

Zbuduj własny System-on-Chip 8051 w VHDL, część 1

Popularność mikrokontrolerów '51 nie podlega dyskusji. Od lat wzmacniają ją także dostępne w bardzo wielu firmach tworzących IP core'y - „wirtualne” wersje tych mikrokontrolerów. Niestety, korzystanie z IP core'ów nie jest możliwe bez poniesienia dość wysokich kosztów związanych z ich zakupem, stąd rozwiązania tego typu są stosowane głównie przez firmy przygotowujące projekty wysokonakładowe. Przynajmniej tak było do dziś...

Rekomendacje:

projekt polecamy wszystkim Czytelnikom, którzy interesują się nowoczesnymi sposobami konstruowania sprzętu, a także tym, którzy chcą poznać w praktyce nowoczesną technologię System-on-Chip.

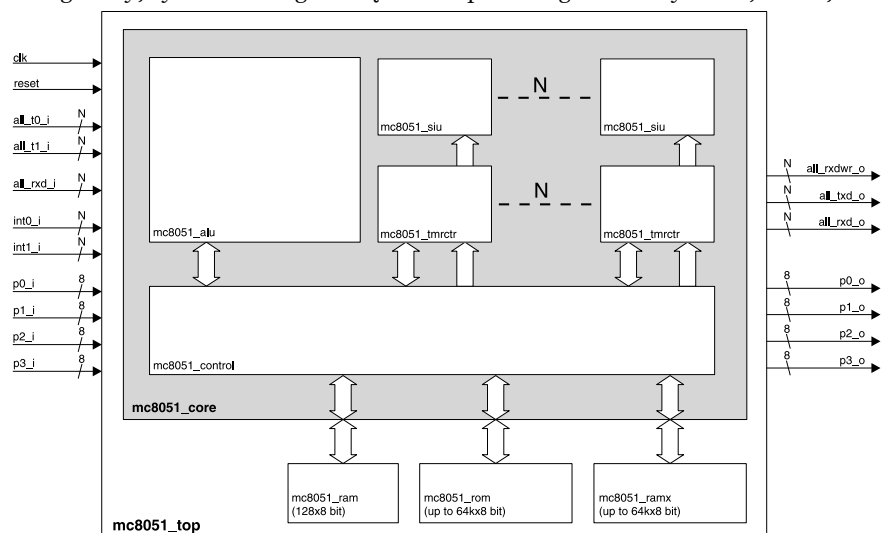
Czytelnicy wystraszeni wizją płacenia kilku tysięcy dolarów za opisany w języku HDL rdzeń mikrokontrolera '51 mogą zapytać: jaki ma to sens, jeżeli doskonale zweryfikowany, kompletny mikrokontroler z wbudowaną dużą pamięcią Flash można kupić za kilka złotych w niemal każdym sklepie z podzespołami elektronicznymi? Po co pokonywać trudności związane z implementacją takiego procesora w układ FPGA (który jest do tego droższy od gotowego mikrokontrolera!), tracić zalety programowanej w systemie pamięci Flash, ułatwiać kopiowanie swojego projektu... Wad takiego rozwiązania jest więcej niż wymieniono, ale ma ono także niewątpliwe - zwłaszcza w projektach przemysłowych - istotne zalety: projektowane urządzenie można w wielu przypadkach „zmieścić” w jednym układzie scalonym, możliwa jest łatwa (także zdalne!) „przebudowa” sprzętu pozwalająca dostosować jego funkcjonowanie do zmieniających się wymagań aplikacji, można także zbudować uniwersalną platformę, której funkcjonalność określa użytkownik poprzez zmianę plików konfiguracyjnych. Niebagatelną za-

letą implementowania kompletnych rozwiązań sprzętowych w układach FPGA jest także łatwość i niski koszt ich uruchamiania - pamięci konfiguracji w układach FPGA są typu SRAM, można więc modyfikować ich zawartość nieskończoną liczbę razy.

