Załącznik nr 2.

Implementacja algorytmu CAMELLIA w strukturach programowalnych.

Spis treści:

I. Algorytm CAMELLIA

I.1 Wprowadzenie do algorytmu blokowego Camellia

CAMELLIA jest szyfrem blokowym, operującym na 128 bitowym bloku danych oraz na kluczu o długości 128, 192 i 256 bitów. W zależności od długości klucza głównego, do poprawnej realizacji szyfrowania i deszyfrowania należy wykonać 18 (128-bitowy klucz) lub 24 (192, 256 bitów) rundy. Algorytm opiera się na sieci Feistela. Specyfikacja algorytmu CAMELLIA jest dostępna w publikacji „Specification of Camellia – a 128-bit block cipher" [1].

I.2 Algorytm Camellia w konkursie NESSIE

Konkurs NESSIE zakończył się w marcu 2003r. Propozycjami na nowy standard szyfrowania danych w Europie, zostały wybrane algorytmy blokowe: AES i CAMELLIA. Mają one wiele podobnych cech, dlatego też wiele możliwych analiz można odnieść do obu algorytmów. W ramach projektu NESSIE nie pojawiła się żadna publikacja, której wynik naruszałby bezpieczeństwo zarówno algorytmu AES jak i CAMELLIA [3].

II. Projekt Implementacji algorytmu

I.3 Założenia projektowe

Następujące założenia projektowe postanowiono przyjąć w momencie rozpoczęcia realizacji projektu implementacji sprzętowej algorytmu CAMELLIA w strukturach programowalnych:

- Realizowaną wersją będzie CAMELLIA, który będzie korzystała z 128 bitowego bloku danych używając 128 bitowego klucza.
- Implementacja w układzie oferowanym przez firmę ALTERA – układ ma być z rodziny układów FLEX 10KE.
- Realizacja projektu będzie realizowana w systemie projektowym ALTERY Max+PlusII.

- Najważniejszym kryterium projektowym jest jednak uzyskanie jak największej szybkości szyfrowania układem.
- Wybrana architektura – iteracyjna, z wcześniejszym przygotowaniem układu.

I.4 Sposób realizacji projektu

Realizacja algorytmu szyfrowania z koncepcją, wcześniejszego przygotowania układu polega na wyznaczeniu wszystkich niezbędnych danych do bezpośredniego wyznaczania podkluczy rund. Tymi niezbędnymi danymi są klucz główny oraz klucz przejściowy. Po ich wyznaczeniu układ jest gotowy do pracy, a sygnalizowane to jest „wysokim" stanem na linii informacyjnej GOTOWY.

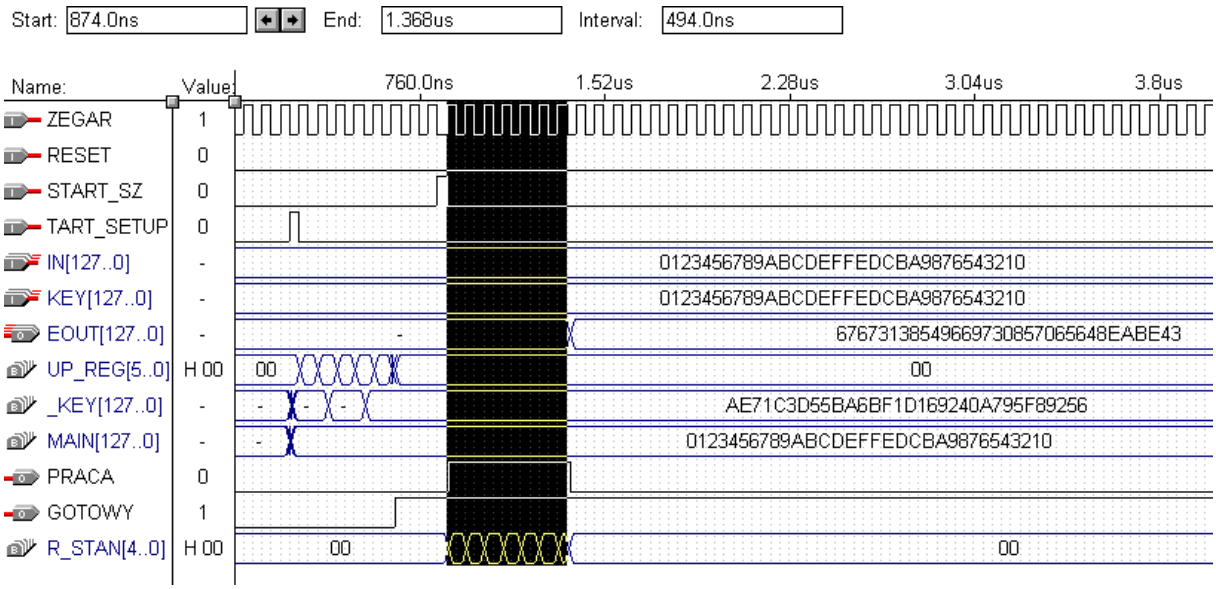
Wraz z pojawieniem się na wejściu START_SZ „narastającego" zbocza sygnału rozpoczyna się praca struktury, sygnalizowana stanem „wysokim" na wyjściu informacyjnym PRACA. Podczas pierwszego cyklu zegara na wejściu danych powinien znajdować się blok 128 bitowy do zaszyfrowania. Podczas kolejnych cykli realizowane są operacje jednoczesnego wykonania trzech rund szyfru oraz wygenerowania trzech podkluczy do kolejnych trzech rund szyfru. Inaczej wygląda cykl szósty, podczas którego wykonują się trzy ostatnie rundy i dodanie 128 bitowego podklucza. Połączenie tych dwóch operacji pozwala zaoszczędzić, jeden cykl, ale jest tylko możliwe do zrealizowania gdy wyznaczony zostanie wcześniej podklucz zamykający operację szyfrowania. Tak naprawdę jest on wyznaczany „w locie" bowiem nie ma nigdzie rejestru przetrzymującego jego wartość. Także bez zapisu w rejestrach wyliczany jest podklucz rozpoczynający operację szyfrowania (tzw. „*pre-whitening*")

