

# PicoBlaze: przygotowywanie programów w środowisku pBlazIDE, część 2



Miesiąc temu pokazaliśmy w jaki sposób wygenerować plik wynikowy (zawierający opis zawartości pamięci programu ROM) za pomocą kompilatora pBlazIDE. Teraz zajmiemy się omówieniem sposobu zaimplementowania tej pamięci wraz z rdzeniem procesora w układzie FPGA z rodziny Spartan 3 firmy Xilinx.



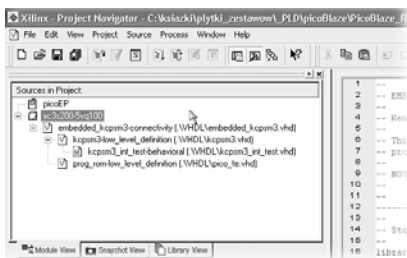
## Co kto lubi

Po uzyskaniu pliku HDL z opisem interfejsu pamięci ROM i jej zawartością, mamy do dyspozycji dwa sposoby połączenia rdzenia mikrokontrolera z pamięcią:

- z wykorzystaniem pliku VHDL zawierającego opis połączeń pomiędzy modułami,
- za pomocą edytora schematów.

W artykule skupiamy się na VHDL-owej wersji PicoBlaze'a. Ken Chapman przygotował także jego wersję w języku Verilog, którą opublikowaliśmy m.in. na CD-EP7/2005B.

W pierwszej wersji można wykorzystać plik *embedded\_kcpsm3.vhd*, który zawiera opis połączeń pomiędzy wyprowadzeniami pamięci i rdzenia PicoBlaze'a. Plik ten jest dostępny wraz z kodem źródłowym rdzenia, a jego zawartość pokazano na list. 1. Na rys. 7 pokazano strukturę plików HDL (łącznie z opisem testów funkcjonalnych, które



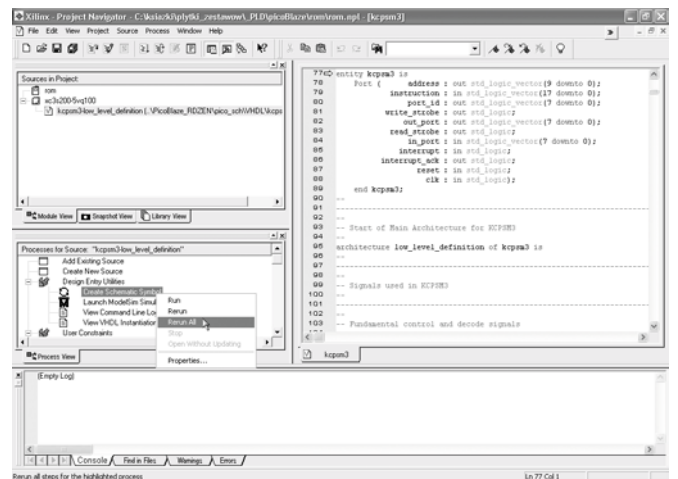
Rys. 7.

są zawarte w *kcp-sm3\_int\_test.vhd*) przy wykorzystaniu pliku z opisem połączeń.

Jeżeli projektant woli operować schematem, należy utworzyć dla plików \*.vhd zawierających opis rdzenia i pamięci programu symbole biblioteczne, które można wykorzystać w edytorze schematów. Przykładowy symbol graficzny dla pamięci ROM pokazano na rys. 6 w poprzedniej części artykułu. Teraz pokażemy jak utworzyć symbol biblioteczny z pliku \*.vhd.

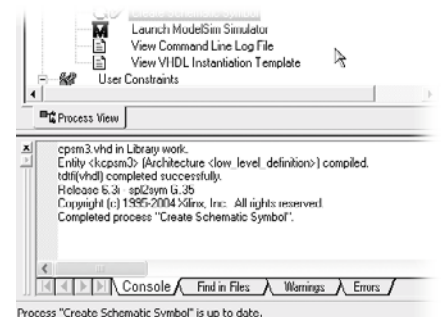
Zaczynamy od założenia nowego projektu, w skład którego będą wchodziły pliki z opisem rdzenia i pamięci ROM (rys. 8). Po zaznaczeniu kursorem (w oknie *Sources In Project*) pliku \*.vhd, dla którego będzie tworzony symbol biblioteczny, należy wybrać w oknie *Processes for Source* opcję *Create Schematic Symbol*.