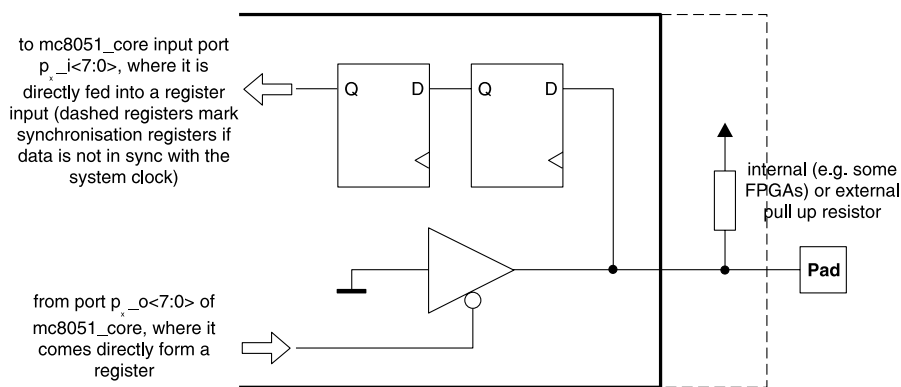
Dostępność „dużych” IP core'ów dla niewielkich firm jest mocno ograniczona ze względu na ich (względnie) wysoką cenę. Wynika ona między innymi z następujących czynników:

- relatywnie niewielkiej konkurencji pomiędzy producentami rdzeni o wysokiej jakości - takich firm jest na rynku nadal niewiele (w Polsce działają trzy),
- praca i czasochłonności opracowania i - przede wszystkim - przetestowania poprawności działania rdzenia, co jest zadaniem pod wieloma względami niebagatelnym.

Firmy tworzące IP core'y zazwyczaj bardzo niechętnie udostępniają swoje opracowania bezpłatnie, więc zazwyczaj jedyną szansą ich poznania jest korzystanie z ich zabezpieczonych (przeznaczonych wyłącznie do celów ewaluacyjnych, przez ograniczony czas) wersji lub



Rys. 1. Schemat blokowy mikrokontrolera Oregono MC8051



Rys. 2. Schemat blokowy portu I/O w mikrokontrolerze Oregono MC051

korzystanie z wersji bezpłatnych, dostępnych między innymi w portalu <http://www.opencores.org>.

Alternatywnym - do korzystania z rozwiązań komercyjnych lub udostępnionych przez wolontariuszy - wyjściem jest samodzielne opisanie w którymś z języków HDL własnych bloków funkcjonalnych. O ile stworzenie prostych interfejsów lub bloków funkcjonalnych (jak choćby te, które dotychczas opisywaliśmy w EP: generator PWM, sterowniki wyświetlacz LED i LCD, obsługa nastawnika-impulsatora) jest proste, to samodzielne opisanie rdzenia mikrokontrolera wraz z peryferiami jest zadaniem dla jednej osoby karkołomnym. Nie oznacza to, że nie da się tego zrobić! Jak pokazała praktyka, największym problemem nie jest napisanie rdzenia, lecz weryfikacja jego działania, o czym zresztą świadczą trudności na jakie napotykają użytkownicy popularnych, bezpłatnych wersji różnych rdzeni. Szukając sensownej propozycji dla Czytelników EP dotarłem do precedensowej oferty austriackiej firmy Oregono Systems, która bezpłatnie udostępniła i ciągle rozwija (czytaj: poprawia) szybki rdzeń mikrokontrolera 8051, który nie tylko można parametryzować dostosowując jego niektóre elementy do wymagań aplikacji, lecz także zwiększyć jego możliwości wykorzystując opcjonalne rozkazy wychodzące poza listę standardową.

Możliwości mikrokontrolera Oregono MC8051

Schemat blokowy mikrokontrolera MC8051 pokazano na rys. 1. Jest to konstrukcja w pełni synchroniczna, której wydajność dzięki zmodyfikowanej w stosunku do

oryginału architekturze jest znacznie większa (zgodnie z danymi producenta, program jest wykonywany przez mikrokontroler Oregono MC8051 przeciętnie 10-krotnie szybciej niż robi to mikrokontroler ze standardowym rdzeniem). W tab. 1 znajduje się zestawienie liczby cykli zegarowych koniecznych do wykonania każdego rozkazu z listy obsługiwanych przez mikrokontrolery: MC8051 i standardowy rdzeń '51.

