STUDIA INFORMATICA 2000
Volume 21 Number 1 (39)

Lech ZNAMIROWSKI
Politechnika Śląska, Instytut Informatyki

CAD/CAE W SPRZĘTOWEJ IMPLEMENTACJI SZYFRACJI PLIKÓW [1)]

Streszczenie. W pracy przedstawiono rozwiązania w zakresie oprogramowania pozwalającego na przygotowanie plików konfiguracyjnych dla układów FPGA i umożliwiającego na bazie odpowiednich układów proste konfigurowanie oraz weryfikację projektów realizowanych w scalonych układach programowalnych FPGA. Tego typu narzędzia CAD/CAE scharakteryzowano z punktu widzenia zastosowań w sprzętowej implementacji procedur kryptografii.

CAD/CAE FOR HARDWARE ENCRYPTION OF FILES

Summary. The paper presents a selected group of software and hardware solutions capable to prepare and implement the configuration data stream into a programmable integrated circuits FPGA. The features of the evaluation boards carrying the FPGA IC applicable to testing the implemented design is reported. The characterization of the CAD/CAE tools was carried out in reference to the hardware implementation of encryption procedures.

1. Wprowadzenie

Implementacja algorytmów kryptograficznych w systemach przekazywania danych może być realizowana poprzez oprogramowanie, sprzęt lub z częściowym wykorzystaniem obu wymienionych technik. Sprzętowa realizacja szyfracji i deszyfracji danych ma na celu zminimalizowanie udziału procesorów w operacjach kryptograficznych lub ich całkowitą eliminację, co z jednej strony przyśpiesza pracę systemu, z drugiej zaś podnosi jego bezpieczeństwo. Pomija-

[1] Praca zrealizowana w ramach PC KBN nr 8T11C 026 98C/4258: "Bezpieczeństwo systemów komputerowych".

jąc zagadnienia dystrybucji kluczy stosowanych w algorytmach kryptograficznych, z punktu widzenia implementacji algorytmu w sprzęt, można wskazać dwa podejścia: jedno związane z realizacją na bazie układów scalonych ASIC (Application Specific Integrated Circuit), drugie natomiast wykorzystujące układy reprogramowalne, zwykle FPGA (Field Programmable Gate Array). Rozwiązania stosujące technologię ASIC (wykorzystujące w zakresie projektowania wybrany software np. Cadence, Mentor Graphic lub Tanner [3, 9, 13]), pozwalają osiągnąć większą szybkość pracy układu w porównaniu z technologią FPGA przy możliwości optymalizowania powierzchni struktury krzemowej i pobieranej mocy, charakteryzują się jednak zwykle ustaloną strukturą implementacji, która w przypadku konieczności zmian wymaga przeprojektowania układu i ponownego wykonania go (np. w wybranym, ogólnie dostępnym procesie technologicznym udostępnianym przez EUROPRACTICE, MOSIS lub ITE [5, 14, 11]).

W chwili obecnej koszt wielowariantowych badań modelu, a także skrócenie czasu przygotowywania prototypów rozwiązań hardware'owych można znacznie obniżyć przez wykorzystanie układów reprogramowalnych FPGA [15, 18, 28, 31, 32].

W pracy przedstawiono rozwiązania w zakresie oprogramowania pozwalającego na przygotowanie plików konfiguracyjnych dla układów FPGA i umożliwiającego na bazie odpowiednich układów proste downloadowanie oraz weryfikację projektów realizowanych w scalonych układach programowalnych. Tego typu narzędzia CAD/CAE scharakteryzowano z punktu widzenia zastosowań w sprzętowej implementacji procedur kryptografii.

2. Wybrane sprzętowe rozwiązania kryptograficzne

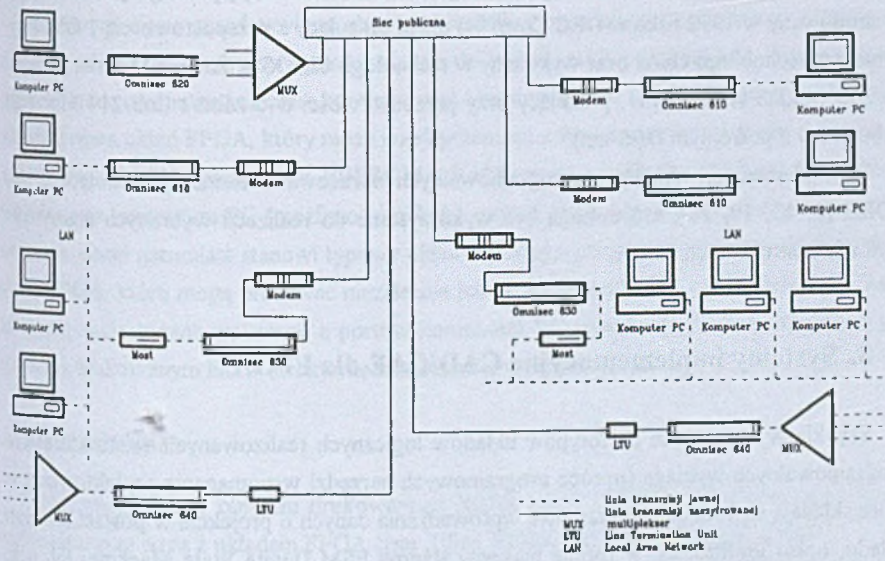
W systemach przekazywania danych proces szyfracji i deszyfracji może być realizowany na strumieniu bitów bądź równolegle na blokach bitów. Z punktu widzenia realizacji układowej jest to układ scalony realizujący algorytm kryptograficzny, a w nowszych rozwiązaniach integrujący także funkcje związanie z kluczem szyfru, timingu układu oraz odpowiedniego interfejsu układu w zależności od struktury i typu medium transmisyjnego.

Na rys. 1 przedstawiono sieciowe rozwiązanie systemu przekazywania danych z roku 1999, bazującego na szyfratorach liniowych firmy Omnisec [10].

System pozwala na strumieniową szyfrację przy przepustowości od 20 kbit/s do 2.048 Mbit/s. Szyfratory liniowe rodziny Omnisec 600 [10] rozbudowane są o układy automatycznej wymiany kluczy sesyjnych. W konfiguracji sieci heterogenicznej niezbędne jest zastosowanie w systemie układów typu most (rys. 1).

