

# Mikrokontroler w FPGA

## Lattice dla elektroników: LatticeMico 8



*Pierwszym producentem układów PLD, który bezpłatnie udostępnił rdzeń (opisany w VHDL i Verilogu) mikroprocesora, była firma Xilinx. Jej opracowanie – PicoBlaze – jest znane Czytelnikom EP z cyklu artykułów, których publikację zaczęliśmy w EP5/2005. Teraz przedstawiamy podobne opracowanie firmy Lattice: LatticeMico8.*

Powszechna dostępność coraz tańszych układów FPGA umożliwia użytkownikom integrację funkcjonalną na niespotykaną wcześniej skalę. Nawet najtańsze wersje współczesnych układów FPGA, niezależnie od producenta, oferują zasoby umożliwiające implementację kompletnych systemów SoC (*System-on-a-Chip*), w których coraz częściej istotną rolę odgrywają kompletne mikroprocesory (nawet takie jak PowerPC). Integracja w jednej strukturze elastycznie konfigurowalnej logiki oraz równie

stycznego (od strony programowej) mikrokontrolera tworzą niezwykle uniwersalną platformę sprzętowo-programową. Dotychczas najczęściej oferowano układy FPGA, w których rdzeń mikrokontrolera był wykonywany „na sztywno” w krzemie (rys. 1a). Od niedawna prym wiodą rozwiązania znacznie bardziej elastyczne: rdzeń mikrokontrolera i jego peryferia są implementowane w strukturze FPGA w taki sam sposób jak pozostałe elementy logiczne – ich opis HDL jest poddawany klasycznej syntezie logicznej i „wpasowywaniu” zoptymalizowanego opisu w dostępne w układzie docelowym zasoby logiczne (rys. 1b). To drugie rozwiązanie ma sporą zaletę: w zależności od wymagań projektu w układ docelowy można „wbudować” albo sam rdzeń, albo rdzeń z wybranymi peryferiami. Użytkownik ma więc możliwość swobodnego kształtowania wykorzystania zasobów platformy sprzętowej, co nie jest możliwe w wersji z krzemowymi procesorami.

### 8-bitowy Lattice

Najczęstszym problemem na jaki napotykają projektanci systemów cyfrowych realizowanych w PLD jest konieczność samodzielnego projektowania automatów (sekwenserów). Ich opisy nie są co prawda – dzięki wykorzystaniu nowoczesnych narzędzi projektowych – trudne do wykonania, ale różnorodność zadań wymagających stosowania automatów sterujących powoduje, że ich projektowanie bywa jednym z najbardziej żmudnych etapów projektu.

Niebagatelne znaczenie praktyczne ma także fakt, że większość projektantów elektroniki woli część zadań realizować na drodze programowej, co również wiąże się z wygodą: łatwiej jest napisać program obsługujący sterownik wyświetlacza LCD niż opisać jego działanie w którymś z języków HDL (ale jest to, oczywiście, możliwe!).

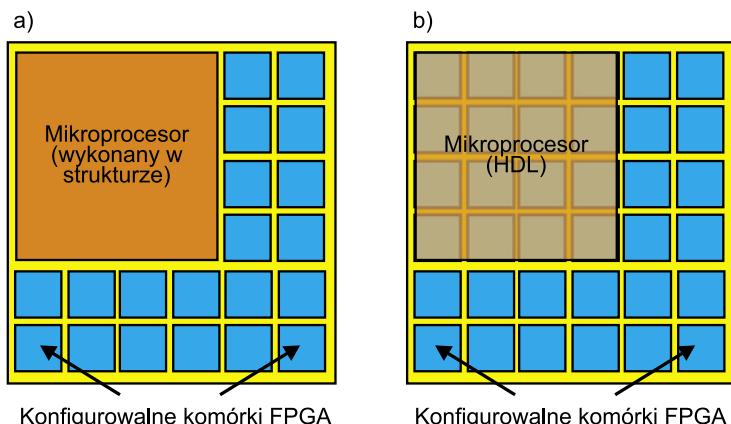
Jakie wyjście mają zatem użytkownicy układów programowalnych? Dość proste: do realizacji niektórych zadań zastosować mikrokontroler, najlepiej wbudowany w FPGA. Od niedawna (kilku tygodni) taką możliwość mają użytkownicy układów PLD firmy Lattice, a to dzięki opracowaniu o nazwie LatticeMico8.