W mikrokontrolerze MC8051 użytkownik może sam zdecydować ile timerów-liczników i interfejsów szeregowych UART ma mieć wbudowanych implementowany mikrokontroler (dostęp do nich umożliwiają dwa dodatkowe rejestry), może także uaktywnić lub zablokować rozpoznawanie przez jednostkę sterującą trzech rozkazów:

- MUL - sprzętowego mnożenia dwóch liczb 8-bitowych (kod rozkazu: 10100100 binarnie),

Czemu IP core'y?

Wbrew pozorom IP core'y nie mają za zadanie ułatwić życia leniwym konstruktorom, lecz przyspieszyć wdrażanie projektów do produkcji. Jak powiedział Jim Sansbury - jeden z założycieli firmy Altera - „Kupując IP core'y kupujesz czas, więc ich cena jest usprawiedliwiona”.

- DIV - sprzętowego dzielenia dwóch liczb 8-bitowych (kod rozkazu: 10000100 binarnie),
- DA - korekcji dziesiętnej zawartości akumulatora (kod rozkazu: 11010100 binarnie).

Zabieg ten pozwala zoptymalizować wykorzystanie zasobów układu FPGA, w którym mikrokontroler będzie implementowany.

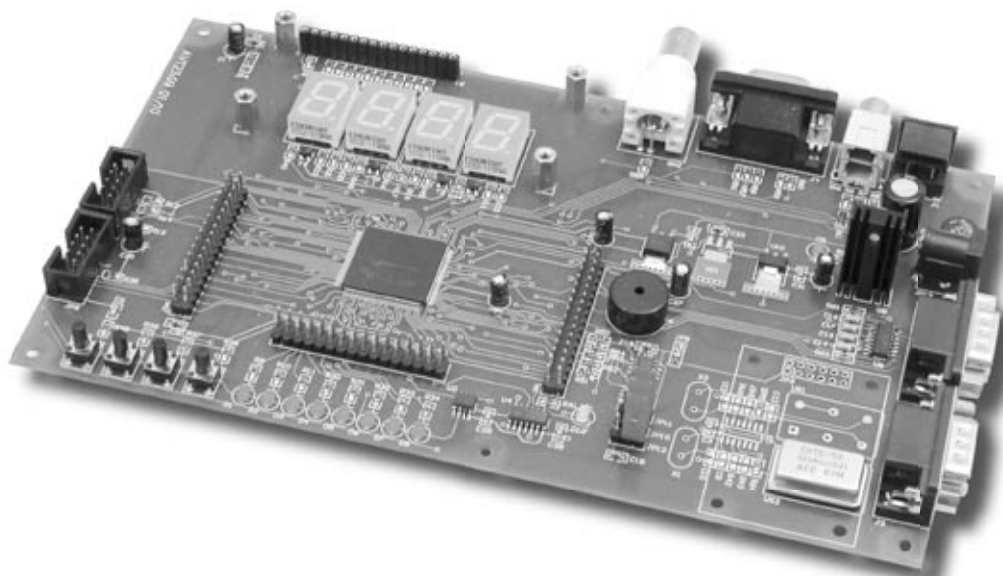
Prezentowany IP core wyposażono w cztery 8-bitowe porty I/O, przy czym w opisie rozdzielono linie wejściowe od wyjściowych (rys. 2). Takie rozwiązanie ułatwia implementację projektu w dowolnych układach FPGA, także z rodzin, w których komórki współpracujące z liniami I/O nie są przystosowane do pracy dwukierunkowej. W takich przypadkach linie wejściowe i wyjściowe są wyprowadzone na osobne wyprowadzenia układów, które należy zewrzeć ze sobą (styki fizyczne) na zewnątrz układu. W zależności od możliwości docelowego układu FPGA może okazać się konieczne zastosowanie zewnętrznych rezystorów *pull-up* dołączonych do napięcia zasilającego porty I/O. Nie jest to konieczne w przypadku współczesnych rodzin układów FPGA, jak choćby Spartan 3 (Xilinx) i Cyclone (Altera), które



Fot. 3. Widok zestawu startowego z układem Spartan 3 firmy Xilinx

Tab. 1. Porównanie liczby cykli zegarowych niezbędnych do wykonania standardowych rozkazów z listy 8051