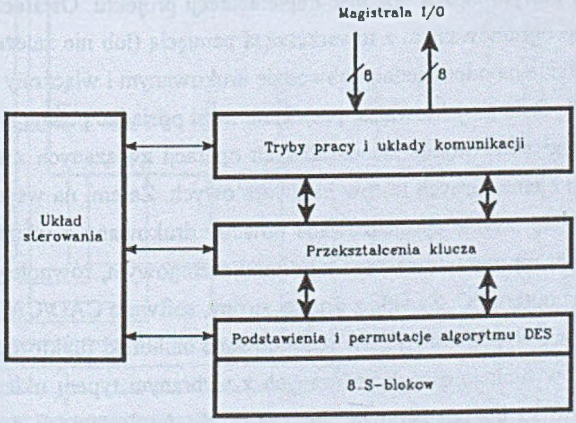
Jedno z pierwszych rozwiązań (z roku 1981) kryptograficznego układu scalonego w zastosowaniu do telekomunikacji i szyfracji danych na dysku, realizujące sprzętowo algorytm Data Encryption Standard (DES) przez układ wykonany w Advanced Micro Devices [7, 12]

na bazie technologii nMOS sterowany standardowym mikroprocesorem, pozwalało szyfrować potokowo 8-bajtowe bloki z przepustowością 14 Mbit/s.



Rys. 1. Sieciowa implementacja szyfratorów liniowych Omnisec [10]
Fig. 1. Network application of serial Omnisec's [10] encoder/decoders

Nowsze rozwiązanie (z roku 1987) szyfrujące na bazie układu scalonego komunikującego się z magistralą przedstawiono na rys. 2.



Rys. 2. Schemat blokowy układu ASIC CRYPTECH [22] implementującego DES
Fig. 2. Block diagram of the DES chip implemented by CRYPTECH [22]

Układ ten wykonany w firmie CRYPTECH [22] w technologii CMOS, realizował algorytm DES z przepustowością 32 Mbit/s.

Do najszybszych, o rozbudowanych funkcjach układów realizujących algorytm DES, należą zbudowany w 1992 roku w DEC Corp. [4] układ pracujący z przepustowością 1 Gbit/s wykonany w technologii GaAs oraz wykonany w technologii CMOS w Analog Devices [2] w roku 2000, układ ADSP-2141, pracujący przy przepustowości 640 Mbit/s (lub 214 Mbit/s dla trybu pracy z potrójnym DES'em).

W implementacji układów reprogramowalnych opracowano elementy biblioteczne typu CORE [8, 15, 18, 28], które mogą być wykorzystane do realizacji wybranych algorytmów szyfracji w strukturach FPGA.

3. Systemy implementacyjne CAD/CAE dla FPGA

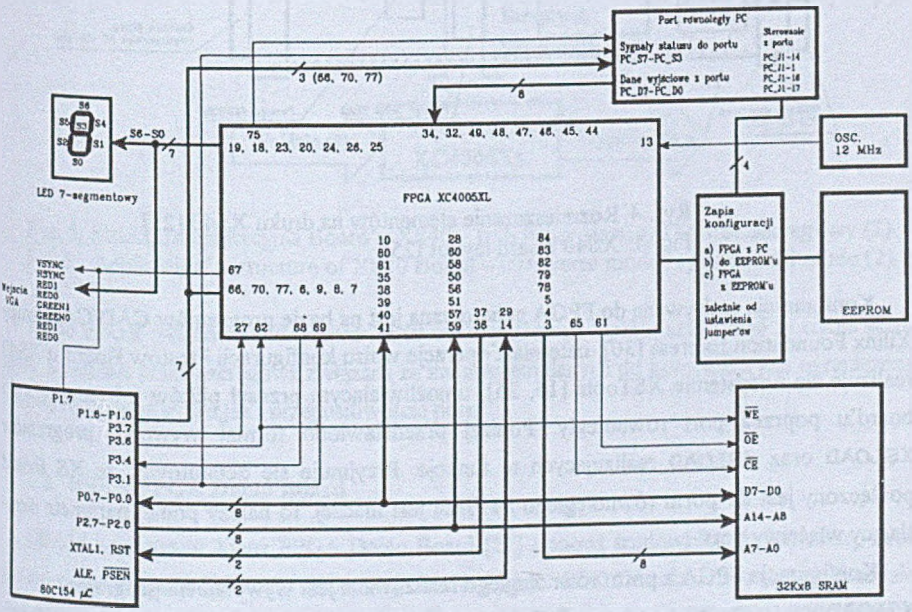
Szybkie wykonywanie prototypów układów logicznych realizowanych w strukturach reprogramowalnych wymaga (oprócz programowych narzędzi wspomagania projektowania, na które składają się moduły programowe wprowadzania danych o projekcie w postaci schematu układu, opisu graficznego w formie maszyny stanów FSM (Finite State Machine) lub opisu formalnego w języku opisu sprzętu HDL (Hardware Description Language), moduły symulatorów różnych typów, moduły syntezy logicznej, moduły rozmieszczania i ciągnięcia połączeń oraz moduły generacji odpowiednio sformatowanych plików konfiguracyjnych dla wykorzystywanego układu reprogramowalnego) dysponowania odpowiednio przygotowanym środowiskiem, w którym można dokonać implementacji projektu. Ostatecznie projekt zapisany w układzie reprogramowalnym z towarzyszącą pamięcią (lub nie zależnie od typu układu) umieszczony zostanie na odpowiednim obwodzie drukowanym i włączony w hardware aplikacji, tym niemniej w fazie projektowania, projektant musi posiadać pewną swobodę uwalniającą go od konieczności realizowania standardowych operacji związanych z obwodami downloadingu i realizacji elementarnych testów hardware'owych. Zatem, na wspomniane wyżej środowisko składają się zwykle specjalizowane obwody drukowane z wbudowanymi układami komunikacji układu reprogramowalnego z portami szeregowym, równoległym lub odpowiednią magistralą komputera PC. Zwykle z drugiej strony, software CAD/CAE systemu wspomagania projektowania wyposażony jest w rozbudowane biblioteki funktorów logicznych, a nawet dużych bloków funkcjonalnych związanych z wybranym typem układu reprogramowalnego. Biblioteki te są bardzo istotnym elementem dla implementacji projektu, bowiem są optymalizowane w sensie dynamiki i wykorzystania powierzchni, szczególnie w przypadku zautomatyzowanych narzędzi syntezy i rozmieszczania. Wykorzystanie takich bibliotek po-

zwala zwykle przyspieszać pracę układu i minimalizować wykorzystanie elementarnych modułów CLB (Configurable Logic Block) związanych z architekturą układu reprogramowalnego, co jest krytyczne w przypadku realizacji dużych projektów.

