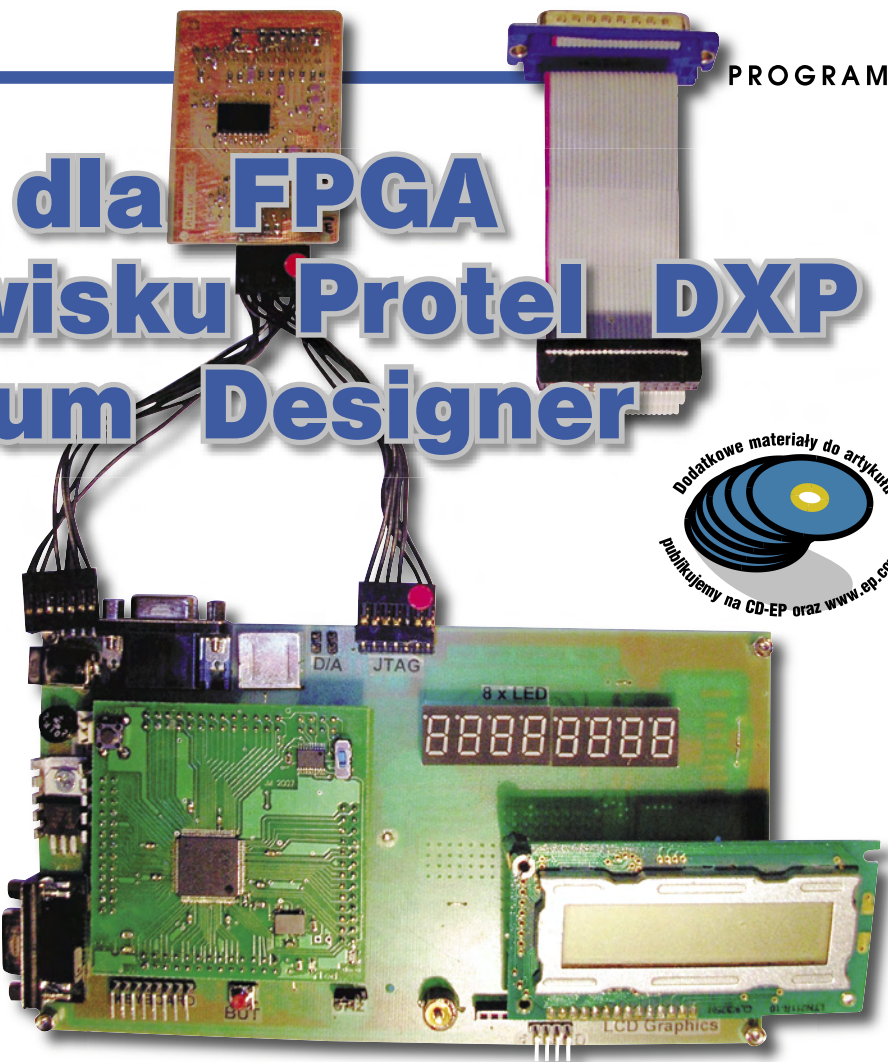


# Projekty dla FPGA w środowisku Protel DXP oraz Altium Designer



Duży nacisk w pakiecie Protel DXP i jego nowszej wersji – Altium Designer – położono na umożliwienie przygotowywania projektów na układy FPGA. Narzędzia zintegrowane w pakiecie projektowym pozwalają przygotowywać projekty dla układów FPGA dwóch największych producentów – firm: Altera i Xilinx. Przed konstruktorami otworzyła się nowa, wygodna droga nowoczesnej implementacji złożonych projektów cyfrowych.