Instrukcja	Liczba bajtów rozkazu	Liczba cykli zegarowych		Wzrost szybkości wykonywania programu
		MC8051	Standardowy '51	
ACALL	2	2	24	12
ADD A, RR	1	2	12	6
ADD A, D	2	3	12	4
ADD A, ATR1	1	2	12	6
ADD A, DATA	2	2	12	6
ADDC A, RR	1	2	12	6
ADDC A, D	2	3	12	4
ADDC A, ATR1	1	2	12	6
ADDC A, DATA	2	2	12	6
AJMP	2	2	24	12
ANL A, RR	1	2	12	6
ANL A, D	2	3	12	4
ANL A, ATR1	1	2	12	6
ANL A, DATA	2	2	12	6
ANL D, A	2	3	12	4
ANL D, DATA	3	3	24	8
ANL C, BIT	2	3	24	8
ANL C, NBIT	2	3	24	8
CJNE A, D	3	3	24	8
CJNE A, DATA	3	3	24	8
CJNE RR, DATA	3	3	24	8
CJNE ATR1, DATA	3	3	24	8
CLR A	1	1	12	12
CLR C	1	1	12	12
CLR BIT	2	2	12	6
CPL A	1	1	12	12
CPL C	1	1	12	12
CPL BIT	2	3	12	4
DA A	1	1	12	12
DEC A	1	1	12	12
DEC RR	1	2	12	6
DEC D	2	3	12	4
DEC ATR1	1	2	12	6
DIV AB	1	3	48	16
DJNZ RR	2	2	24	12
DJNZ D	3	3	24	8
INC A	1	1	12	12
INC RR	1	2	12	6
INC D	2	3	12	4
INC ATR1	1	2	12	6
INC DPTR	1	4	24	6
JB	3	3	24	8
JBC	3	3	24	8
JC	2	2	24	12
JMP A, DPTR	1	1	24	24
JNB	3	3	24	8
JNC	2	2	24	12
JNZ	2	2	24	12
JZ	2	2	24	12
LCALL	2	2	24	8
LJMP	3	3	24	8
MOV A, RR	1	2	12	6
MOV A, D	2	3	12	4
MOV A, ATR1	1	2	12	6
MOV A, DATA	2	2	12	6
MOV RR, A	1	1	12	12
MOV RR, D	2	3	24	8
MOV RR, DATA	2	2	12	6
MOV D, A	2	2	12	6
MOV D, RR	2	2	24	12
MOV D, D	3	3	24	8
MOV D, ATR1	2	2	24	12
MOV D, DATA	3	3	24	8
MOV ATR1, A	1	1	12	12
MOV ATR1, D	2	3	24	8
MOV ATR1, DATA	2	2	12	6
MOVC A, ATDPTR	1	2	24	12
MOVC A, ATPC	1	2	24	12
MOVC A, ATR1	1	2	24	12
MOVC A, ATDPTR	1	2	24	12
MOVC ATR1, A	1	1	12	12
MOVC ATPTR, A	1	1	24	24
MOV C, BIT	2	3	12	4
MOV BIT, C	2	2	24	12
MOV DPTR, DATA	3	3	24	8
MUL AB	1	3	48	16
NOP	1	1	12	12
ORL A, RR	1	2	12	6
ORL A, D	2	3	12	4
ORL A, ATR1	1	2	12	6
ORL A, DATA	2	2	12	6
ORL D, A	2	3	12	4
ORL D, DATA	3	3	24	8
ORL C, BIT	2	3	24	8
ORL C, NBIT	2	3	24	8
POP	2	2	24	12
PUSH	2	3	24	8
RET	1	3	24	8
RETI	1	3	24	8
RL A	1	1	12	12
RLC A	1	1	12	12
RR A	1	1	12	12
RRC A	1	1	12	12
SETB C	1	1	12	12
SETB BIT	2	2	12	6
SJMP	2	2	24	12
SUBB A, RR	1	2	12	6
SUBB A, D	2	3	12	4
SUBB A, ATR1	1	2	12	6
SUBB A, DATA	2	2	12	6
SWAP A	1	1	12	12
XCH A, RR	1	3	12	4
XCH A, D	2	4	12	3
XCH A, ATR1	1	2	12	6
XCHD A, ATR1	1	2	12	6
XRL A, RR	1	2	12	6
XRL A, D	2	3	12	4
XRL A, ATR1	1	2	12	6
XRL A, DATA	2	2	12	6
XRL D, A	2	3	12	4
XRL D, DATA	3	3	24	8



Fot. 4. Widok zestawu startowego z układem Cyclone firmy Altera

wyposażono w programowo włączane rezystory oraz przerytuki *bus-hold*, które zapewniają utrzymanie poprawnych stanów logicznych na wszystkich wejściach.