Do realizacji projektu w układzie FPGA wykorzystano bezpłatny system WebPack ISE, który można ściągnąć ze strony internetowej firmy Xilinx, wielokrotnie publikowaliśmy go także na płytach CD-EP.



Rys. 8.

*matic Symbol*. Należy ją wskazać myszką i nacisnąć prawy przycisk myszki, co spowoduje otworzenie kona z dostępnymi opcjami – należy wybrać *Rerun All*. W taki sam sposób postępujemy z drugim plikiem \*.vhd. Jeżeli wszystko jest w po-



Rys. 9.

```

List. 1. Zawartość pliku embedded_kcpsm3.vhd zawierającego opis
połączeń pomiędzy rdzeniem Pico-Blaze i pamięcią programu ROM

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity embedded_kcpsm3 is
Port ( port_id : out std_logic_vector(7
downto 0);
write_strobe : out std_logic;
read_strobe : out std_logic;
out_port : out std_logic_vector(7 downto 0);
in_port : in std_logic_vector(7 downto 0);
interrupt : in std_logic;
interrupt_ack : out std_logic;
reset : in std_logic;
clk : in std_logic);
end embedded_kcpsm3;

architecture connectivity of embedded_kcpsm3 is

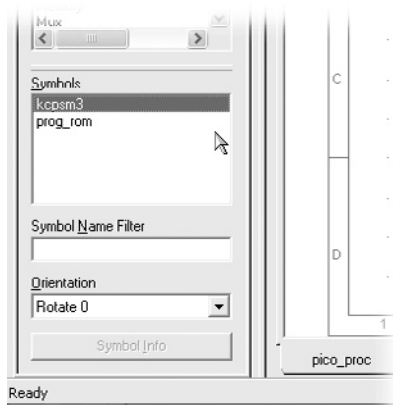
component kcpsm3
Port ( address : out std_logic_vector(9
downto 0);
instruction : in std_logic_vector(17 downto 0);
port_id : out std_logic_vector(7 downto 0);
write_strobe : out std_logic;
out_port : out std_logic_vector(7 downto 0);
read_strobe : out std_logic;
in_port : in std_logic_vector(7 downto 0);
interrupt : in std_logic;
interrupt_ack : out std_logic;
reset : in std_logic;
clk : in std_logic);
end component;

component prog_rom
Port ( address : in std_logic_vector(9
downto 0);
instruction : out std_logic_vector(17 downto 0);
clk : in std_logic);
end component;

signal address : std_logic_vector(9 downto 0);
signal instruction : std_logic_vector(17
downto 0);

begin
processor: kcpsm3
port map(address => address,
instruction => instruction,
port_id => port_id,
write_strobe => write_strobe,
out_port => out_port,
read_strobe => read_strobe,
in_port => in_port,
interrupt => interrupt,
interrupt_ack => interrupt_ack,
reset => reset,
clk => clk);

program: prog_rom
port map(address => address,
instruction => instruction,
clk => clk);
end connectivity;
    
```



Rys. 10.

rzędu, przy opcji *Create Schematic Symbol* pojawia się zielony symbol „OK/checked”, a w oknie komunikatów informacja o pomyślnym zakończeniu pracy kompilatora (rys. 9).