Wysiłek włożony przez twórców pakietu DXP we wbudowanie narzędzi do tworzenia projektów z zastosowaniem układów FPGA przyniósł imponujące efekty. Projekty można przygotowywać za pomocą schematów, opisów HDL oraz nowych urządzeń jak interfejs Open-Bus. Producent przygotował wiele gotowych modułów funkcjonalnych, które użytkownicy mogą wykorzy-



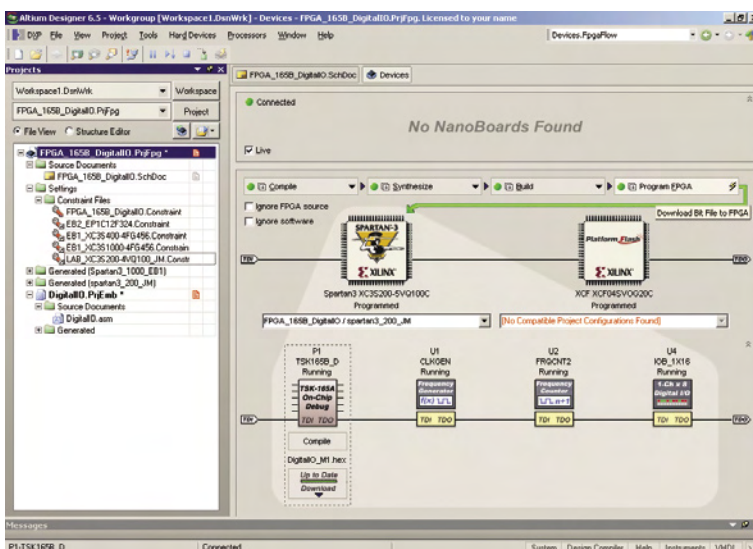
stywać we własnych projektach. Najważniejsze wśród nich są rdzenie mikrokontrolerów i mikroprocesorów typu 8051, PIC16, Z80 oraz innych, w tym także 32-bitowych. Programy dla nich można przygotowywać i kompilować w języku GNU C, ale standardowo w system jest wbudowany kompilator Tasking, debugger OCDS i oczywiście w assemblerach. Obawy Czytelników

o to, czy takie procesory nie są zbyt duże dla zasobów dostępnych w FPGA są – jak pokazały eksperymenty – bezpodstawne. Przykładowo w tanim układzie XC3S200 mieszczą się dwa rdzenie: 8051 i PIC16 – każdy z pamięcią programu o pojemności 4 kB.

Twórcy pakietu DXP/Designer założyli, że eksperymenty z FPGA będą prowadzone na bazie zestawów uruchomieniowych NanoBoard firmy Altium. Problem ten można jednak ominąć, co pozwala na prowadzenie eksperymentów na dowolnych zestawach uruchomieniowych z układami FPGA firm Altera, Xilinx, Lattice i Actel. Rozwiązanie przedstawiamy w artykule.

## Przykładowy projekt

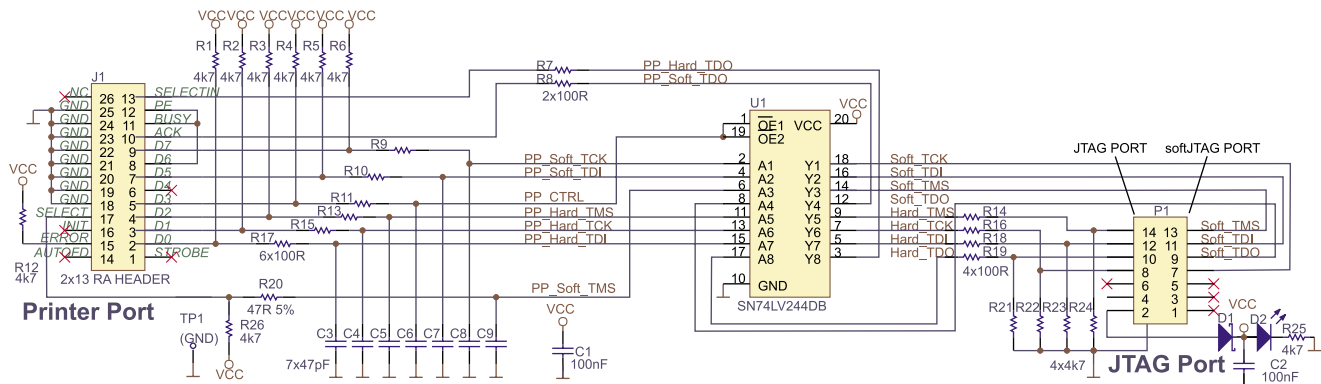
Wśród projektów demonstracyjnych dystrybuowanych z pakietem DXP znajduje się projekt o nazwie *Digital I/O* (rys. 1). Skupmy się na strukturze zbiorów drzewa projektu. Widać w niej, że cały projekt FPGA narysowano w postaci schematu (plik o rozszerzeniu *.SchDoc*). Do takiego projektu należy dodać własny zbiór zawierający przypisanie sygnałów do wy-