W dalszym ciągu przedstawiono dwa rozwiązania pozwalające na realizację projektów zbudowanych na układach reprogramowalnych do końca, czyli do uzyskania implementacji projektu, którą można natychmiast konfigurować i hardware'owo testować; jedno z tych rozwiązań zawiera układ FPGA, który może współpracować z wbudowanym na druku mikrokontrolerem typu 80C51, pamięciami EEPROM i RAM przy wpisywaniu konfiguracji z portu równoległego komputera PC (możliwości układu przy tej konfiguracji są zresztą szersze) bez lutowania, drugi natomiast stanowi typowy układ szybkiego prototypowania zawierający dwa układy FPGA, które mogą pracować niezależnie lub ze wzajemną wymianą danych, natomiast ich konfiguracje można wpisywać z portów komputera PC, pamięci ROM wbudowanych na obwodzie drukowanym lub w strukturze łańcuchowej układów FPGA.

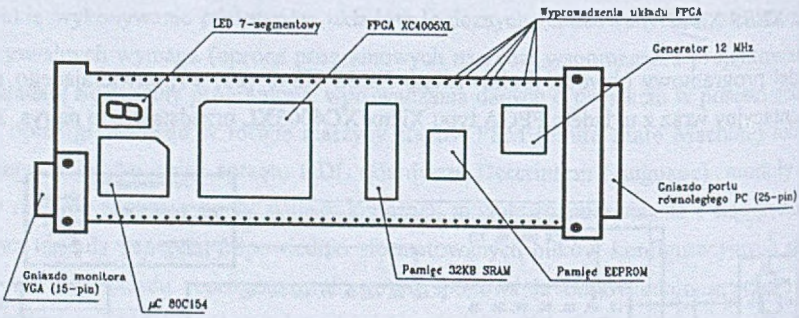
3.1. XESS XS40

Model programowy obwodu drukowanego XESS XS40 [23, 24] zawierającego system implementacyjny wraz z układem FPGA typu Xilinx XC4005XL przedstawiono na rys. 3.



Rys. 3. Model programowy Board'u XS40 [23, 24]
Fig. 3. XS40 Board programmer's model [23, 24]

Układ XC4005XL [20] pozwala na realizację projektów wielkości 5000 ekwiwalentnych bramek logicznych, a projektant dysponuje 196 blokami CLB. Wpis konfiguracji FPGA można realizować z portu równoległego komputera PC lub pamięci EEPROM, przy czym pamięć zapisać można także z portu równoległego. Odpowiednie tryby pracy realizowane są za pomocą jumper'ów (rys. 3). Wybrane pady FPGA połączono z wyjściami, które stanowią 7-segmentowy wyświetlacz LED i łączówka wejścia monitora (możliwość synchronizacji i sterowania kolorem – wymaga to realizacji odpowiedniej logiki w FPGA, drabinek rezystorów na rys. 3 nie przedstawiono). Board XS40 zawiera mikrokontroler i pamięć 32kB RAM, co przy prostej możliwości zapisu pamięci z portu równoległego komputera PC pozwala na bazie FPGA budować reprogramowalne systemy mikroprocesorowe. Dane z systemu XS40 do komputera PC można zwrotnie czytać poprzez 4 linie portów mikrokontrolera (trzy linie z P1 oraz jedna linia z P3). Rozmieszczenie elementów i wyprowadzenia zewnętrzne Board'u XS40 przedstawiono na rys. 4.



Rys. 4. Rozmieszczenie elementów na druku XS40 [25]
Fig. 4. XS40 Board layout [25]

Konfiguracja wpisywana do FPGA generowana jest na bazie programów CAD/CAE grupy Xilinx Foundation Express [30], natomiast operacje wpisu konfiguracji i testów Board'u XS40 realizuje się w systemie XSTools [16, 25], umożliwiającym przesył plików z komputera do board'u poprzez port równoległy. Poniżej przedstawiono format wywołań programów XSLOAD oraz XSLOAD realizujących te funkcje. Przyjmuje się defaultowo, że XS Board podłączony jest do portu równoległego #1. Jeśli jest inaczej, to należy podać parametr określający właściwy port.

Konfiguracja FPGA z portu równoległego realizowana jest wywołaniem programu XSLOAD:

```
XSLOAD CIRCUIT.BIT
```

gdzie `CIRCUIT.BIT` jest plikiem konfiguracji utworzonym przez program Xilinx Foundation dla elementu serii XC4000.

Zapis konfiguracji FPGA do EEPROM'u:

```
XSLOAD -SERIAL_EEPROM CIRCUIT.BIT
```

Przy zapisie konfiguracji FPGA do EEPROM'u i downloadowaniu z EEPROM'u, ustawienie jumperów należy zrealizować zgodnie z [25].

Konfiguracja FPGA z portu równoległego i zapis pamięci RAM board'u XS40:

```
XSLOAD FILE.HEX CIRCUIT.BIT
```

gdzie `FILE.HEX` jest plikiem, którego zawartość wpisywana jest do pamięci.

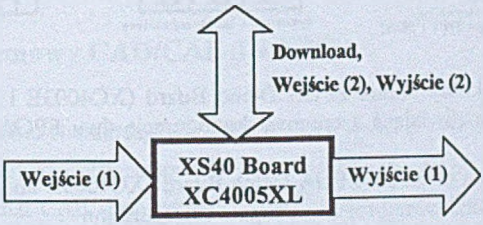
Testowanie projektu w FPGA z portu równoległego realizowane jest programem XSPORT:

$$\texttt{XSPORT}\ b_7b_6b_5b_4b_3b_2b_1b_0$$

przy następującym przyporządkowaniu bitów b_i pinom układu XC4005XL:

$$b_0 - 44,\ b_1 - 45,\ b_2 - 46,\ b_3 - 47,$$
$$b_4 - 48,\ b_5 - 49,\ b_6 - 32,\ b_7 - 34.$$

Struktura aplikacyjna Board'u XS40 przedstawiona została na rys. 5.



