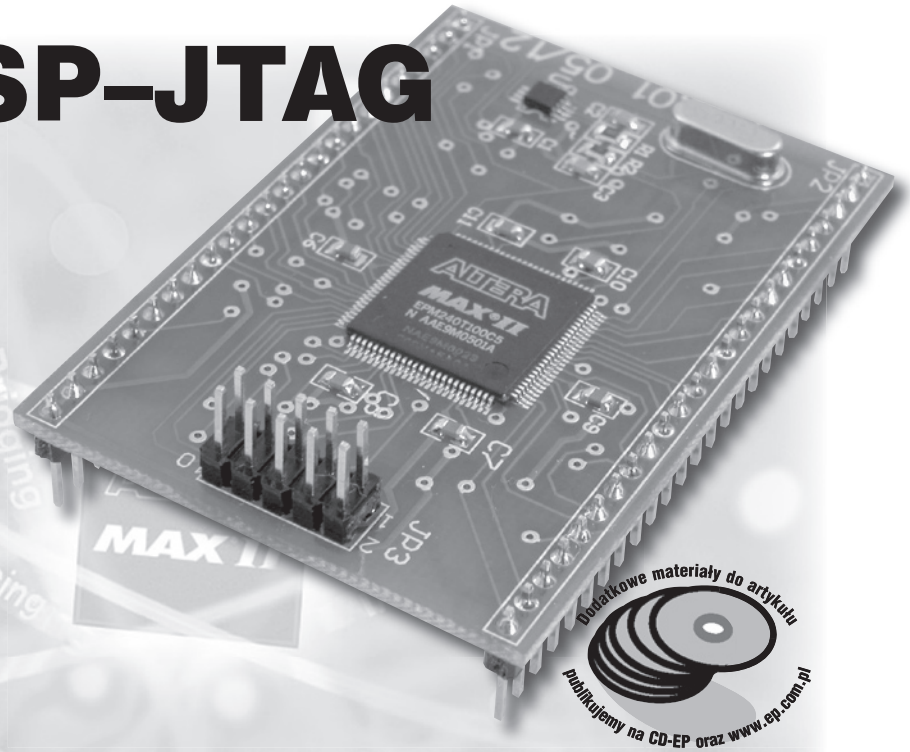
Flash z ISP–JTAG
AVT–921



Jednymi z największych zalet pamięci Flash są możliwość wygodnego, wielokrotnego reprogramowania, duża pojemność i niewysokie ceny. Połączenie tych cech spowodowało, że pamięci Flash zdobyły niezwykłą popularność aplikacyjną (zwłaszcza w przemyśle), ale mimo tego (a może właśnie dlatego) nie rozwiązano jednego problemu, który świadczy o pewnej nie–współczesności Flashy: nie ma jak ich wygodnie programować w systemie. O tym jak sobie poradzić z tym problemem, piszemy w artykule.

Rekomendacje:
projekt, który powinien zainteresować wszystkich potencjalnych użytkowników pamięci Flash, chcących zapewnić sobie możliwość szybkiego i wygodnego jej programowania w systemie.

PODSTAWOWE PARAMETRY

- Płytka o wymiarach: 43 x 62 mm
- Pojemność pamięci Flash: 32 Mb (8/16 bitów)/max. 512 Mb
- Napięcie zasilania: 3,3 V
- Emulacja pamięci Flash z interfejsem ISP
- Linie I/O zgodne z LV–TTL/LV–CMOS

Interfejs JTAG jest współcześnie fetyszem, którego posiadaniem chcą się pochwalić producenci coraz większej grupy podzespołów elektronicznych. Niestety, nie idzie w ślad za tym powszechna wiedza o rzeczywistych możliwościach tego interfejsu, co skrzętnie wykorzystują „szamani" (jak choćby niektórzy producenci mikrokontrolerów), obiecujący użytkownikom urządzeń/podzespołów z JTAG–iem rewelacje, o których komitetowi standaryzacyjnemu IEEE1149 (to ten od JTAG–a) się nie śniło. W artykule też trochę „poszamanię", przez co chcę powiedzieć, że nie zajmę się szczegółowym wyjaśnianiem co to jest JTAG i jakie ma możliwości (w EP pisaliśmy o tym kilka razy od podstaw), lecz skupię się na pokazaniu jednej z możliwości zastosowania go w codziennej praktyce.