Rys. 1. Widok okna projektu *Digital I/O*

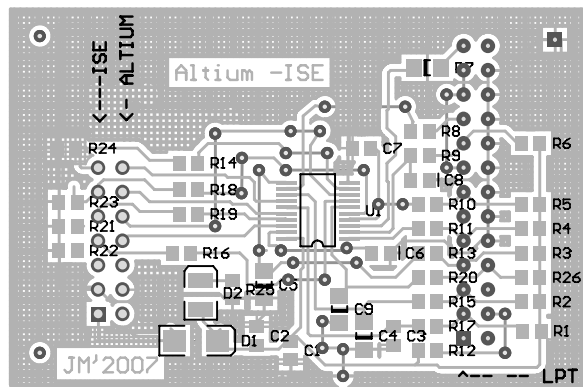
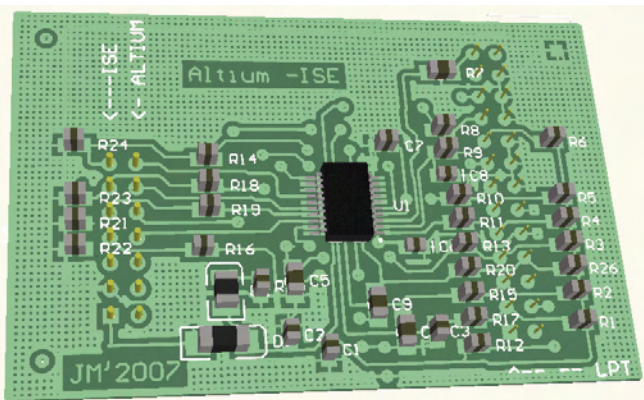
**Interfejs do konfiguracji FPGA z poziomu Protela DXP/Altium Designera**

Schemat elektryczny interfejsu przedstawiono na rysunku poniżej. Najważniejszym elementem urządzenia jest bufor 74LVC244. Interfejs jest zasilany od strony programowanego układu FPGA. Kondensatory o pojemności 47 pF są niezbędne, gdy długość taśmy łączącej z komputerem PC nie jest większa niż 50 cm.



Widok płytki obwodu drukowanego interfejsu przedstawiono na kolejnym rysunku.

Zaprojektowano ją jako jednowarstwową, w związku z czym niezbędne są połączenia wykonane przewodem, tak jak to zaznaczono na schemacie montażowym poniżej.



Po „drugiej” stronie interfejsu znajduje się złącze dwurzędowe. Jeden z rzędów to „klasyczne” złącze JTAG kompatybilne ze standardem narzuconym przez firmę Xilinx w jej programatorach. Drugi rząd ma jedynie cztery aktywne wyprowadzenia. Można je podłączyć do dowolnych, nie wykorzystanych w aplikacji wyprowadzeń I/O układu FPGA.