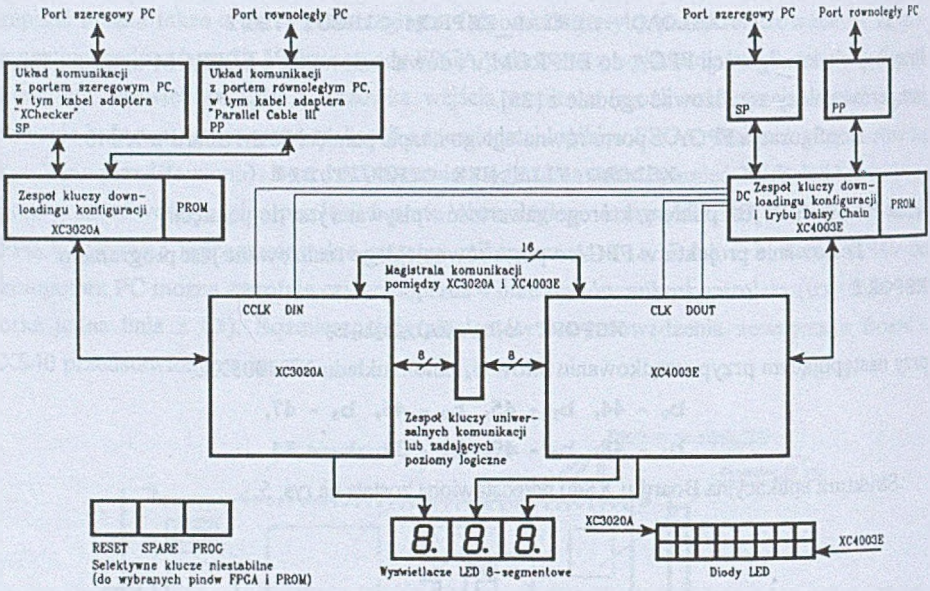
Rys. 5. Struktura aplikacyjna Board'u XS40 – tryb pracy skrośny (1) oraz szeregowy (2)
Fig. 5. Application's structure of XS40 Board – transverse mode (1) and serial mode (2)

Układ FPGA wbudowany w Board XS40 może pracować z pełną szybkością w trybie skrośnym, natomiast praca szeregowa związana ze zwracaniem danych do komputera jest ograniczona zarówno długością słowa,jak i przepustowością portu.

3.2. Xilinx FPGA Demo Board

Obwód drukowany Xilinx FPGA Demo Board [27] stanowi środowisko implementacji projektów w układy serii XC3000 i XC4000 Xilinx'a. Układ XC3020A zamontowany na boardzie charakteryzuje się 1500 ekwiwalentnych bramek przy 64 CLB, natomiast XC4003E 3000 bramek lub 100 elementami CLB. Struktura systemu nie zawiera procesora, stanowi więc typowy układ

programowania układu FPGA dla szybkiego prototypowania i weryfikacji. Schemat blokowy Demo Board'u przedstawiono na rys. 6.



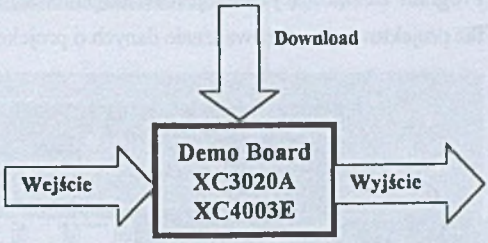
Rys. 6. Schemat blokowy Xilinx FPGA Demo Board (XC4003E i XC3020A), DC – część połączeń umożliwiająca szeregową konfigurację dwu FPGA z jednego portu (Daisy Chain)

Fig. 6. Block diagram of Xilinx FPGA Demo Board (XC4003E and XC3020A), DC – Daisy Chain additional wiring for one PC port configuration

Pliki konfiguracji projektu wpisywanego do układów tworzy się standardowym narzędziem CAD/CAE Xilinx'a [30], natomiast jest kilka możliwości realizacji wpisu konfiguracji zarówno z portów szeregowych i równoległych komputera PC, jak i pamięci PROM. Przewidziano także możliwość szeregowego konfigurowania dwu układów z jednego portu w trybie Daisy Chain. Struktury downloadingu realizowane są na bazie zespołów kluczy zestawiających odpowiednie połączenia elementów [27] (rys. 6).

Na obrzeżach układów FPGA znajdują się piny stanowiące wyprowadzenia wszystkich padów układów FPGA umożliwiające podłączenie sond i generatorów. Do wybranych padów podłączone zostały 8-segmentowe wyświetlacze LED i 8-elementowe grupy diod luminescencyjnych. Elementy te zwykle wykorzystuje się w trakcie debuggingu zaimplementowanego projektu. Możliwa jest wymiana danych pomiędzy układami FPGA oraz statyczne pobudzanie wybranych wejść poziomami logicznymi (zespół kluczy uniwersalnych).

Struktura aplikacyjna Demo Board'u przedstawiona została na rys. 7.



Rys. 7. Struktura aplikacyjna Xilinx Demo Board – praca w trybie skrośnym
Fig. 7. Application of Xilinx Demo Board in a transverse mode

Zarówno Board XS40, jak i Xilinx Demo Board, pozwalają na realizację projektów średniej wielkości. W przypadku konieczności wykorzystania większych układów (np. Xilinx XC4062XL z 62000 bramek lub 2304 CLB czy XC4085XL z 85000 bramek lub 3136 CLB) można wykorzystać odpowiednio układy downloadingu lub komunikacji z pamięciami konfiguracji, jednakże wymaga to pewnych przeróbek na płytkach drukowanych.