LatticeMico8 to rdzeń 8-bitowego mikrokontrolera o dość prostej budowie (rys. 2), która jednak zapewnia realizację wszystkich zadań stawianych układom tego typu. Rdzeń składa się z następujących elementów:

- 2-taktowego rdzenia zintegrowanego z ALU,
- zespołu składającego się z 32 rejestrów 9-bitowych z możliwością bezpośredniej wymiany ich zawartości z zawartością pamięci,
- 16-poziomowego stosu,
- jednostki obsługi przerwań,
- rejestru flag (jeden zestaw, bez ich wymiany podczas obsługi przerwań),

**Mikroprocesor na „miętko”**

Coraz więcej peryferiów cyfrowych jest dostępnych w postaci opisów HDL, często nazywanymi peryferiami „miękkimi” (*soft cores*). „Miętkość” odnosi się do sposobu implementacji – narzędzia syntezy logicznej potrafią dostosować opis HDL do możliwości i budowy docelowego układu PLD (zazwyczaj FPGA). Inaczej mówiąc, mając przygotowany opis HDL jakiegoś modułu, możemy go zastosować dla praktycznie dowolnego układu docelowego (oczywiście, jeśli ma wystarczające zasoby logiczne).



Rys. 1. System-on-Chip w wersji z mikroprocesorem wbudowanym w strukturę krzemową a) i w wersji implementowanej z wykorzystaniem IPCore'a b)

# Cypress PSoC™

nowa jakość konfiguracji peryferiów



## PSoC™ - matryca analogowa i cyfrowa

- elastyczne bloki analogowe i cyfrowe
- 8 bit CPU
- tanie elementy
- low power
- tanie narzędzia uruchomieniowe

## Tanie narzędzia uruchomieniowe

- łatwe testowanie właściwości PSoC
- zawiera moduł z LCD
- zawiera in-system programator USB MiniProg

## PSoC™ Express

- intuicyjne projektowanie graficzne bez konieczności pisania kodu C

## PSoC Designer

- pełna kontrola na poziomie kodu C



Zgłoś się po broszurę PSoC

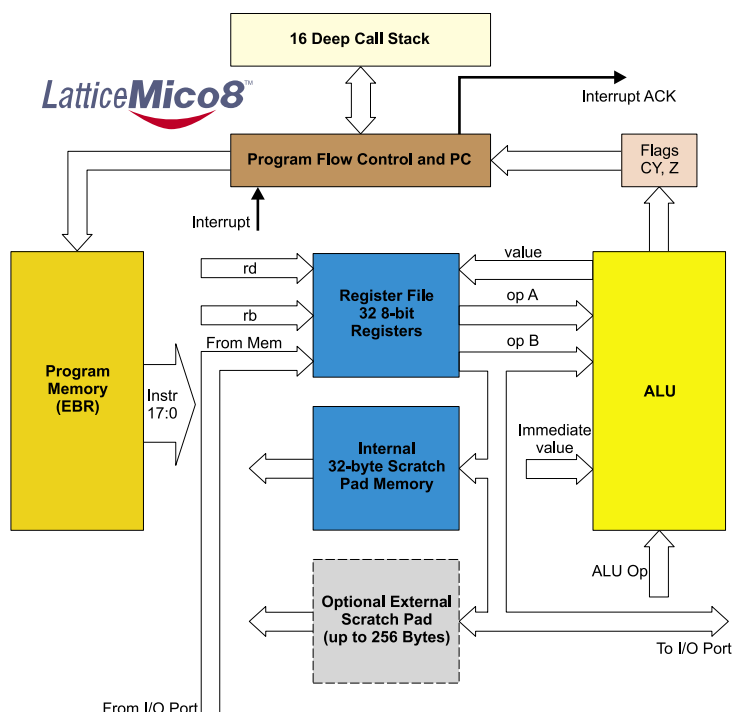


(32) 330 54 50  
Gliwice@msc-ge.com

MSC Polska Sp. z o.o.  
ul. Zygmunta Starego 11  
44-100 Gliwice  
Tel.: (32) 330 54 50  
Fax: (32) 330 54 52  
www.msc-ge.pl  
www.msc-ge.com



LatticeMico8™



Rys. 2. Schemat blokowy rdzenia Mico8

- 32-bajtowej pamięci SRAM dla podręcznych danych z możliwością powiększenia jej pojemności do 256 bajtów,
- do 256 linii I/O.

Pamięć programu LatticeMico8 jest implementowana w blokach EBR (*Embedded Block RAM* - uniwersalne zespoły SRAM) układów FPGA. Słowo rozkazowe ma szerokość 18 bitów. Standardowo przestrzeń adresowa (dla pamięci programu) jest 9-bitowa, co pozwala zaadresować do 512 słów programu. Niewielka modyfikacja w opisie HDL umożliwiła powiększenie przestrzeni adresowej,

co może spowodować zmniejszenie maksymalnej dopuszczalnej częstotliwości taktowania rdzenia.

Lista rozkazów LatticeMico8 składa się z 50 instrukcji, wśród których są dostępne zarówno instrukcje operacji logicznych i arytmetycznych (w tym porównań), skoków i wywołań procedur, operacji na znacznikach oraz portach I/O, a także instrukcje sterujące.