**Podłączenie interfejsu**

Prezentowany interfejs może być podłączony do dowolnego układu FPGA firmy Xilinx. Wystarczy przeznaczyć 4 wolne linie I/O aby mógł działać interfejs soft JTAG. Jego przeznaczenie to debugowanie procesorów oraz sterowanie wirtualnymi przyrządami w trakcie działania układu FPGA. Szczegóły przedstawimy przy okazji opisu przykładowego projektu.

prowadzeń fizycznego układu, tzw. *constraint file*. W drzewie projektu widocznym na rys. 1 jest to zbiór o nazwie *LAB\_XC3S200-4VQ100\_JM.Constraint*. Prawa strona rysun-

ku zawiera widok konfiguracji środowiska sprzętowego – w górnym łańcuchu JTAG znajdują się dwa układy, bo w testowym zestawie znajdowały się dwa fizyczne ukła-

dy: FPGA i konfigurator. Dolny łańcuch soft JTAG zawiera wirtualne układy umożliwiające debugowanie i sterowanie w trakcie działania układu FPGA. Wystarczy

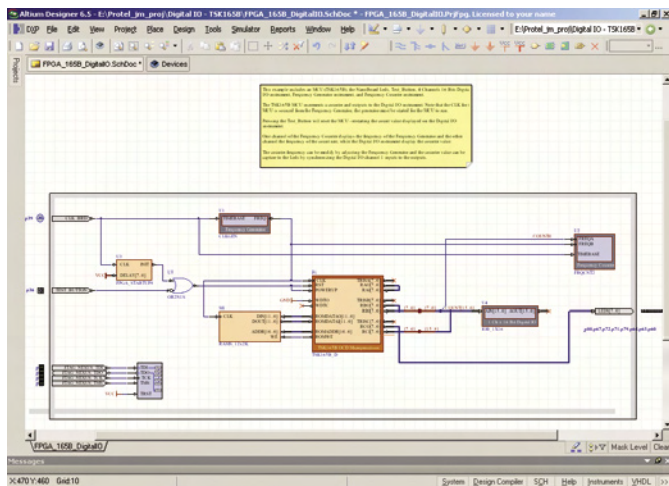
**List. 1. Zbiór zawierający informacje o przypisaniach sygnałów I/O do wyprowadzeń FPGA**

```
;Constraints File
Record=FileHeader | Id=DXP Constraints v1.0
Record=Constraint | TargetKind=PCB | TargetId = Live XC3S200 VQ100
Record=Constraint | TargetKind=Part | TargetId= XC3S200-5VQ100C

;System Clock
Record=Constraint | TargetKind=Port | TargetId=CLK_BRD | FPGA_PINNUM=p39
Record=Constraint | TargetKind=Port | TargetId=CLK_BRD | FPGA_CLOCK_PIN=TRUE
Record=Constraint | TargetKind=Port | TargetId=TEST_BUTTON | FPGA_PINNUM=p36 | pullup

; LEDs
Record=Constraint | TargetKind=Port | TargetId=LEDS[7..0] | FPGA_PINNUM= p80, p67, p72, p71, p79, p64, p63, p60

; NEXUS JTAG Soft-Device Chain Connections
Record=Constraint | TargetKind=Port | TargetId=JTAG_NEXUS_TCK | FPGA_PINNUM=p1
Record=Constraint | TargetKind=Port | TargetId=JTAG_NEXUS_TDO | FPGA_PINNUM=p2
Record=Constraint | TargetKind=Port | TargetId=JTAG_NEXUS_TDI | FPGA_PINNUM=p5
Record=Constraint | TargetKind=Port | TargetId=JTAG_NEXUS_TMS | FPGA_PINNUM=p87
```



Rys. 2. Schemat projektu Digital I/O

włączyć przycisk *Live* oraz nacisnąć przycisk *Program FPGA*, aby dokonać kompilacji i skonfigurowania FPGA projektem zapisanym w zbiorze z rozszerzeniem *.bit*.