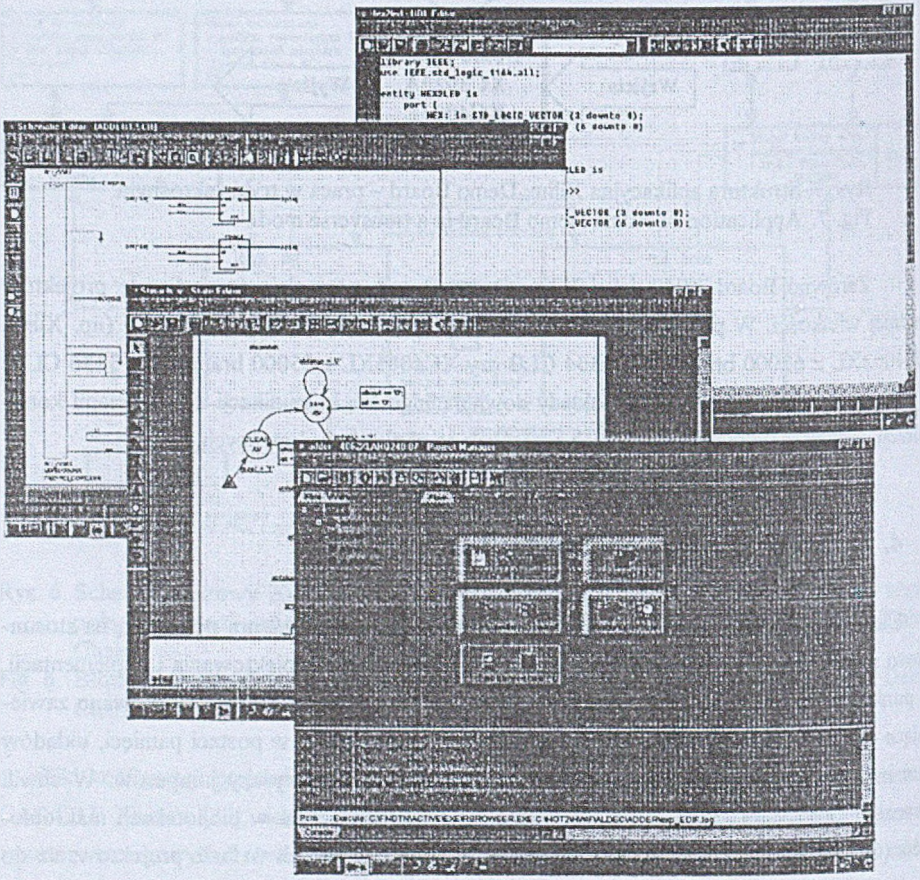
4. Software systemowy CAD/CAE dla FPGA

Układy reprogramowalne charakteryzujące się ustaloną architekturą pozwalają na stosunkowo prostą integrację narzędzi programowych wspomagania projektowania i implementacji. Z punktu widzenia narzędzi CAE sprawę ułatwiają odpowiednie obwody drukowane zawierające oprócz układów FPGA, wbudowane dodatkowe elementy w postaci pamięci, układów komunikacji z portami lub magistralami komputera, wyświetlaczy czy jumperów. W chwili obecnej można zauważyć tendencję projektowania i umieszczania w bibliotekach makrobloków (CORE) realizujących w pełni funkcje interfejsu dołączanych w fazie projektowania do projektu aplikacji i łącznie downloadowanych w układ FPGA. Software systemowy zatem zwykle powstaje przez połączenie poszczególnych narzędzi wykorzystywanych pod nadzorem programu zarządzającego ułatwiającego wykorzystanie bibliotek, tworzenie projektów hierarchicznych przy różnym tworzeniu modułów projektu (schemat, FSM, HDL) oraz weryfikację i raportowanie.

4.1. Xilinx Foundation Express

Kompletny system Xilinx Foundation Express dla projektowania i implementacji projektów w układy FPGA i CPLD (Complex Programmable Logic Devices bazujące na technologii

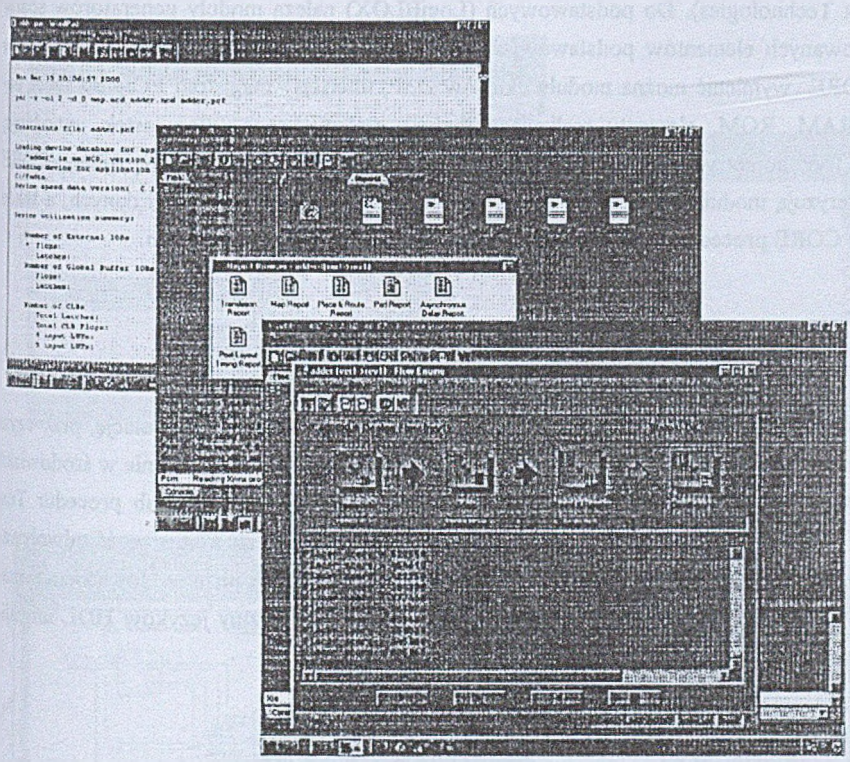
EEPROM) składa się z dwu podstawowych części: bloku wprowadzania danych dla projektu, oraz bloku implementacji. Program zarządzający – Project Manager umożliwia efektywną realizację pięciu podstawowych faz projektowania: wprowadzanie danych o projekcie (rys. 8), z możliwością



Rys. 8. Xilinx Foundation Express 2.1i – Project Manager i wprowadzanie projektu [30]
Fig. 8. Xilinx Foundation Express 2.1i – Project Manager and design entry [30]

symulacji, fazy implementacji projektu z możliwością symulacji po mapowaniu elementów, wykonaniu rozmieszczania i routingu (rys. 9) oraz fazy downloadowania. W fazie wprowadzania projektu można zapisać projekt w postaci schematu, grafu maszyny stanów FSM oraz wybranego języka opisu hardware'u: VHDL, Verilog lub Abel6.

