

# System w układzie programowalnym z mikrokontrolerem Propeller

Piotr Kardasz

Politechnika Białostocka, Wydział Elektryczny, ul. Wiejska 45A, 15-351 Białystok

**Streszczenie:** Artykuł przedstawia system w układzie programowalnym FPGA z mikrokontrolerem P8X32A Propeller. Kod tego mikrokontrolera został opublikowany w sierpniu 2014 r. na otwartej licencji GPL w wersji 3. System, zawierający sterowniki grafiki i dźwięku, a także klawiatury, myszy komputerowej i pamięci masowej, przeznaczony jest do zastosowań związanych z przetwarzaniem sygnałów dźwiękowych. Przedstawiona została struktura zaprojektowanego systemu oraz możliwości jego zastosowań.

**Słowa kluczowe:** FPGA, mikrokontroler, SoPC, HDL, Verilog

## 1. Wprowadzenie

### 1.1. Układy programowalne FPGA

Układy programowalne FPGA (ang. *Field Programmable Gate Array*) zdobywają w ostatnich latach coraz większe zainteresowanie. Ich podstawową zaletą jest elastyczność – stanowiąc matrycę uniwersalnych jednostek logicznych pozwalają na implementację dowolnego (w ramach pojemności i prędkości zastosowanego układu FPGA) układu cyfrowego [1–3]. Wady, które do niedawna stanowiły duże ograniczenie w wykorzystaniu tych układów – ich niewielka pojemność, wysoka cena i trudności w projektowaniu systemów z ich użyciem – wraz z rozwojem technologii tracą na znaczeniu. W chwili obecnej dostępne są na rynku zarówno układy FPGA, jak i zestawy uruchomieniowe na nich oparte, których cena zbliżona jest do cen podobnych urządzeń opartych na mikrokontrolerach. Jednocześnie kompilatory udostępniane przez producentów tych układów są coraz szybsze i udostępniają coraz większe możliwości.

Klasyczne układy sterowania i inne urządzenia tego rodzaju składają się z mikrokontrolera oraz dołączonych do niego układów peryferyjnych. W przypadku układu FPGA zarówno mikrokontroler, jak i układy peryferyjne mogą zostać zaprogramowane wewnątrz jego struktury. Jeśli z jakichś przyczyn potrzebna jest zmiana lub rozbudowa systemu, dzięki elastyczności układów programowalnych nie ma potrzeby zmiany schematu elektrycznego – nowy projekt może być wprowadzony do takiego systemu na miejscu jego pracy.

Przedstawione cechy układów FPGA powodują, że ich popularność w ostatnich latach szybko rośnie i coraz częściej można spotkać na rynku urządzenia oparte na tych układach.

### 1.2. Układy FPGA a mikrokontrolery

Projektowanie systemów w układzie programowalnym FPGA polega na stworzeniu opisu układu cyfrowego w jednym z dostępnych języków (VHDL, AHDL, Verilog [4, 5]), bądź zaprojektowaniu go przy użyciu narzędzi graficznych. Możliwe jest także łączenie w jednym projekcie różnych języków i technik projektowania. Tak stworzony projekt należy następnie skompilować. Narzędzia do tego celu są udostępniane przez producentów układów FPGA. Projekt po kompilacji może być przetestowany za pomocą symulatora, a następnie uruchomiony w układzie docelowym. Ponieważ pamięć FPGA jest ulotna, producenci oferują specjalizowane układy zawierające pamięć nieulotną, a które po załączeniu zasilania programują układ FPGA zapisaną do nich zawartością.

Dostępna pojemność układów programowalnych pozwala na implementację w ich strukturze zarówno procesora, jak i układów peryferyjnych. Możliwe jest więc zaprojektowanie własnego, dostosowanego do potrzeb użytkownika, mikrokontrolera i zbudowanie wokół niego niezbędnych układów peryferyjnych. Niestety, tak zbudowany system pozbawiony jest jakiegokolwiek oprogramowania i należy dla niego stworzyć zarówno zestaw narzędzi programistycznych (*toolchain*), jak i – w następnej kolejności – niezbędne oprogramowanie. Działania takie były podejmowane w celach demonstracyjnych (program demonstracyjny *Parallelogram* Linusa Akessona, [www.linusakesson.net](http://www.linusakesson.net)), jednakże ze względu na ich pracochłonność podejście takie nie może zostać zastosowane przy projektowaniu systemów zarówno w celach praktycznych, jak również eksperymentalnych.