**Schemat projektu**

Na rys. 2 przedstawiono schemat implementowanego projektu. Składa się on z rdzenia PIC165X (TSK165B\_D) pracującego w trybie z debugowaniem. Dołączona jest do niego pamięć programu typu RAM, a nie ROM. Zrobiono tak, żeby było możliwe wstawianie *breakpointów*. Na schemacie są także widoczne wirtualne instrumenty (ich panel pokazano na rys. 3). Układ CLKGEN pozwala dzielić częstotliwość oscylatora i wpływać na szybkość działania procesora. Program

dla procesora, napisany w języku C, symuluje działanie licznika binarnego na portach RB i RC. Zliczane stany mogą być uwidocznione na diodach LED prawdziwych i wirtualnych IOB\_1X16. Dodatkowo na schemacie umieszczono wirtualny układ FRQCNT2 zliczający zbocza sygnału zegarowego. Aby można korzystać z trybu debugowania procesora oraz wirtualnych instrumentów potrzebny jest na schemacie moduł NEXUS\_JTAG\_PORT (*soft JTAG*).

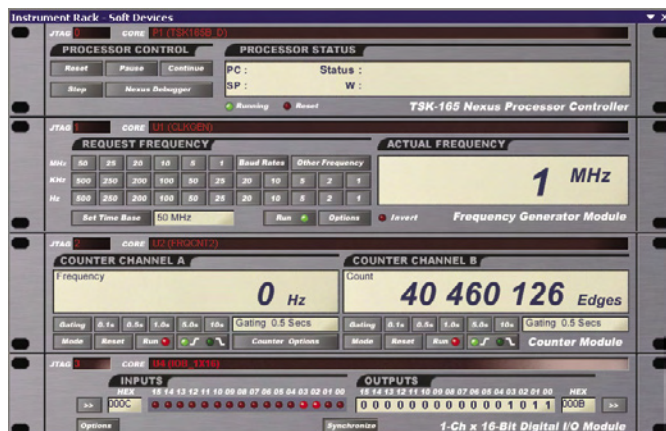
**Zbiór constraint**

Wszystkie wyprowadzenia układu FPGA muszą zostać przypisane do fizycznych pinów I/O układu FPGA. Takie przypisanie można zrobić w zbiorze tekstowym typu *.Constraint*. Na list. 1 pokazano

przykładowy zbiór dla prezentowanego schematu. Wystarczy wstawić numery pinów dla układu FPGA, którego chcemy użyć. Warto zauważyć że w zbiorze *constraint* opisane są także miejsca (piny) podłączenia interfejsu *soft JTAG*.

Struktura zbioru *constraint* w paczce DXP jest inna niż stosowana przez firmę Xilinx. Ten zbiór można zdefiniować, opisując wszystkie możliwe sygnały znajdujące się na płycie zestawu z układem FPGA. W zbiorach *constraint* stosowanych w projektach dla środowiska WebPack nie można stosować „nadmiarowych” definicji opisujących przypisanie sygnałów do wyprowadzeń FPGA.

**Jacek Majewski  
PWr**



Rys. 3. Panel wirtualnych instrumentów projektu Digital I/O

R
E
K
L
A
M
A

**Pomiar współczynnika fali stojącej oraz mocy**  
 SWR: 1:1 – 1:3  
 F: 1,7 – 30MHz  
 Moc: 10 / 100W  
 Impedancja: 50 ohm  
 Wymiary: 8,5 x 5,5 x 5,6

kod: URZ0514  
 CENA BRUTTO: 40,00 zł

**Pomiar współczynnika fali stojącej oraz natężenia pola**  
 SWR: 1:1 – 1:3  
 F: 1,7 – 30MHz  
 Impedancja: 50 ohm  
 Wymiary: 8,5 x 3,5 x 5,6

kod: URZ0513  
 CENA BRUTTO: 35,00 zł

**Pomiar współczynnika fali stojącej, mocy oraz natężenia pola**  
 SWR: 1:1 – 1:3  
 F: 1,7 – 30MHz  
 Moc 10 / 100W  
 Impedancja: 50 ohm

kod: URZ0522  
 CENA BRUTTO: 53,10 zł

AVT Korporacja, 03-197 Warszawa, ul. Leszczyńska 11  
 tel. 022 257 84 50, fax 022 257 84 55, handlowy@avt.pl, www.sklep.avt.pl