W fazie implementacji dla zadeklarowanego typu elementu realizowana jest synteza układu logicznego (syntezator Synopsys), rozmieszczanie komórek, routing, a także zbierane są dane



Rys. 9. Xilinx Foundation Express 2.1i – struktura realizacji projektu [30]
Fig. 9. Xilinx Foundation Express 2.1i – Flow Engine (translate, map, place&route, configure) [30]

dla raportów związane z analizą statyczną timingu, dane dla symulacji uwzględniającej opóźnienia routingu oraz dane o wykorzystaniu struktury FPGA. Przykładowo, raport z rys. 9 dotyczy wykorzystania elementu XC4062 do realizacji sumatora 16-bitowego współpracującego poprzez element LogiCORE PCI_CORE z magistralą PCI (fragment):

```
Map report:     Number of CLB:            624 out of 2304      27%
                Number of bounded IOBs:   174 out of 193       90%
                Total equivalent gates:   10956        (dla XC4062 – 62000).
```

4.2. Biblioteki (Xilinx)

Biblioteki makrobloków, które mogą być dołączane do realizowanych projektów w ramach systemu Xilinx Foundation Express, są znacznie rozbudowane [26, 28, 29] zarówno na bazie elementów Xilinx'a, jak też jego partnerów (np. Mentor Graphics, CAST, MDS,

Phoenix Technologies). Do podstawowych (LogiBLOX) należą moduły generatorów sparametryzowanych elementów podstawowych, jak: liczniki, rejestry i multipleksery. W grupie LogiCORE wymienić można moduły układów DSP, interfejsy magistrali PCI, elementy pamięci RAM, ROM, elementy realizujące funkcje matematyczne (pierwiastek, mnożenie, funkcje trygonometryczne) oraz szybkie układu LUT (Look-Up Table). Grupę AllianceCORE charakteryzują moduły interfejsu w tym USB i PCMCIA, moduły transmisji danych, a także moduły CORE procesorów (w tym RISC) i układów z nimi współpracujących.

5. Weryfikacja formalnego opisu projektów

Weryfikacja projektu przed implementacją jest realizowana poprzez symulację, przy czym w zależności od skali realizowanych testów wystarczające może być testowanie w środowisku symulatora na bazie wektorów testujących deklarowanych bezpośrednio lub procedur Test Bench, natomiast w przypadkach bardziej złożonych może zachodzić konieczność odwoływania się do wektorów testowych i poprawnych wyników zapisanych poza środowiskiem symulatora, np. w plikach znajdujących się na dysku. Zwykle formalizmy języków HDL umożliwiają takie operacje.

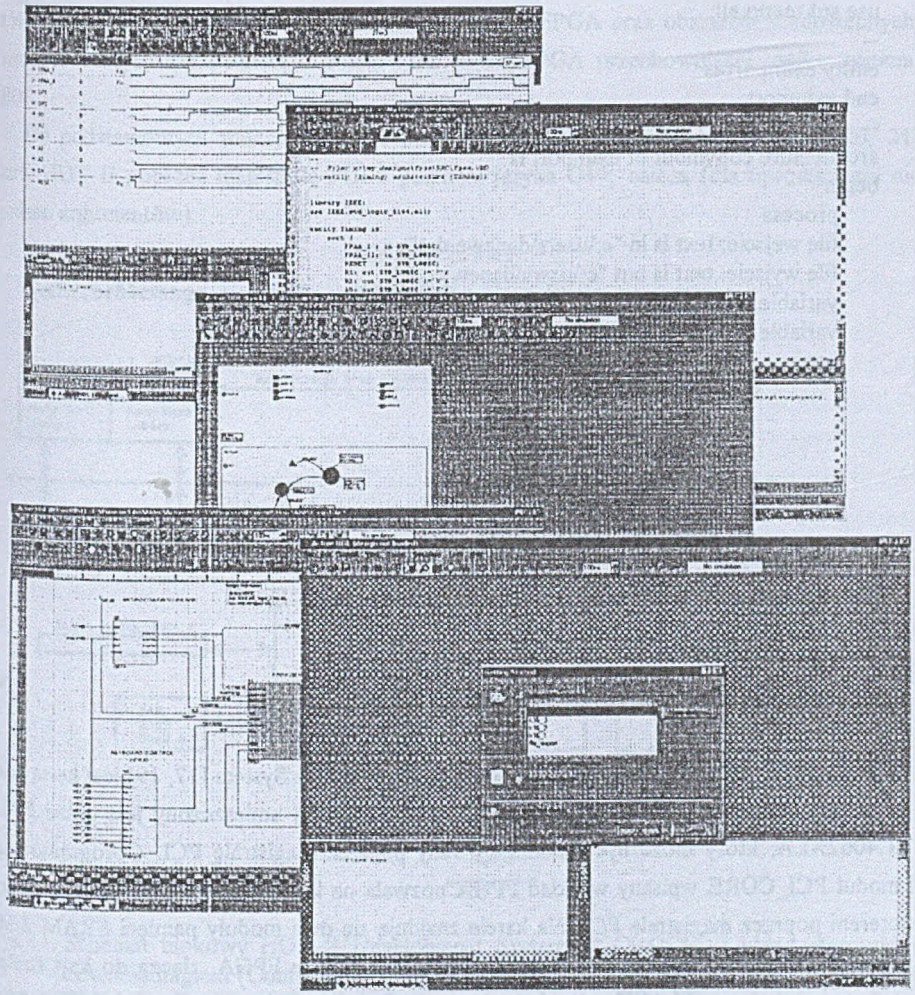
5.1. Active-HDL

System oprogramowania Active-HDL [1] stanowi zintegrowane środowisko (rys. 10) symulatora realizującego funkcje, które można scharakteryzować następująco:

1. Wprowadzanie projektu w postaci schematu, HDL, FSM oraz hierarchii.
2. Zarządzanie projektami i bibliotekami.
3. Interaktywna symulacja behavioralna i strukturalna.
4. Generacja Test Bench.
5. Interfejsy do narzędzi CAD/CAE syntezy logicznej oraz rozmieszczania i rutingu innych producentów software'u EDA.