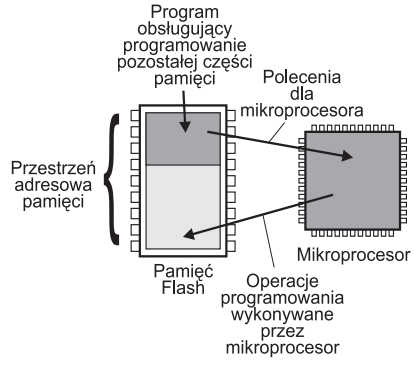
Idea

Programowanie pamięci Flash wymaga ze strony użytkownika (a w zasadzie mikrokontrolera zaprzęgnięto przez niego do pracy) wykonania kilku operacji, które uruchamiają jej wewnętrzny automat samodzielnie realizujący kolejne etapy programowania poszczególnych komórek (sektorów). Dzięki temu program użytkownika (poprzez jakąś parametryzację) nie musi szczegółowo „znać" budowy i parametrów pamięci, co stwarza dość komfortowe warunki dla programisty i otwiera drogę do zdalnego modyfikowania zawartości pamięci.

Okazuje się jednak, że w wielu systemach pamięć Flash jest traktowana wyłącznie jak wygodniejsza pamięć EPROM. Samodzielne operacje na sektorach pamięci, konieczność blokowania/odblokowywania bootsektorów, a także dobre poznanie dokumentacji pamięci przygotowanej przez producentów powodują, że „na wszelki wypadek" programy wykonywane przez mikrokontrolery nie są wyposażane w procedury zaawansowanego operowania na zawartości pamięci

Nie takie trudne

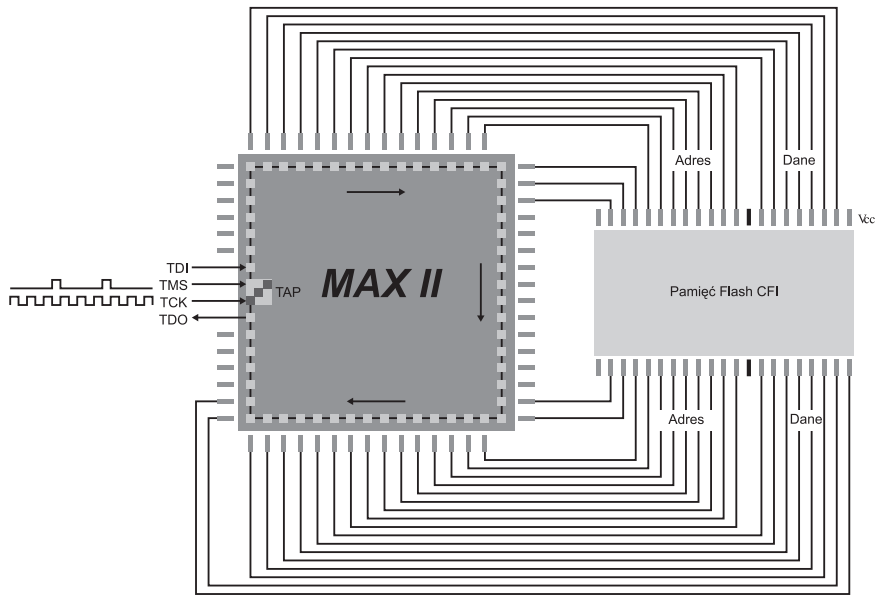
Obsługa współczesnych pamięci Flash jest wygodna i przejrzysta m.in. dzięki wprowadzeniu zestandaryzowanego „interfejsu" CFI (*Common Flash Interface*), który opisaliśmy w artykule opublikowanym w EP1/2006.



