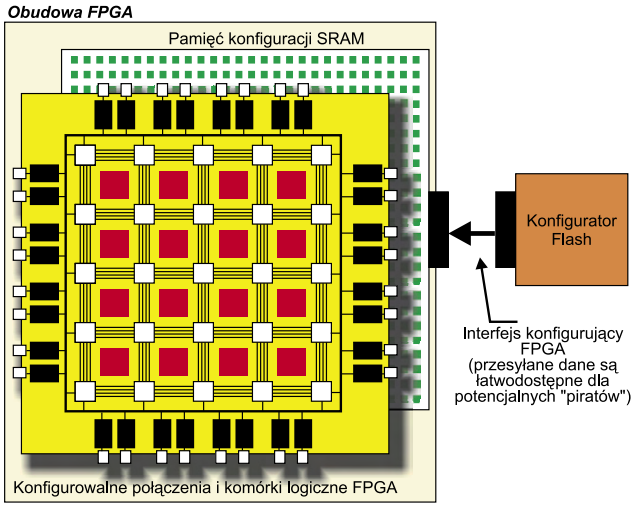
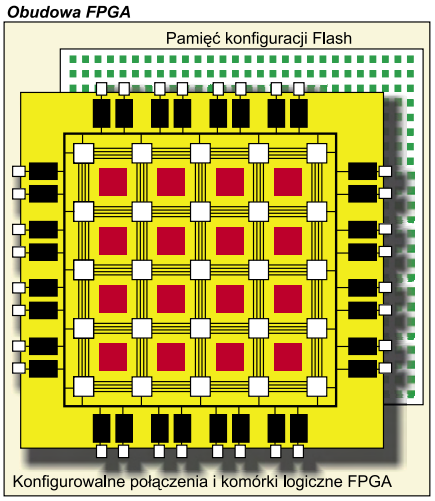
Bezpieczeństwo projektów implementowanych w układach FPGA firmy Xilinx

Ochrona projektów implementowanych w FPGA nabiera coraz większego znaczenia, co owocuje powstawaniem różnych mitów. W artykule przedstawimy inżynierskie realia na przykładzie układów FPGA firmy Xilinx.

Firma Xilinx jest twórcą koncepcji układów FPGA, które od zarania ich dziejów były wykonywane w technologii SRAM (z natury rzeczy są to pamięci ulotne). Takie rozwiązanie ma



Rys. 1. Układy FPGA z pamięcią konfiguracji typ SRAM każdorazowo po włączeniu zasilania kopiują konfigurację z zewnętrznej pamięci nieulotnej



Rys. 2. Niewiele typów FPGA jest wyposażonych w nieulotną pamięć konfiguracji

W artykule skupiamy się wyłącznie na układach FPGA firmy Xilinx. Nieco szerzej sprawę zabezpieczania własności intelektualnej twórców projektów dla układów FPGA naświetlimy w listopadowym wydaniu EP.

niepodważalne zalety, między innymi umożliwia produkowanie układów o dużych zasobach logicznych w przystępnych cenach, ale – niestety – nie jest pozbawione wad. Najpoważniejszą z nich jest konieczność stosowania zewnętrznych pamięci nielotnych, w których jest przechowywana konfiguracja wszystkich komórek FPGA. Zawartość tej pamięci musi być każdorazowo po włączeniu zasilania kopiowana do pamięci konfiguracji (SRAM) układu FPGA (**rys. 1**). Takie rozwiązanie ma dwie (choć to nie wszystkie) oczywiste wady:

- podnosi koszt wykonania urządzenia, bowiem zewnętrzne pamięci nieulotne wykorzystywane do konfigurowania FPGA są relatywnie niezbyt tanie,
- dane wykorzystywane do konfigurowania FPGA można łatwo skopiować co powoduje, że bezpieczeństwo projektu jest bardzo małe, o czym świadczy zakwalifikowanie przez NIST (amerykański *National Institute of Standards and Technology*) takich rozwiązań do kategorii *Security Level 1*, co w praktyce oznacza brak ochrony.

W naszych rozważaniach nie bierzemy pod uwagę możliwości „podglądania" projektów za pomocą klasycznych procedur *reverse–engineering*, czyli szlifowania struktur i ich analizy. W krajowych warunkach korzystanie z tak zaawansowanych metod kopiowania projektów jest mało prawdopodobne, ale trzeba sobie zdawać sprawę, że możliwe. Jedynym ograniczeniem są możliwości finansowe zleceniodawcy.