5.2. Testy dyskowe

Zgodnie z normą języka VHDL [6], w środowisku Active-HDL można zrealizować komunikację pomiędzy projektem zapisanym w VHDL'u a danymi wektorami testującymi zapisanymi w plikach znajdujących się na dysku. Poniżej, w przedstawionym listingu, w czasie symulacji, moduł VHDL nazwany dalej eximport.VHD czyta z dysku z pliku danewe.dat, czterobitowy wektor do zmiennej w VHDL nazwanej wektor, a więc udostępnia zmiennej wartość za-



Rys. 10. Active-HDL V. 3.5 – zarządzanie i wprowadzanie projektu, wyniki symulacji [1]
Fig. 10. Active-HDL V. 3.5 – Design Explorer, design data entry, and simulation [1]

pisaną na dysku, a następnie zapisuje tę wartość na dysk do pliku danewy.dat, a więc wyprowadza wartość ze zmiennej w module VHDL na dysk. Operacja zależy od tego, co będzie w ciele procesu (w tym przypadku jest tylko pętla loop i kopiowanie dysk-dysk przez VHDL kończy się z chwilą odczytu końca pliku danewe.dat).

```
--
-- File: c:\my designs\file_export\SRC\eximport.VHD
--{entity {eximport} architecture {copyinout}}
```

```
--
use std.textio.all;
--
entity eximport is
end eximport;
--
architecture copyinout of eximport is
begin
    process
    file wejscie: text is in "c:\users\danewe.dat";
    file wyjscie: text is out "c:\users\danewy.dat";
    variable linia1, linia2: line;
    variable wektor: bit_vector (3 downto 0);
    begin
        while not (endfile (wejscie)) loop
            readline (wejscie, linia1);
            read (linia1, wektor);
            write (linia2, wektor);
            writeline (wyjscie, linia2);
        end loop;
    end process;
end copyinout;
```

6. HOT II Hardware Object Technology

HOT II XL (Virtual Computer Corporation) Development System [17, 19] jest kartą komputera PC komunikującą się z magistralą PCI. Na karcie tej umieszczony jest układ FPGA XC4062XLA, który może być rekonfigurowany poprzez magistralę PCI. Oprogramowanie i moduł PCI_CORE wpisany w układ FPGA pozwala na komunikowanie się układu z komputerem poprzez magistralę PCI. Na karcie znajdują się dwa moduły pamięci SRAM 2 MB połączone niezależnymi magistralami 32-bitowymi z układem FPGA, złącza do kart rozszerzeń, pamięć typu FLASH przechowująca konfigurację PCI_CORE oraz układ HOS (Hardware Operating System) sterujący komunikacją z otoczeniem dla startu układu po włączeniu zasilania. Pamięć CACHE wykorzystywana jest w rekonfiguracji FPGA wraz z układem CCM zarządzającym downloadingiem. Schemat blokowy karty HOT II DS VCC przedstawiono na rys. 11.

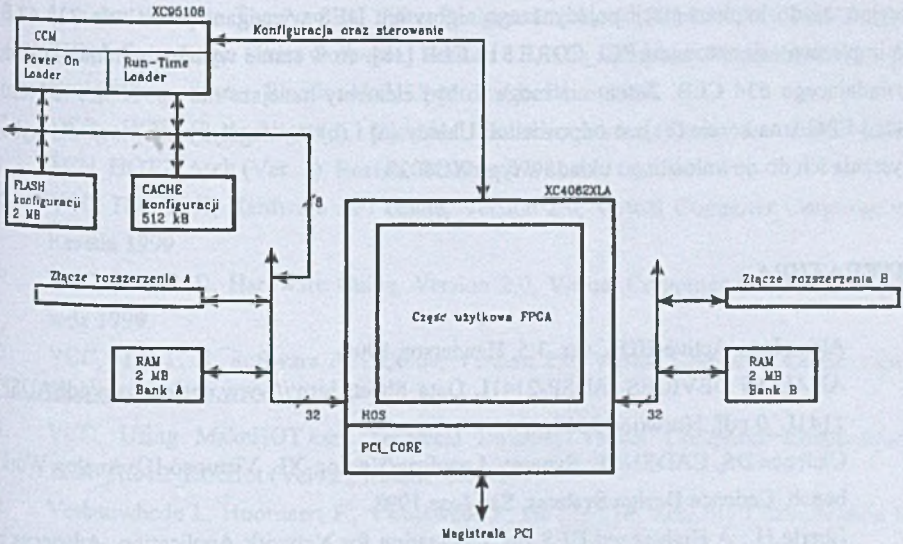
Dla przygotowanego pliku konfiguracyjnego (np. programem Xilinx Foundation Express), który ma w FPGA realizować projekt, programem makeHOT.exe generuje się zbiór plików, które umożliwiają pracę karty w trybie testowania lub z poziomu programu napisanego w języku C++. Opracowana została biblioteka funkcji sterujących podstawowymi operacjami karty HOT II DS. Funkcje te włączane są w kod aplikacji napisanej w C++. Dostęp do elementów

karty został zrealizowany na zasadzie podziału adresów pomiędzy elementy pamięci dwu banków, pamięci układu CCM, wydzielonego obszaru dla FPGA oraz obszarów niedostępnych, zarezerwowanych dla pamięci CACHE oraz części FPGA przechowującej logikę systemu HOS.

Do podstawowych funkcji, które realizują operacje sterowania i komunikacji [20, 21] karty HOT II poprzez magistralę PCI z poziomu języka C++, należą (dla uproszczenia nie podano argumentów):

Hot2 - inicjalizacja karty,

GetErrorMessage - informacja o błędach,



Rys. 11. Schemat blokowy HOT II Development System VCC: CCM – układ sterowania downloadingiem (Configuration Cache Manager), HOS – interfejs części użytkowej FPGA do elementów HOT II DS (Hardware Operating System)

Fig. 11. Block diagram of HOT II Development System VCC: CCM – Configuration Cache Manager, HOS – VCC's HOT II DS Hardware Operating System

GetCompType - identyfikacja układu FPGA,

LoadConfig - zapisz konfigurację z pliku do pamięci,

LoadCache - zapisz konfigurację z pamięci do CACHE,

RtrCache - rekonfiguruj FPGA,

Reset - resetuj kartę,

Write - wpisz dane do karty (w tym także burst mode),

Read - czytaj dane z karty (w tym także burst mode),
SetClockFrequency - ustaw moduł zegara.

7. Wnioski

Rozpatrzone systemy (a) XESS XS40 Board V1.2, (b) Xilinx FPGA Demo Board oraz (c) VCC HOT II Development System umożliwiają efektywną implementację projektów cyfrowych w struktury FPGA. Dla układów kryptograficznych jednakże praca układu (a) w trybie pracy szeregowej z mikrokontrolerem jest za wolna. Z drugiej strony na podstawie [8] można przyjąć, że do implementacji pojedynczego algorytmu DES wymagane jest około 316 CLB, dla implementacji natomiast PCI_CORE 318 CLB [18], co w sumie wymaga minimum układu posiadającego 634 CLB. Zatem nie mogą to być elementy mniejsze niż np. Xilinx XC4020. Układ FPGA na karcie (c) jest odpowiedni. Układy (a) i (b) wymagałyby adaptacji dla wykorzystania ich do downloadingu układów typu XC4020.