III. Analiza wyników implementacji

I.5 Poprawność funkcjonalna układu

Poprawność funkcjonalna układu mogła być przeprowadzona za pomocą danych testowych, które zostały dostarczone wraz ze specyfikacją szyfru do konkursu NESSIE. Sprawdzenie jej polega na porównaniu wyników szyfrowania oraz deszyfrowania otrzymanego układu z wynikami, które są wektorami testowymi. Przeprowadzono symulację dla każdego z układów, które otrzymano w wyniku syntezy logicznej.

- Realizacja funkcji szyfrowania algorytmem CAMELLIA.



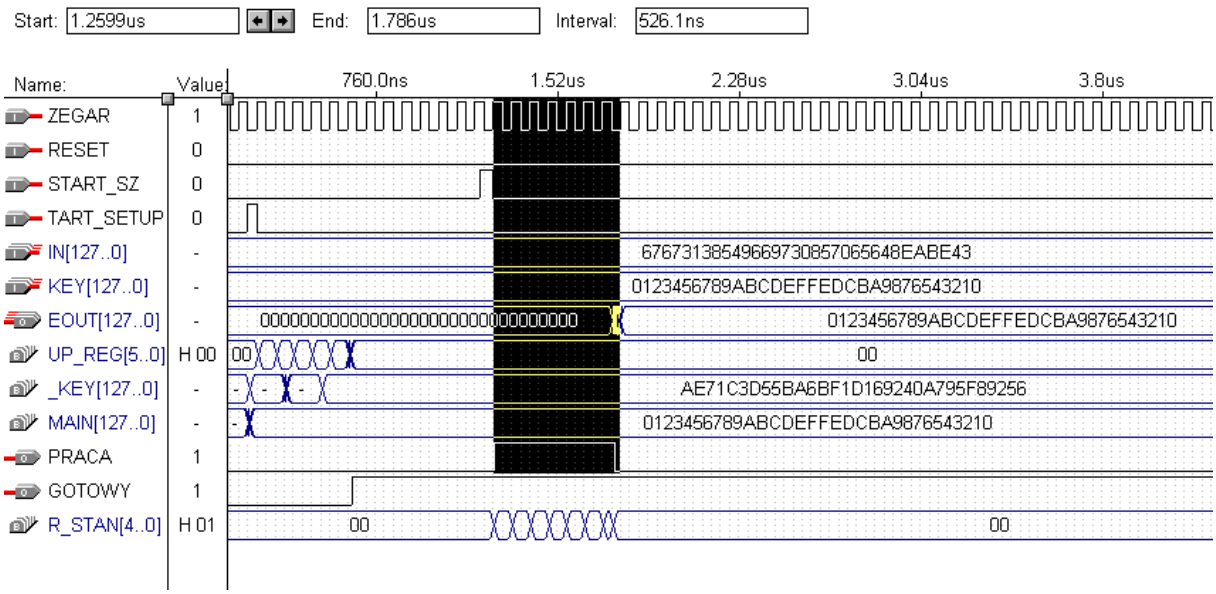
Rys. 3.1: Wyniki symulacji układu SZYFROWANIE

Po pojawieniu się „narastającego zbocza” na linii START_SETUP rozpoczyna się przygotowanie układu. Długość trwania tego procesu to 6 cykli zegara.