Mikrokontroler MC8051 jest przystosowany do współpracy z zewnętrzną pamięcią programu lub pamięcią ROM zaimplementowaną w układzie FPGA. Maksymalna przestrzeń adresowa pamięci programu wynosi 64 kB, a całkowita pojemność pamięci SRAM (łącznie z SFR) - 256 bajtów. W przypadku takiej konieczności, można także wyposażyć mikrokontroler z blok dodatkowej pamięci SRAM do przechowywania danych, której maksymalna pojemność wynosi 64 kB.

W przypadku implementacji tej pamięci wewnątrz układu FPGA konieczne jest wykorzystanie specyficznych (dla danej architektury) mechanizmów i narzędzi programowych, które zapewniają opty-

malnie „wpasowanie” bloków pamięci w zasoby układu.

Dystrybucja

Jak wspomniano na początku, projekt Oregano MC8051 jest dystrybuowany bezpłatnie na bazie licencji GNU/LGPL (jest tekst zamieszczone za miesiąc na płycie CD-EP). Wraz z plikami źródłowymi (w VHDL) jest dystrybuowana także podstawowa dokumentacja projektu, przykładowe programy testowe oraz pliki testowe dla symulatora ModelSim. Obecnie dostępna wersja 1.4 z listopada 2004 zawiera sporo udoskonaleń w stosunku do wersji poprzednich, między innymi poprawnie działają przerwania od timerów, które w poprzedniej wersji 1.2 sprawiały sporo kłopotów.

Niestety, wraz z udoskonalaniem opisów HDL, nie była przez twórców rdzenia rozwijana dokumentacja projektu, której skąpość utrud-

nia samodzielne skompilowanie projektu. Z tego powodu na CD-EP3/2005B umieścimy kompletny projekt przygotowany dla Quartusa II i układów Cyclone firmy Altera.

Platformy uruchomieniowe

Prezentowany rdzeń testowano na dwóch platformach uruchomieniowych:

- Spartan 3 Starter Kit firmy Xilinx (fot. 3) - zestaw zawiera m.in. układ XC3S200, projekt przygotowano za pomocą bezpłatnego pakietu WebPack ISE 6.3i,

- ZL2PLD (fot. 4) z układem EP1C3 z rodziny Cyclone firmy Altera, projekt realizowano z wykorzystaniem bezpłatnej wersji pakietu Quartus II 4.1 SP2.

Jakkolwiek opisy HDL przygotowane przez firmę Oregano są uniwersalne, wyniki uzyskane w przypadku implementacji na układy Spartan 3 nie są w chwili powstawania tego artykułu najlepsze (oczywiście z winy autora, a nie układów czy narzędzi). Dlatego w kolejnych częściach artykułu zostanie przedstawiona implementacja na układach Cyclone.

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

Coś dla bascomowców

Pomimo implementowania mikrokontrolera Oregano MC8051 w zaawansowanych układach FPGA, programy dla niego można pisać także w Bascomie. Za pomocą tego właśnie kompilatora napisano programy testowe, zastosowane do weryfikacji poprawności działania mikrokontrolera.

MONTAŻ SMT

- na paście
- na kleju

PROGRAMOWANIE KONSTRUOWANIE

- sterowników na bazie mikrokontrolerów 8-bitowych, 16-bitowych, 32-bitowych

PROJEKTOWANIE

- układów elektronicznych
- obwodów drukowanych

PONADTO OFERUJEMY:

- montaż mieszany: przewlekany, SMT
- lutowanie na fall lutowniczej SOLTEC MIDI z podwójną falą typu SMART WAVE

MCD Electronics
 34-300 Żywiec ul. Lelewela 26
 tel/fax: 33/861 60 35
 e-mail: smt@mcd.com.pl
 http://www.mcd.com.pl