LITERATURA

1. Aldec Inc., Active-HDL, ver. 3.5, Henderson 1998.
2. ANALOG DEVICES, ADSP-2141L Data Sheet, http://www.analog.com/pdf/ADSP-2141L_0.pdf, Norwood 2000.
3. Cadence DS, CADENCE: Synergy, Leapfrog/Verilog-XL, Virtuoso IC, Analog Workbench, Cadence Design Systems, San Jose 1993.
4. Eberle H.: A High-speed DES Implementation for Network Application, Advances in Cryptology – CRYPTO '92 Proceedings, Springer-Verlag, pp. 521-539, Berlin 1992.
5. EUROPRACTICE Software Support Services (Academic), Rutherford Appleton Laboratory, RAL, Oxfordshire, October 1995.
6. Institute of Electrical and Electronics Engineers, IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-1993, New York, (June), 1994.
7. MacMillan D.: Single Chip Encrypts Data at 14 Mb/s, Electronics, Vol. 54, No. 12, June 16, pp. 161-165, 1981.
8. MDS, XF-DES Data Encryption Standard Engine Core, Alliance CORE, Product Specification, Memec Design Services, Mesa, AZ, November 1998.
9. Mentor Graphics' Products, Wilsonville, http://www.mentorg.com/
10. OPTIMUS S.A., Rodzina szyfratorów liniowych, OPTIMUS S.A. Oddział II w Warszawie, OMNISEC AG, Szwajcaria (Folder), Warszawa 1999.

11. Pilch M., Znamirowski L.: Realizacja programu ASIC/DI w Instytucie Technologii Elektronowej i Politechnikach, ZN Pol. Śl. s. Informatyka z. 27, Gliwice 1994.
12. Schneier B.: Applied Cryptography, Protocols, Algorithms, and Source Code in C, John Wiley & Sons, New York 1994.
13. Tanner Tools Pro, Tanner Research, rev.5/96, Pasadena 1996.
14. Tomovich C.: MOSIS User Manual, Release 3.1, Information Science Institute, University of Southern California, Marina del Rey 1988.
15. Trybicka K., Ziębiński A.: Sprzętowa realizacja algorytmu szyfrującego DES w strukturę FPGA z wykorzystaniem aplikacji Active-CAD, Materiały I Krajowej Konf. Nauk., "Reprogramowalne Układy Cyfrowe", Szczecin, 12-13 marca 1998.
16. Van den Bout D.: The Practical Xilinx Designer Lab Book, XILINX STUDENT EDITION, Foundation Series Software, Version 1.3, The Complete Programmable Logic Design Environment, Prentice Hall, Upper Saddle River 1998.
17. VCC, HOT II Architecture, Technical Bulletin, Virtual Computer Corporation, TCN_HOT2_Arch (Ver. 1), Reseda, October 1998.
18. VCC, H.O.T. II, Hardware API Guide, Version 2.0, Virtual Computer Corporation, Reseda 1999.
19. VCC, H.O.T. II, Hardware Guide, Version 2.0, Virtual Computer Corporation, Reseda 1999.
20. VCC, H.O.T. II, Software API Guide, Version 2.0, Virtual Computer Corporation, Reseda 1999.
21. VCC, Using MakeHOT.exe, Technical Bulletin, Virtual Computer Corporation, TCN_HOT2_MKHot (Ver. 2), Reseda, May 1999.
22. Verbauwhede I., Hoornaert F., Vandewalle J., De Man H.: Security Consideration in the Design and Implementation of a New DES Chip, Advances in Cryptology – EUROCRYPT '87 Proceedings, Springer-Verlag, pp. 287-300, Berlin 1987.
23. XESS Corporation, XS Board Schematic Version 1.2, X Engineering Software Systems (XESS), http://www.xess.com/FPGA/schematics.pdf, Raleigh 1998.
24. XESS Corporation, XS40 FPGA Board Programmer's Model, X Engineering Software Systems (XESS), http://www.xess.com/FPGA/prgmdl40-v1_2.pdf, Raleigh 1999.
25. XESS Corporation, XS40, XSP, and XS95 Board Manual, X Engineering Software Systems (XESS), http://www.xess.com/FPGA/xs40-manual-v1_1.pdf, Raleigh 1998.
26. Xilinx Inc., CORE Solutions, San Jose 1998.
27. Xilinx Inc., Hardware User Guide, San Jose 1997.
28. Xilinx Inc., LogiCORE PCI Data Book, San Jose 1997.

29. Xilinx Inc., The Programmable Logic Data Book, San Jose 1998.
30. Xilinx Inc., Xilinx Foundation Series, version 2.1i, San Jose 1999.
31. Znamirowski L., Ziębiński A.: Stanowisko badawcze CAD/CAE do implementacji sprzętowej szyfracji plików, T. III, PC KBN nr 8T11C 026 98C/4258: "Bezpieczeństwo systemów komputerowych", Instytut Informatyki, Politechnika Śląska, Gliwice 1999.
32. Znamirowski L., Ziębiński A.: Projekt rozwiązania sprzętowego szyfratora i deszyfratora opartego o matryce programowalne FPGA, T. IV, PC KBN nr 8T11C 026 98C/4258: "Bezpieczeństwo systemów komputerowych", Instytut Informatyki, Politechnika Śląska, Gliwice 1999.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 28 kwietnia 2000 r.

Abstract

The paper presents a selected group of software and hardware solutions capable to prepare and implement the configuration data stream into a programmable integrated circuits FPGA. The features and possibilities of the evaluation boards (Figs. 3, 6, and 11) carrying the FPGA IC applicable to testing the implemented design is reported. The characterization of the CAD/CAE tools carried out in reference to the hardware implementation of DES encryption procedure has shown, that the number of CLBs in the FPGA placed on boards presented in Figs. 3 and 6 is insufficient. Estimated number of CLBs for single DES with PCI CORE is approximately equal 634, and does not overstep 27% of CLBs accessible in the Development System presented in Fig. 11.