Prezentowany projekt został opracowany wyłącznie (jak na razie) w języku Verilog. Symbol graficzny modułu mikrokontrolera utworzony po zsyntezowaniu jego opisu

Tab. 1. Zestawienie najważniejszych parametrów wybranych implementacji rdzenia LatticeMico8 w układach FPGA firmy Lattice

Numer konfiguracji	Opis konfiguracji	Typ układu	Liczba LUT	Rejestry	Liczba SLICE	f <sub>MAX</sub> [MHz]
1	16 rejestrów uniwersalnych, 16 B SP, bez zewnętrznej pamięci SRAM	LFXP3C-4, LFEC3E-4	198	71	114	71,4 (LFXP3C-4) 77,1 (LFEC3E-4)
2	32 rejestry uniwersalne, 16 B SP, bez zewnętrznej pamięci SRAM	LFXP3C-4, LFEC3E-4	247	71	138	62,4 (LFXP3C-4) 68,8 (LFEC3E-4)
3	32 rejestry uniwersalne, 32 B SP, bez zewnętrznej pamięci SRAM	LFXP3C-4, LFEC3E-4	243	71	136	63,4 (LFXP3C-4) 70,0 (LFEC3E-4)
4	32 rejestry uniwersalne, 32 B SP, bez zewnętrznej pamięci SRAM	LFXP3C-4, LFEC3E-4	275	73	151	62,3 (LFXP3C-4) 65,6 (LFEC3E-4)

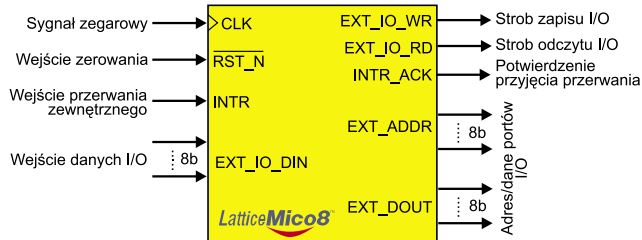
**PLD vs FPGA**

PLD (*Programmable Logic Devices*) to nazwa wszystkich programowalnych układów logicznych, wśród których jedną z podrodzin tworzą układy FPGA (*Field Programmable Logic Devices*).

(w wersji bez magistrali zewnętrznej pamięci danych) pokazano na rys. 3. W odróżnieniu od PicoBlaze'a, prezentowany mikroprocesor nie ma wyprowadzonych na „zewnątrz” magistral danych i adresowej, służących do dostępu do pamięci programu – jest ona zintegrowana z rdzeniem.

**Narzędzia**

Nawet najlepszy mikroprocesor nie jest wiele wart bez kompilatora umożliwiającego kompilację programów napisanych w jakimś języku mnemonicym. Firma Lattice zadbała o potencjalnych użytkowników mikroprocesora LatticeMico8 i przygotowała dwa proste (niestety bez IDE, ale pracują pod WinXP) programy narzędziowe: kompilator i symulator. Udostępnione zostały programy źródłowe obydwu programów, co ułatwi samodzielne udoskonalanie tych narzędzi.



Rys. 3. Symbol biblioteczny mikrokontrolera Mico8

**Implementacja**

Implementacja mikroprocesora LatticeMico8 jest możliwa przy użyciu bezpłatnego środowiska ispLever (dostępne na stronie [www.latticesemi.com](http://www.latticesemi.com)). Zsyntezowany rdzeń zajmuje 198...275 komórek LUT oraz 71...73 rejestrów (w zależności od wersji – tab. 1). Biorąc pod uwagę, że najmniejsze dostępne układy FPGA z rodziny LatticeXP wyposażono

żono w 3100 komórek LUT, w pojedynczym układzie tego typu można zintegrować kilka mikroprocesorów Mico8 (ograniczeniem będzie liczba dostępnych bloków konfigurowalnej pamięci) oraz zaawansowane peryferia. Biorąc dodatkowo pod uwagę, że wydajność każdego mikroprocesora przekracza (w najgorszym przypadku) 30 MIPS, a niektórych przypadkach dochodzi do 39 MIPS, zyskujemy wydajną (w stosunku do ceny) platformę obliczeniową, której zaletą jest możliwość praktycznie dowolnej aranżacji budowy sprzętowej. Mówiąc w skrócie: nowa jakość.

**Piotr Zbysiński, EP**  
[piotr.zbysinski@ep.com.pl](mailto:piotr.zbysinski@ep.com.pl)

**Przenosiny? Nie tak łatwo!**

Zastosowanie opisu HDL przygotowanego przez firmę Lattice nie da się bezpośrednio przenieść do innego środowiska projektowego niż ispLever. Wynika to z faktu wykorzystania przez twórców opisu LatticeMico8 makr predefiniowanych w tym środowisku.