Rys. 1. Programowaniem pamięci Flash w systemie może zajmować się mikrokontroler

JTAG i Flash – to już było!

Producentem pamięci Flash wyposażonych w JTAG jest STMicroelectronics (przedstawiana na łamach EP rodzina PSD 8xx). Ich najpoważniejszą wadą jest konieczność stosowania wyspecjalizowanego programatora JTAG, nie wszystkich zadowolą także relatywnie niewielkie pojemności tych pamięci. Do niewątpliwych zalet można zaliczyć integrację w tej samej obudowie pamięci SRAM oraz niewielkich struktur PLD, które można wykorzystać m.in. do mapowania pamięci.



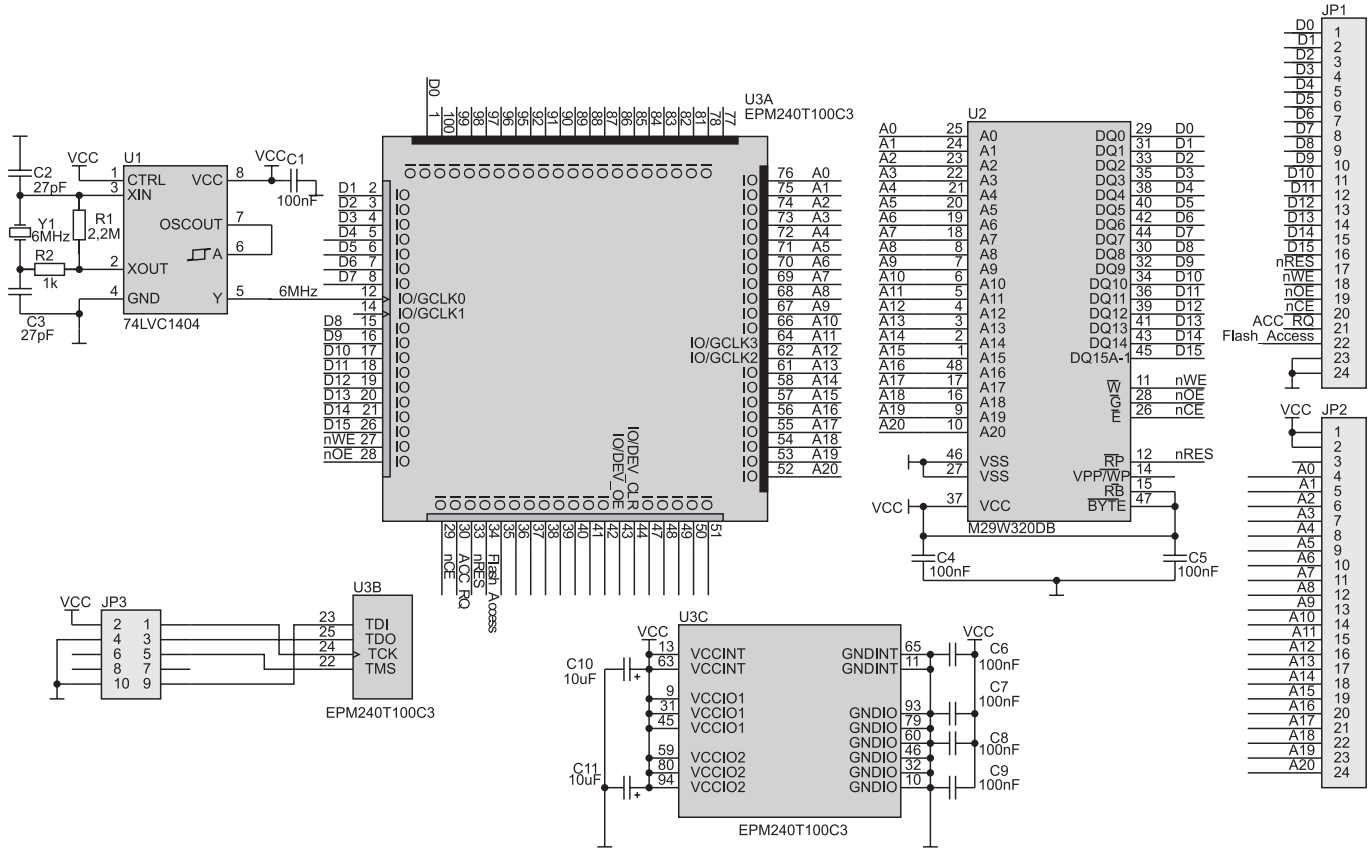
Rys. 2. Wykorzystanie JTAG-a do programowania pamięci Flash jest możliwe, ale wymaga zastosowania specjalnego adaptera