Do niedawna nie było możliwe zaimplementowanie w układzie FPGA żadnego z dostępnych na rynku mikrokontrolerów. Producenci nie ujawniali bowiem ich wewnętrznej struktury, nie mówiąc już o udostępnianiu gotowych rdzeni napisanych w jakimkolwiek języku HDL. Istnieje, co prawda, witryna internetowa, [opencores.org](http://opencores.org), na której dostępne są tego rodzaju rdzenie – kłony popularnych procesorów, jednak są to wyniki pracy

#### Autor korespondujący:

Piotr Kardasz, [pik@we.pb.edu.pl](mailto:pik@we.pb.edu.pl)

#### Artykuł recenzowany

nadesłany 18.11.2014 r., przyjęty do druku 26.01.2015 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

hobbystów, w żaden sposób nie są wspierane, a często zwalczane przez producentów sprzętu, którzy nakazują zaprzestanie publikowania takiego kodu. Często są to też implementacje częściowe, pozbawione niektórych funkcji oryginalnych układów lub nie do końca z nimi kompatybilne. Korzystanie z tego rodzaju rozwiązań przy projektowaniu własnych urządzeń jest więc ryzykowne, a może być również sprzeczne z prawem.

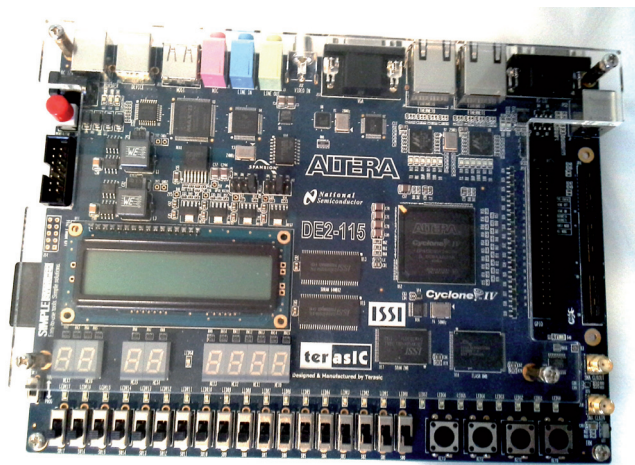
Istnieją także rozwiązania od początku otwarte, takie jak OR1000/1200, dysponujące zarówno dobrze rozwiniętym zestawem narzędzi, jak również bogatymi bibliotekami oprogramowania, jednak i to rozwiązanie nie jest wspierane przez żadną firmę, ani dostępne w formie gotowego układu scalonego.

W tej sytuacji ważnym wydarzeniem w dziedzinie projektowania systemów z układami FPGA stała się w sierpniu 2014 r. publikacja na zasadach otwartej licencji GPL 3.0 kodu mikrokontrolera Propeller – P8X32A. Mikrokontroler ten, produkowany od 2006 r., dostępny jest w postaci układu scalonego i posiada wsparcie zarówno producenta, jak również aktywnej społeczności projektantów i programistów. Dostępna jest dla niego bogata biblioteka oprogramowania, zarówno narzędziowego, jak również biblioteki procedur spełniających różnorodne funkcje.

## 2. System w układzie FPGA

### 2.1. Założenia systemu i wybór mikrokontrolera

Opisywany system powstał w ramach pracy badawczej prowadzonej na Wydziale Elektrycznym Politechniki Białostockiej. Praca ta poświęcona jest zaawansowanym algorytmom przetwarzania sygnałów akustycznych. Środowisko układów programowalnych FPGA umożliwia projektowanie układów realizujących tego rodzaju algorytmy w sposób równoległy, umożliwiając dzięki temu ich szybką pracę.



