

W przypadku formalnej weryfikacji układu cyfrowego, zrealizowanego w technice rewersyjnej znacznie korzystniejsze jest modelowanie bramek rewersyjnych jako elementy cyfrowe. Poniżej przedstawiono modele bramek opisane w języku VHDL.

```

Library ieee;
Use ieee std_logic.1164..all;
Entity fredkin is
Port(A, B, C : in std_logic;
P, Q, R : out std_logic);
end fredkin;
architecture ckt of fredkin is
signal Abar, S1, S2, S3, S4 : std_logic;
begin
P<= A;
Abar<= not A;
S1<=Abar and B;
S2<= A and C;
Q<= S1 xor S2;
S3<= Abar and C;
S4<= A and B;
R<= S3 xor S4;
End ckt;

```

```

Library ieee;
Use ieee std_logic.1164..all;
Entity toffoli is
Port(A, B, C : in std_logic;
P, Q, R : out std_logic);
end toffoli;
architecture ckt of toffoli is
signal s1 : std_logic;
begin
P<= A;
Q<= B;
S1<=A and B;
R<= S1 xor C;
End ckt;

```

PODSUMOWANIE

Przy syntezie układów cyfrowych jednym z głównych etapów jest symulacyjna weryfikacja poprawności działania układu. Na wielu uczelniach powstało dedykowane oprogramowanie do symulacji układów opartych o logikę rewersyjną. Autorzy proponują wykorzystanie standardowego symulatora do analizy układów rewersyjnych. Zaletą proponowanego rozwiązania jest możliwość symulacji „hybrydowej” tzn. jednoczesna symulacja układów logicznych, opartych na algebrze Boole’a i układów rewersyjnych

BIBLIOGRAFIA

1. Anas N. Al-Rabadi, *Reversible logic synthesis*. Springer 2004.
2. Gennady P. Berman., *Introduction to Quantum Computers*. World Scientific 1999.
3. Pniewski R. *Metoda oceny bezpieczeństwa cyfrowych systemów automatyki kolejowej*. Monografie Wyd. UTH Radom, ISSN 1642-5278

SIMULATION REVERSIBLE LOGIC IN QUCS

Abstract

The speed digital systems (including modern computers) limit phenomena associated with energy losses and heat generation. An alternative is to use in the synthesis of reversible logic digital systems. The article presents reversible gates and methods of reverse modeling of a simulator QUCS.

Autorzy:

dr inż. **Roman Pniewski** – Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Transportu i Elektrotechniki, e-mail: r.pniewski@uthrad.pl

Julia Pniewska – studentka Fizyki, Wydział Matematyki, Fizyki i Informatyki UMCS w Lublinie