Flash. Trudność polega m.in. na tym, że program modyfikujący zawartość pamięci znajduje się w pamięci, której zawartość ma być zmieniana (**rys. 1**)... Z tymi problemami, oczywiście, sobie poradzono, ale lęki wśród konstruktorów pozostały, co często widać m.in. w korespondencji kierowanej do redakcji EP.

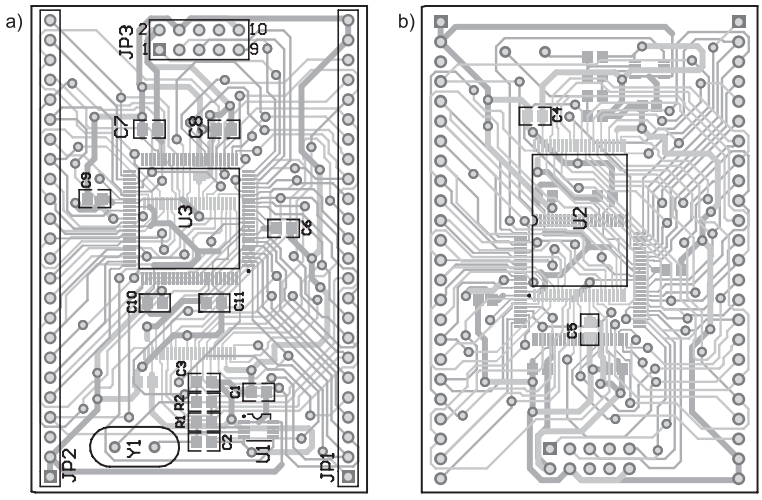
Idąc za modą (ach ten JTAG!) i jednocześnie chcąc stworzyć alternatywę dla programowania pamięci Flash w sposób klasyczny (czyli za pomocą odpowiednio zaprogramowanego mikrokontrolera lub programatora równoległego) opracowaliśmy prostą przystawkę dla większości dostępnych na rynku pamięci Flash z CFI. Umożliwia ona programowanie pamięci za pomocą JTAG–a! Proponowane rozwiązanie pozwala na szybkie zmodyfikowanie zawartości pamięci, do czego będzie potrzebny prosty interfejs (odpowiednik ByteBlastera firmy Altera) oraz komputer PC z odpowiednim oprogramowaniem (**rys. 2**).

Opis układu

Schemat elektryczny proponowanego rozwiązania pokazano na **rys. 3**. Cała „tajemnica" projektu tkwi w niezwykłych możliwościach układu U3, który jest najmniejszym członkiem



Rys. 3. Schemat elektryczny urządzenia



Rys. 4. Schemat montażowy płytki: a) strona elementów, b) strona lutowania

rodziny układów programowalnych MAX II firmy Altera. O tym, co znajduje się w jego wnętrzu – za chwilę.

Jak widać, oprócz CPLD, na płytce znajduje się pamięć Flash (zastosowano pamięć o pojemności 32 Mb – ale można stosować większe do 512 Mb, przy czym muszą być zgodne z CFI – M29W320DB) oraz generator taktujący, wykonany na układzie U1. Na jego wyjściu występuje przebieg prostokątny (formowany w wewnętrznym przerzutniku Schmitta) o częstotliwości 6 MHz, który służy do taktowania automatu wbudowanego w U3. Układ CPLD (U3) obsługuje wszystkie sygnały magistralowe i sterujące pamięci U2. Wbudowany w niego automat „przekazuje" polecenia i dane przesyłane przez JTAG do pamięci Flash w taki sposób, że pamięć można programować i odczytywać jej zawartość szeregowo.