Rys. 1. Płytką uruchomieniową DE2-115

Fig. 1. DE2-115 development board

W celu realizacji zadania wybrana została płytką uruchomieniową DE2-115 z układem FPGA Cyclone IV (rys. 1) firmy Altera, dysponującym ok. 115 000 jednostek logicznych. Oprócz układu FPGA płytką ta zawiera pamięć operacyjną (2 MB SRAM i 128 MB SDRAM), video DAC z wyjściem VGA, kodek audio umożliwiający pracę z częstotliwością próbkowania 96 kHz i rozdzielczością 24 bity, gniazdo klawiatury PS2, gniazdo karty SD, oraz wiele innych układów peryferyjnych. Nadaje się w związku z tym do realizacji zadań związanych z przetwarzaniem sygnałów, które wymagają ich wprowadzenia do systemu i wyprowadzenia ich z niego po obróbce. Możliwa jest również wizualizacja wyników pracy na standardowym monitorze oraz sterowanie systemem za pomocą klawiatury i myszy. Dostępność pamięci operacyjnej i masowej umożliwia realizację procesów,

które ze względu na złożoność nie mogą być przeprowadzone w czasie rzeczywistym.

Projektowanie w układzie FPGA wszystkich bloków funkcjonalnych odpowiedzialnych za sterowanie układami peryferyjnymi byłoby czasochłonne i kłopotliwe; w celu sterowania niektórymi z nich wygodniej jest użyć mikroprocesora lub mikrokontrolera. Do chwili opublikowania kodu mikrokontrolera Propeller możliwe były rozwiązania, polegające na:

- użyciu zewnętrznego mikrokontrolera, połączonego za pomocą magistrali GPIO z płytką uruchomieniową FPGA,
- użyciu jednego z dostępnych na portalu *opencores.org* rdzeni mikroprocesorów lub mikrokontrolerów,
- zastosowaniu mikroprocesora NIOS II firmy Altera.

To ostatnie rozwiązanie, mimo że wydaje się w pierwszej chwili optymalne dla wybranego sprzętu, wiąże się jednak z wieloma problemami. Zastosowanie pełnej wersji tego procesora wymaga zakupu pełnej wersji oprogramowania Quartus II, co pociąga za sobą duże koszty. Wersja NIOS II/e dostępna bezpłatnie jest powolna – maksymalna częstotliwość stabilnej pracy osiągnięta podczas eksperymentów nie przekroczyła 80 MHz; jako że jedna instrukcja tego procesora wymaga co najmniej 6 taktów zegara systemowego, osiągnięta szybkość przetwarzania rzędu 10 MIPS okazała się nie być satysfakcjonująca. Kod mikroprocesora NIOS II nie jest udostępniany przez firmę Altera, co uniemożliwia jego modyfikację.

Eksperyment pokazał, że rozwiązanie polegające na połączeniu płytki z układem FPGA z zewnętrznym mikrokontrolerem za pomocą magistrali GPIO charakteryzuje się małą odpornością na zakłócenia zewnętrzne oraz niewystarczającą prędkością transmisji danych.

W tej sytuacji została podjęta wstępna decyzja o zastosowaniu jednego z ogólnodostępnych rdzeni RISC (OR1000/OR1200), jako pozbawionego wad zarówno procesora NIOS II, jak i zewnętrznego mikrokontrolera. Jednak w czasie podejmowania tej decyzji, został opublikowany na otwartej licencji kod mikrokontrolera Propeller. Jego zalety, w porównaniu z innymi, to wysoka wydajność, wielordzeniowość, prostota programowania, determinizm czasowy, bogate biblioteki oprogramowania, aktywna społeczność i wsparcie producenta. Mikrokontroler ten został w tej sytuacji wybrany w celu sterowania docelowym systemem.

### 2.2. Mikrokontroler P8X32A Propeller

Mikrokontroler Propeller o symbolu P8X32A jest produkowany od 2006 r. przez amerykańską firmę Parallax [6, 7]. Dostępny jest w 40-nóżkowej obudowie DIP, a także w 44-nóżkowych obudowach QFP i QFN przeznaczonych do montażu powierzchniowego. Zasilany jest napięciem 3,3 V; także porty wejścia i wyjścia przystosowane są do pracy z tym napięciem.