Jak sobie z tym problemem poradzić?

Najprościej czyli najtrudniej

Z pewnością Czytelnicy mający doświadczenia z mikrokontrolerami i układami CPLD odpowiedzą, że dobrym rozwiązaniem problemu bezpieczeństwa byłoby zastąpienie matrycy SRAM pamięcią nieulotną, najlepiej gdyby była to pamięć reprogramowalna (Flash lub EEPROM), jak to pokazano na **rys. 2**. Niestety z powodu ograniczeń technologicznych (przekładających się na cenę struktury układu) nie jest możliwe (*de facto* nieopłacalne) wyprodukowanie układów FPGA o dużych zasobach logicznych w technologii Flash, bowiem powierzchnia tak wykonanej struktury niezbędna do wykonania układu programowalnego byłaby bardzo duża. Z tego między innymi wynikają dość wysokie ceny układów CPLD o liczbie mikrokomórek powyżej 80…100 – płacimy głównie za rozrzutne zagospodarowanie powierzchni krzemu.

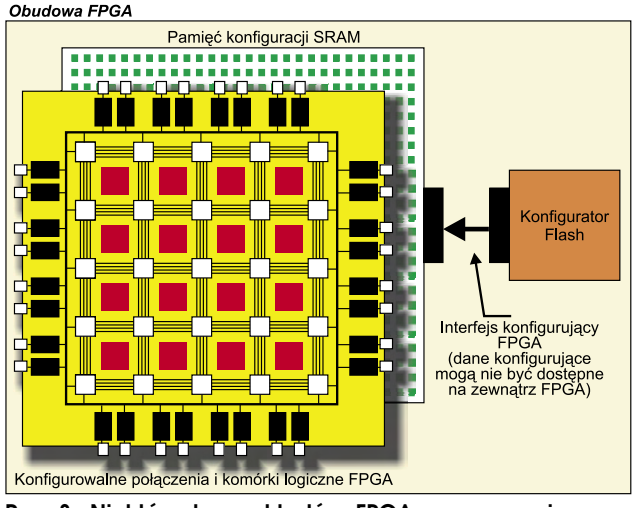
Jak zatem wybrnąć z sytuacji?

Flash do środka

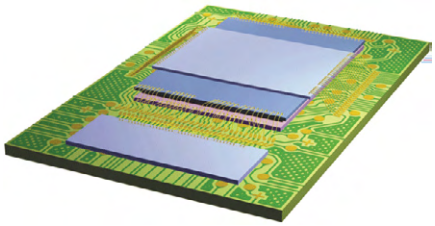
Alternatywnym i niemal równie skutecznym – jak „osadzenie" martycy konfigurowalnej na pamięci Flash – rozwiązaniem rozważanego problemu jest zintegrowanie pamięci konfigurującej (zazwyczaj nieulotnej pamięci z interfejsem szeregowym) w jednej obudowie ze strukturą FPGA. Ponieważ nie jest możliwe tanie połączenie „gęstej" technologii SRAM z technologią Flash w jednej strukturze, producenci dość często wybierają inne rozwiązanie: w jednej obudowie łączą ze sobą wyprodukowane niezależnie struktury układu FPGA i pamięci Flash (**rys. 3**). Taka technologia jest nazywana MCP (*Multichip Package*) lub MDP (*Multi–die Package*) – **rys. 4**.

Integracja konfiguratora i FPGA w jednej obudowie znacznie utrudnia nieuprawnione skopiowanie danych konfigurujących, które są pobierane przez FPGA z pamięci nieulotnej. W ocenie NIST tego typu rozwiązania osiągają *Security Level 3*, w niektórych przypadkach nawet *Security Level 4*.

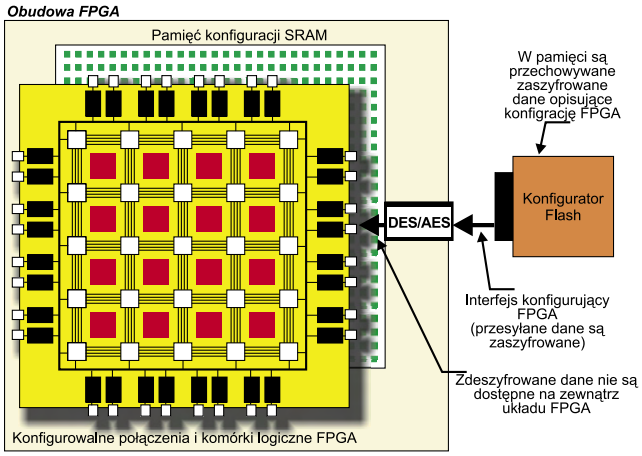
Niestety, z nieznanych przyczyn firma Xilinx nie wykorzystała możliwości zabezpieczenia własności intelektualnej (IP – *Intellectual Property*) ukrytej w projektach implementowanych w układach Spartan 3AN, które wyposażono w wewnętrzny konfigurator (z interfejsem SPI). Mechanizmy oparte na weryfikacji oryginal-