Jak można zauważyć z tego krótkiego opisu (nie bardzo przecież jest co od strony sprzętowej opisywać), „tajemnica" sukcesu tkwi we wnętrzu U3.

To tylko fragment możliwości

Megafunkcja *ALTPARALLEL_FLASH_LOADER* ma większe możliwości niż przedstawiono i wykorzystano w projekcie. Czytelników zainteresowanych ich poznaniem zachęcam do przejrzenia dokumentacji umieszczonej na CD–EP3/2006B.

WYKAZ ELEMENTÓW

Rezystory
R1: 2,2 MΩ 0805
R2: 1 kΩ 0805

Kondensatory
C1, C4...C9: 100 nF
C2, C3: 27 pF
C10, C11: 10 µF/10 V

Półprzewodniki
U1: 74LVC1404
U2: M29W320DB
U3: EPM240T100C3

Inne
Y1: 6 MHz
JP1, JP2: gold–pin 1x24
JP3: ZWS10

Montaż układu

Schemat montażowy płytki drukowanej pokazano na **rys. 4**. Zastosowano elementy SMD, jedynym wyjątkiem są złącza szpilkowe *gold–pin* (JP1, JP2 i JP3) oraz rezonator kwarcowy. Elementy ulokowano po obydwu stronach płytki drukowanej, dzięki czemu udało się uzyskać jej niewielkie wymiary. Wszystkie zastosowane podzespoły da się przylutować ręcznie (elementy pasywne mają rozmiar 0805, raster układów scalonych wynosi 0,5 mm) i to za pomocą standardowych lutownic (nie transformatorowych) – nie są potrzebne ani specjalne groty, ani specjalne środki pomocnicze, poza miedzianą taśmą rozlutowniczą (najlepiej posypaną pyłem z kalafonii – ułatwia on odciąganie cyny z punktów lutowniczych).

Tajniki układu MAX II

Wybór układu PLD zastosowanego w projekcie był podwójnie nieprzypadkowy:

– po pierwsze, układy z rodziny MAX II jako jedyne na świecie umożliwiają wykorzystanie JTAG–a (służącego przede wszystkim do programowania ich w systemie) do własnych zadań użytkownika,

– po drugie, producent udostępnił w ramach bezpłatnego pakietu projektowego Quartus II (projekt zrealizowano za pomocą wersji 5.1SP1) gotowy blok konwersji JTAG<–>CFI (megafunkcja *ALTPARALLEL_FLASH_LOADER* – **rys. 5**).

Otrzymujemy więc wszystko, co jest niezbędne do szybkiego wykonania projektu. Czytelników zainteresowanych możliwościami układów z rodziny MAX II odsyłam do ramki ze skróconym opisem ich architektury, a także do EP4/2004 i EP2/2005, w których to numerach opisaliśmy układy MAX II nieco dokładniej.

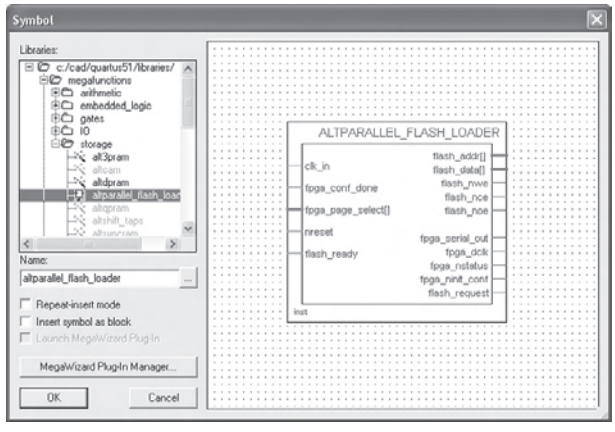
Wszystko na CD–EP3/2006B

Komplet materiałów związanych z tym projektem, w tym skompilowany projekt dla układu MAX II publikujemy na CD–EP3/2006B.