Propeller ma 8 32-bitowych rdzeni RISC, określanych w opisie mikrokontrolera jako *cogs*. Każdy z nich ma do dyspozycji 2 kB pamięci programu i danych. Pamięć ta, zorganizowana jako 512 32-bitowych komórek, pełni również rolę rejestrów danego rdzenia. 496 z nich stanowi rejestry uniwersalne, mogące zawierać zarówno instrukcje procesora jak i dane, 16 pozostałych to rejestry specjalne (tab. 1) sterujące różnymi funkcjami rdzenia.

Tego rodzaju architektura umożliwia pisanie samomodyfikującego się kodu – w tym celu opracowano specjalne instrukcje asemblera, umożliwiające niezależną zmianę kodu instrukcji, adresu źródła oraz adresu docelowego w danej komórce pamięci rdzenia.

Wszystkie rdzenie współdzielą między sobą główną pamięć mikrokontrolera. Pamięć ta składa się z 32 kB pamięci stałej ROM oraz 32 kB pamięci RAM. Pamięć RAM służy do przechowywania programów i danych, które mogą być z niej ładowane do pamięci poszczególnych rdzeni. Pamięć ROM zawiera podstawowe oprogramowanie umożliwiające start mikrokontrolera po załączeniu zasilania, interpreter dedykowanego języka pro-

**Tab. 1. Rejestry specjalne mikrokontrolera Propeller**  
Tab. 1. Propeller microcontroller special registers

Adres	Nazwa	Funkcja
496 (\$1F0)	PAR	Parametr startowy
497 (\$1F1)	CNT	Licznik taktów zegara
498 (\$1F2)	INA	Wejście portu A
499 (\$1F3)	INB	Wejście portu B
500 (\$1F4)	OUTA	Wyjście portu A
501 (\$1F5)	OUTB	Wyjście portu B
502 (\$1F6)	DIRA	Kierunek danych portu A
503 (\$1F7)	DIRB	Kierunek danych portu B
504 (\$1F8)	CTRA	Konfiguracja licznika A
505 (\$1F9)	CTRB	Konfiguracja licznika B
506 (\$1FA)	FRQA	Częstotliwość licznika A
507 (\$1FB)	FRQB	Częstotliwość licznika B
508 (\$1FC)	PHSA	Stan licznika A
509 (\$1FD)	PHSB	Stan licznika B
510 (\$1FE)	VCFG	Konfiguracja generatora video
511 (\$1FF)	VSCL	Skalowanie video

gramowania Spin oraz tablice funkcji sinus oraz logarytmicznej i eksponentialnej. Można w niej także znaleźć definicje czcionek ekranowych w matrycy  $16 \times 32$  px. Czcionki te zawierają także znaki ułatwiające rysowanie schematów elektrycznych.

Wspólna pamięć RAM, określana jako *HUB RAM*, dostępna jest na równych prawach dla wszystkich ośmiu rdzeni. Odbywa się to w ten sposób, że każdy rdzeń otrzymuje do niej dostęp raz na 16 taktów zegara systemowego.

Każdy z rdzeni ma również dwa liczniki o rozbudowanych funkcjach, umożliwiające między innymi implementację przetworników cyfrowo-analogowych (przez sterowanie współczynnikiem wypełnienia sygnału prostokątnego) oraz analogowo-cyfrowych typu delta-sigma. Rozdzielczość tego rodzaju przetwarzania jest rzędu 10 bitów dla częstotliwości akustycznych. Rdzenie wyposażone są również w jednostkę video. Może być ona wykorzystana do generowania sygnału video w standardzie PAL i NTSC, jak również sygnału VGA. Generowany obraz może składać się z czterech kolorów z palety ponad 100 (wyjście TV) oraz 64 (VGA). W przypadku, gdy sterownik video nie jest wykorzystywany do tworzenia obrazu, można go wykorzystywać w celu szeregowej transmisji danych albo generowania sygnału prostokątnego o zmiennym współczynnikiem wypełnienia (PWM).