Rys. 3. Niektóre typy układów FPGA są wyposażane w konfiguratory montowane w obudowach układów jako niezależne struktury



Rys. 4. Widok struktury układu wykonanego w technologii MCP i jednocześnie MDP



Rys. 5. Droższe wersje FPGA wyposażono w moduły kryptograficzne deszyfrujące dane konfigurujące z wykorzystaniem klucza/kluczy zadanych przez konstruktora

ności pliku konfigurującego w oparciu o „numer seryjny" DNA, w jaki wyposażono układy Spartan 3AN, są niestety kłopotliwe w stosowaniu i – niestety – podatne na „złamanie".

Czy jesteśmy zatem bez szans?

Nie ma to jak kryptografia

Na szczęście nie, chociaż poprawa bezpieczeństwa kosztuje: projekty, którym trzeba zapewnić maksymalną ochronę przed kopiowaniem lub modyfikowaniem, należy implementować w układach wyposażonych w kryptograficzne systemy ochrony danych konfigurujących. Uproszczoną budowę takiego systemu pokazano na **rys. 5**.

W ofercie firmy Xilinx są dostępne układy wyposażone w deszyfratory DES3 (Virtex II/Virtex II PRO) oraz AES z 256–bitowym kluczem ustalanym przez użytkownika (Virtex 4 i Virtex 5). Wymienione układy FPGA wyposażono w dodatkową pamięć SRAM służącą do przechowywania klucza (co jest niezbędne do poprawnego zdeszyfrowania danych konfigurujących), która jest dostępna wyłącznie w trybie do zapisu poprzez interfejs JTAG. Żeby zapewnić prawidłową, automatyczną konfigurację FPGA po włączeniu zasilania, pamięć klucza trzeba zasilać za pomocą zewnętrznego ogniwa (baterii litowej). Z dodatkowego zasilania można zrezygnować, ale w takim przypadku po włączeniu zasilania i przed rozpoczęciem konfigurowania FPGA trzeba „ręcznie" wprowadzić do pamięci 256–bitowy klucz, w przeciwnym przypadku konfiguracja nie powiedzie się.

Podsumowanie

Z tego krótkiego przeglądu można wyciągnąć wniosek, że projekty implementowane w FPGA nie są bezpieczne. Sytuacja rzeczywiście nie zbyt dobra, ale w ostatnich latach ulega szybkiej poprawie: są bowiem wprowadzane na rynek układy FPGA wyposażone w coraz bardziej zaawansowane mechanizmy ochronne, których (niestety) wspólną cechą jest odczuwalny wpływ na ceny układów w nie wyposażonych. Wszystkie interesujące rozwiązania przedstawimy już za miesiąc.

Tymczasem konstruktorzy korzystający ze współczesnych, tanich wersji FPGA, mogą korzystać z alternatywnych metod ochrony swoich projektów, na przykład dwuetapową konfigurację (wprowadzenie modyfikacji–klucza do aktualnej konfiguracji pobranej z konfiguratora, za pomocą mikrokontrolera poprzez JTAG) czy też wyposażenie urządzenia w generator autoryzacji (ciągu bitów spełniających rolę klucza) wykonany na układzie CPLD lub mikrokontrolerze z zabezpieczoną pamięcią programu. Są to jednak sposoby na utrudnienie skopiowania projektu, a nie jego gwarantowaną ochronę. Warto je jednak stosować w wymagających tego przypadkach: trudności związane z szybkim skopiowaniem projektu chronionego w nietypowy sposób są często wystarczającym środkiem ochronnym.

Piotr Zbysiński, EP
piotr.zbysinski@ep.com.pl

Liczba możliwych kluczy w przypadku algorytmu AES256 (stosowany do zabezpieczenia plików konfigurujących w układach Virtex 4 i Virtex 5) wynosi aż $1{,}1 \times 10^{77}$, co praktycznie czyni niemożliwym złamanie tego zabezpieczenia.