Wyprowadzenia

Moduł przedstawiony w artykule spełnia rolę pamięci Flash z interfejsem JTAG (sygnały JTAG nie zostały wyprowadzone na złącza szpilkowe podstawy). Oprócz standardowych sygnałów pamięci Flash (magistrale: adresowa, danych i sterująca), na złącza JP1 i JP2 wyprowadzono dwa dodatkowe sygnały, które umożliwiają programowanie pamięci via JTAG po jej zainstalowaniu w systemie. Są to:

– *FlashAccess* – sygnał wejściowy. Dla *FlashAccess*=1 pamięć może być programowana/odczytywana via JTAG, dla *FlashAccess*=0



Rys. 5. „Screen" projektu jest megafunkcja dostępna w systemie Quartus II

dostęp do pamięci via JTAG nie jest możliwy.
– *ACC_RQ* – sygnał wyjściowy, który sygnalizuje (gdy *ACC_RQ*=1) aktywność JTAG–a (mówiąc inaczej, jest to zgłoszenie przerwania wymuszającego dostęp do programowania pamięci Flash).

Sygnały te umożliwiają synchronizację programowania pamięci Flash z resztą systemu.

Programowanie pamięci

Programowanie pamięci Flash w tak wykonanym module wymaga zastosowania któregoś z interfejsów ByteBlaster (lub odpowiednika) oraz oprogramowania Quartus II, które jest dostępne bezpłatnie (*www.altera.com > Download*). Sposób przygotowania plików HEX/BIN do programowania pamięci został szczegółowo opisany w dokumentacji megafunkcji *ALTPARALLEL_FLASH_LOADER*.

Podsumowanie

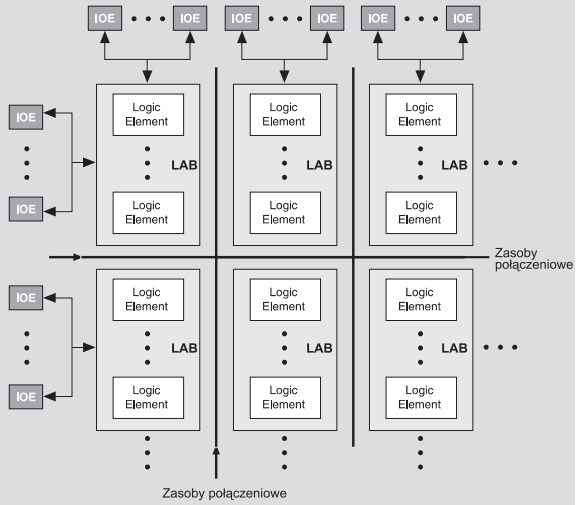
Prezentowany w artykule moduł pamięci sprawdzono w praktyce we współpracy z mikrokontrolerem LPC2292 firmy Philips. Co interesujące, interfejs pamięci Flash i JTAG mikrokontrolera były połączone w łańcuch (szeregowo), co umożliwiało programowanie obydwu układów, ale sprzętowy debugger w LPC działać nie chciał (taka jest, niestety, natura większości dostępnych na rynku interfejsów JTAG).