Na potrzeby symulacji zdecydowano taktować układ zegarem o długości trwania cyklu 76ns. Cała operacja szyfrowania trwa 7 cykli zegara. W oknie *interval* rysunku widać czas trwania całego procesu szyfrowania ok. 500 ns.

Możemy łatwo obliczyć, że szybkość takiego rozwiązania wynosi ok. 240 Mb/s.

- Realizacja funkcji deszyfrowania algorytmem CAMELLIA.



Rys. 3.1: Wyniki symulacji układu DESZYFROWANIE.

Wyniki szyfrowania i deszyfrowania oczywiście zgadzają się. Wartości wyjściowe operacji szyfrowania podane na wejściu układu DESZYFROWANIE, przy tym samym kluczu, dają na wyjściu wartości takie same jak na wejściu SZYFROWANIA. Są także zgodne z danymi testowymi zaproponowanymi przez twórców szyfru (dostępne w załączniku nr 3.).

Czas trwania cyklu w procesie deszyfrowania 76ns, ilość cykli – 7, szybkość deszyfrowania – 240Mb/s. W oknie *interval* widać, ze proces ten trwa ok. 500 ns.

I.6 Wyniki syntezy logicznej

Wyniki syntezy logicznej dla układu SZYFROWANIE i DESZYFROWANIE.

Wykorzystana struktura - EPF10K200EBC600-1.

Liczba wykorzystanych:

- wyprowadzeń wejściowych - 260,
- wyprowadzeń wyjściowych – 130,
- komórek pamięci – 2973,
- bitów pamięci wbudowanej – 49150.

Nazwa układu	Liczba zajmowanych komórek LE / bitów EAB	Procentowy udział zajętości układu
SZYFROWANIE	2973 / 49150	29% / 100 %
Warstwy sboxów (runda)	0/49150	0% / 100%
Funkcja P	128	1,3 %
Sterowanie	32	0,3 %
Rejestr wejściowy	257	2,6 %
Rejestr wyjściowy	128	1,3 %
Rejestr podklucza	912	9,1 %
Generacja podkluczy	1664	16,6 %
Dane klucza	389	3,9 %
Funkcja FL	96	1 %
Odwrotność FL	96	1 %
EPF10K200EBC600-2	9986/49150	100% / 100%

IV. Wnioski

Zaprezentowana w załączniku implementacja algorytmu CAMELLIA jest zrealizowana w taki sam sposób jak zaprezentowany w pracy projekt, który wykorzystuje operację wcześniejszego ustawienia układu. Maksymalna szybkość szyfrowania i deszyfrowania (240 Mb/s) jest nieco szybsza niż zaprezentowane przez twórców rozwiązanie iteracyjne i nieco gorsza niż rozwiązanie kombinacyjne. Wynika to oczywiście z faktu, że w pracy zostało zaprezentowane rozwiązanie hybrydowe: kombinacyjno – iteracyjne (3 rundy – 1 takt).

Wyniki szybkości implementacji algorytmów CAMELLIA i HIEROCRYPT zaprezentowanych w pracy oraz przez autorów algorytmów przedstawia poniższa tabela.

Rodzaj implementacji	Szybkość działania
TOSHIBA (Hierocrypt-3)	52,6 Mb/s
Wersja z krótkim okresem ustawienia układu (Hierocrypt-3)	115 Mb/s
Wersja z długim okresem ustawienia układu (1 cykl) (Hierocrypt-3)	190 Mb/s
Wersja z długim okresem ustawienia układu (3cykle) (Hierocrypt-3)	304 Mb/s
Wersja koncepcyjna w układach STRATIX (Hierocrypt-3)	351 Mb/s
MITSUBISHI : iteracyjna (Camellia)	227 Mb/s
MITSUBISHI : kombinacyjna (Camellia)	401 Mb/s
Wersja z ustawieniem układu (Camellia)	240 Mb/s

Tabela potwierdza twierdzenie przyjęte w podsumowaniu pracy, że dyskwalifikacja algorytmu HIEROCRYPT była nieuzasadniona. Rekomendowany algorytm CAMELLIA nie osiągnął lepszych wyników podczas przeprowadzonej analizy implementacji.

Bibliografia:

1. Nippon Telegraph and Telephone Corporation, Mitsubishi Electric Corporation – „Specification of Camellia – a 128-bit block cipher”,
2. Nippon Telegraph and Telephone Corporation, Mitsubishi Electric Corporation – “Performance of Camellia” (info.isl.ntt.co.jp/camellia/Publications/ Camellia_Performance.pdf),
3. “Portfolio of recommended cryptographic primitives”, NESSIE joury – (https://www.cosic.esat.kuleuven.ac.be/nessie/ decision-final.pdf)