Zarówno główny zegar systemowy, jak i częstotliwości niezbędne do pracy liczników i układów video generowane są przy użyciu układów PLL. Standardową częstotliwością zegara systemowego jest 80 MHz. Uzyskiwana jest ona przy użyciu układu PLL stabilizowanego generatorem kwarcowym o częstotliwości 5 MHz. Ponieważ jedna instrukcja wykonywana jest w czterech taktach zegara systemowego, maksymalna wydajność jednego rdzenia pracującego ze standardową prędkością to 20 MIPS, a całego mikrokontrolera – 160 MIPS. Mikrokontroler pracuje

stabilnie również z częstotliwością 100 MHz, pozwalając dzięki temu osiągnąć wydajność do 200 MIPS.

Propeller nie ma mechanizmu przerwań; reakcje na zdarzenia zewnętrzne obsługiwane są tu na zasadzie oczekiwania na zdarzenie. Instrukcje typu WAIT czekają na zdarzenie, wprowadzając rdzeń w stan uśpienia, w którym pobiera on minimalny prąd zasilania, rzędu mikroamperów. Po wystąpieniu zdarzenia rdzeń wznawia pracę w ciągu dwóch taktów zegara systemowego, podejmując natychmiast obsługę tego zdarzenia. Tego rodzaju podejście mogło zostać zastosowane ze względu na wielordzeniowość układu: jeden lub kilka rdzeni można oddelegować do obsługi zdarzeń, podczas gdy inne wykonują nieprzerwanie swój program zapewniając przewidywalność czasu wykonania poszczególnych zadań.

Podstawowym językiem programowania mikrokontrolera Propeller jest język wysokiego poziomu Spin. Język ten jest językiem strukturalnym z elementami obiektowości, łatwym do nauczenia się i prostym w użyciu, a przede wszystkim dopasowanym do architektury mikrokontrolera. Dostępne są również kompilatory języka C/C++, a także wielu innych (Basic, Forth). W celu tworzenia szybko działającego kodu dla poszczególnych rdzeni używany jest również assembler.

### 2.3. Struktura systemu

Opracowany system składa się z modułu sterownika audio i video, współpracującym z pamięcią SRAM o pojemności 2 MB, organizacji 1 M słów 16-bitowych i czasie dostępu 10 ns. Dodatkowo kontroler pamięci SDRAM zapewnia dostęp do dużej pamięci roboczej o pojemności 128 MB. Moduły te zostały napisane w języku Verilog, a następnie połączone w całość za pomocą edytora graficznego programu Quartus.

#### 2.3.1. Podsystem graficzny

Zarówno pojemność, jak i czas dostępu do pamięci SRAM umożliwia jej wykorzystanie w celu wyświetlania grafiki o wysokiej rozdzielczości i jednocześnie dużej głębi kolorów. Rozdzielczość  $1920 \times 1200$  px przy 24-bitowej głębi kolorów wymaga bufora ramki o pojemności większej niż 6 MB. Także moc obliczeniowa wybranego mikrokontrolera nie jest wystarczająca do współpracy z pamięcią ekranu tej wielkości. Wysoka rozdzielczość jest jednak niezbędna do wizualizacji danych dotyczących sygnałów akustycznych, takich jak kształt fali bądź charakterystyka widmowa.

W tej sytuacji głębia kolorów przy rozdzielczości  $1920 \times 1200$  px ograniczona została do czterech. Aktywny obszar ekranu ma rozdzielczość  $1792 \times 1120$  px i otoczony jest ramką jednolitego koloru. Kolor ten może być dowolnie wybrany poprzez zapis odpowiedniej wartości do przeznaczonego w tym celu rejestru. Rozwiązanie takie zostało przyjęte w celu zapewnienia czasu potrzebnego na wykonanie przez podsystem graficzny niezbędnych operacji – czas standardowego wygaszania poziomego okazał się na to zbyt krótki. Istnieje także możliwość ograniczenia poziomej rozdzielczości wyświetlania do 896 px lub 448 px – możliwe jest wtedy uzyskanie 4-bitowej (16 kolorów) lub 8-bitowej (256 kolorów) głębi koloru. Kolory te wybierane są z 24-bitowych rejestrów palety. System posiada 1024 takie rejestry, pozwalając na wybór – w zależności od rozdzielczości – jednego z 4, 64 lub 256 banków kolorów. Dostępnych jest łącznie 12 trybów graficznych o różnej rozdzielczości i głębi kolorów (tab. 2).