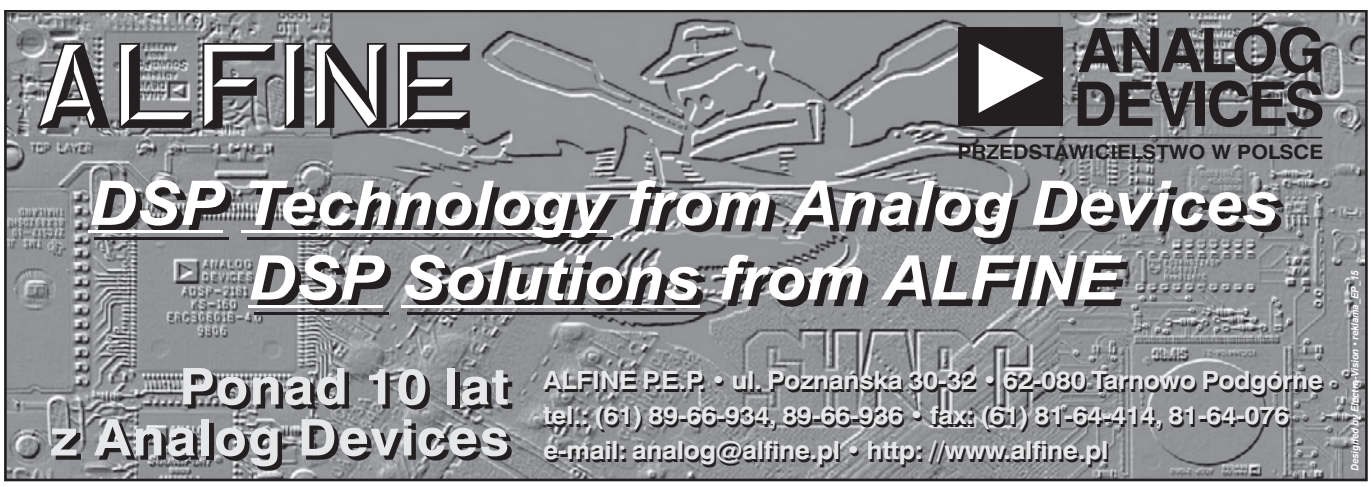
Prezentując ten projekt, wypełniliśmy istniejącą, choć może niezbyt wielką, lukę na rynku. Od teraz ISP zagarnia kolejny obszar aplikacyjny, którego obsługa w tradycyjny sposób bywała dokuczliwa.

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

Układy MAX II od środka

Podstawowym elementem tworzącym architekturę rodziny MAX II są bloki logiczne LE (*Logic Element*), których może być 240, 570, 1270 lub 2210 w zależności od typu układu. W skład każdego LE wchodzi przerzutnik o programowanej funkcji, zespół multiplekserów umożliwiających skonfigurowanie ścieżek sygnałowych, a także 4–wejściowa tablicę LUT (*Look–up Table*) – typowy element „konstrukcyjny" układów FPGA. Zadaniem tego fragmentu LE jest realizacja dowolnych, 4–argumentowych funkcji logicznych sygnałów pobieranych z zewnętrznych zasobów połączeniowych. Komunikację z otoczeniem logiki wbudowanej w układy MAX II umożliwiają specjalne komórki I/O, których zasoby logiczne producent ograniczył do minimum.



Istotną nowością zastosowaną w układach MAX II jest wbudowana w strukturę nieulotna pamięć Flash (UFM – *User Flash Memory*). Jej pojemność i organizacja są takie same we wszystkich układach tworzących rodzinę MAX II (8 kb/2 sektory). Dostęp do tej pamięci jest możliwy z zewnątrz, mają go także moduły wewnętrzne, zaimplementowane w programowalnej części układu. Producent przewidział możliwość zastosowania interfejsu szeregowego SPI lub równoległego (jak w klasycznych pamięciach Flash z wyprowadzonym sygnałem gotowości pamięci), można zastosować także natywny interfejs szeregowy, w jaki wyposażono tę pamięć. Korzystanie z pamięci UFM jest możliwe dzięki wyposażeniu systemu projektowego Quartus II (układy MAX II są obsługiwane za jego pomocą od wersji 4.0) w bibliotekę parametryzowanych makrofunkcji, wśród których są dostępne moduły pamięci Flash z różnymi interfejsami.

Służący do programowania i konfiguracji układów MAX II interfejs JTAG ma dwie cechy wyróżniające go wśród konkurencji:
– jest zgodny z IEEE1532, co zapewnia m.in. krótki czas programowania pamięci konfigurującej i wysoki poziom zabezpieczenie projektu przed nieuprawnionym kopiowaniem.
– TAP obsługuje dwie dodatkowe instrukcje (*user0* i *user1*), których znaczenie użytkownik może zdefiniować samodzielnie, dzięki czemu interfejs JTAG może być wykorzystywany także po zaprogramowaniu pamięci konfigurującej.