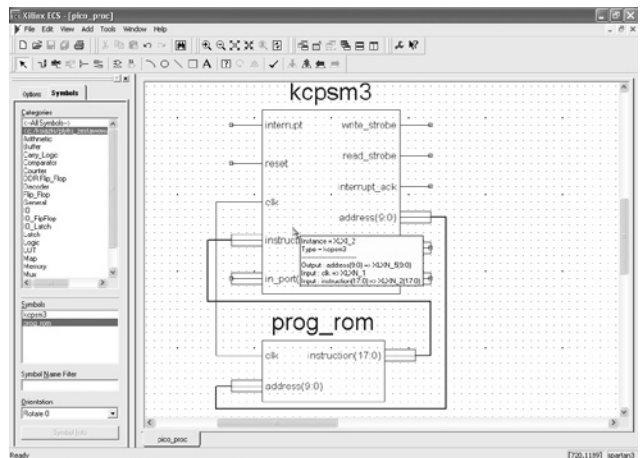
Kolejnym krokiem jest utworzenie pliku schematowego, w wyniku czego jest uruchamiany edytor schematów ECS. Na liście elementów bibliotecznych pojawią się – poza standardowymi elementami dostarczonymi wraz z systemem WebPack ISE – także dwa bloki utworzone przez nas: *kpasm3* i *prog\_rom* (rys. 10). Po umieszczeniu ich na planie schematu można wykonać połączenia

między odpowiednimi portami. Na rys. 11 pokazano przykładowy fragment schematu z wykonanymi połączeniami pomiędzy pamięcią i rdzeniem mikrokontrolera, bez podłączeń do fizycznych wyprowadzeń układu FPGA.

**Co dalej?**

Niezależnie od przyjętego sposobu realizacji projektu, efektem jego kompilacji będzie plik służący do zapisania w pamięci konfigurującej (konfiguratorze) FPGA lub służący do zapisu bezpośrednio w pamięci konfiguracji SRAM układu FPGA. Ponieważ zdecydowana większość pytań dotyczy tego właśnie etapu implementacji projektu, sposobom konfiguracji układu Spartan 3 poświęcimy kolejną część artykułu.

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



Rys. 11.

**ACSELEKTRONIK**

Szydłowiec 26-500 ul.Kolejowa 11 e-mail: [acs@acs.ats.pl](mailto:acs@acs.ats.pl)  
 Tel/fax 0486176000, tel 0600332061

**OSCYSKOPIY CYFROWE**

[www.acs.ats.pl](http://www.acs.ats.pl)

**ADS 220 2x60MHz 200MSPS**



- ✓ pasmo 2x60MHz
- ✓ rozdzielczość 8bitów/kanał
- ✓ próbkowanie 2x200MSPS, 3.3 x pasmo
- ✓ zakres 5mV-5V/DIV (1:1)
- ✓ zewnętrzny kanał wyzwalania EXT
- ✓ analiza FFT
- ✓ interpolacja przebiegów sin(x)/x
- ✓ autokalibracja 24bitowa
- ✓ wyjście kompensacji sond pomiarowych
- ✓ impedancja wejściowa 1M, pojemność 20pF
- ✓ połączenie z komputerem IEEE1284-ECP
- ✓ pełny resampling przebiegu (możliwość zmiany czasu i wzmocnienia na zatrzymanym przebiegu)
- ✓ automatyczne pomiary: częstotliwość, okres, peak to peak, RMS, wartość średnia
- ✓ symulacja wirtualnej płyty czołowej oscyloskopu
- ✓ oparty na układach AnalogDevices, Burr-Brown, Xilinx
- ✓ w połączeniu z komputerem notebook - idealne stanowisko pomiarowe

**PROGRAMATORY PAMIĘCI**

**Uniwersalne programatory Vi-LAB, ERICA, Ps32**

- ✓ Vi-LAB wirtualne laboratorium
- ✓ programator 1400 układów, ZIF 48Pin 0,3"-0,6"
- ✓ tester TTL, CMOS, PLD
- ✓ emulator czasu rzeczywistego (8MB-16Bit 27xxx, 62xxx, 24Cxx, 93Cxx, 25/95xxx)
- ✓ komunikacja port drukarkowy ECP
- ✓ samodzielne dodawanie nowych algorytmów język ISPA



**Profesjonalne programatory XELTEK**



- SuperPRO 8000, 2000, 680, V, LX, 280, Z
- ✓ obsługa ponad 8000 układów
- ✓ modele z LCD pracujące bez komputera
- ✓ programatory wielokrotnie o wydajności 1000 układów/h
- ✓ praca z układami większymi niż 100końcówek