Aby odciążać mikrokontroler od konieczności programowej obsługi niewielkich ruchomych obiektów graficznych, takich jak kursor myszy, istnieje możliwość zdefiniowania do sześciu obiektów (*sprites*) o rozmiarach  $32 \times 32$  px, których wyświetlanie obsługiwane jest sprzętowo, a mikrokontroler musi jedynie wpisać do odpowiednich rejestrów ich aktualną pozycję na ekranie.

W związku z ubogą głębią kolorów wprowadzony został również mechanizm listy wyświetlania (*display list*), który pozwala na zmianę głębi i palety kolorów poczawszy od dowolnej współrzędnej pionowej ekranu bez angażowania w tym celu mikro-

Tab. 2. Tryby graficzne systemu

Tab. 2. System's graphic modes

Tryb	Rozdzielczość pozioma	Rozdzielczość pionowa	Liczba kolorów
0	1792	1120	4
1	1792	560	4
2	1792	280	4
3	1792	140	4
4	896	1120	16
5	896	560	16
6	896	280	16
7	896	140	16
8	448	1120	256
9	448	560	256
10	448	280	256
11	448	140	256

kontrolera. Zmian takich można dokonywać wielokrotnie. Dodatkowo możliwa jest jednorazowa zmiana palety kolorów począwszy od podanej współrzędnej poziomej. Wszystkie te mechanizmy pozwalają na uzyskanie dużych możliwości wizualizacji przy posługiwaniu się niewielkim (do 512 kB) buforem ramki i mikrokontrolerem o stosunkowo niedużej mocy obliczeniowej.

Podsystem graficzny współpracuje z 30-bitowym przetwornikiem video DAC ADV7123. Ze względu na brak odpowiednich połączeń na płycie uruchomieniowej możliwe jest wykorzystanie tylko 8 z 10 bitów dla każdej składowej koloru.

### 2.3.2. Podsystem dźwięku

Podsystem dźwięku umożliwia odtwarzanie sygnałów dźwiękowych oraz wspomaganą sprzętowo syntezę przebiegów o dowolnym kształcie fali. Możliwe jest również bezpośrednie przesyłanie próbek z mikrokontrolera do przetwornika cyfrowo-analogowego oraz pobieranie próbek z przetwornika analogowo-cyfrowego.

Do odtwarzania sygnałów dźwiękowych służy bufor, którego zadaniem jest buforowanie danych przekazywanych przez mikrokontroler do wyjścia audio bez konieczności natychmiastowej reakcji na żądanie nowej próbki przez układ przetwornika DAC. Bufor ten mieści 1024 próbki (ok. 23 ms dźwięku przy częstotliwości próbkowania 44 100 Hz) i podzielony jest na dwa banki po 512 próbek, tak aby w czasie, gdy wysyłane są na wyjście audio dane z jednego banku, mikrokontroler mógł wypełnić drugi z nich nowymi danymi.

Cztery kanały syntezy dźwięku umożliwiają generację przebiegu o dowolnym kształcie i częstotliwości. Po wprowadzeniu do pamięci RAM danych o kształcie jednego okresu fali można określić częstotliwość i amplitudę generowanego przebiegu. Ponieważ długość definicji takiego przebiegu jest ograniczona wyłącznie pojemnością dostępnej pamięci SRAM, kanały te można wykorzystać również jako bufor dla sygnału dźwiękowego odtwarzanego z pamięci masowej.

Dodatkowo podsystem dźwięku zawiera również pojedynczy rejestr, do którego mikrokontroler może na bieżąco wpisać wartość próbki sygnału, a z którego pobrana ona zostanie przez przetwornik cyfrowo-analogowy. Umożliwia to programową syntezę dowolnego rodzaju sygnału dźwiękowego.

Sygnał, wprowadzony na wejście audio zestawu DE2-115 przetwarzany jest do postaci cyfrowej, a bieżąca próbka umieszczana

jest w rejestrze, skąd może zostać pobrana przez mikrokontroler. Zarówno żądanie nowej danej, jak i sygnał o tym, że została przetworzona nowa próbka danych wejściowych, są sygnalizowane mikrokontrolerowi poprzez jedną z jego linii wejściowych.

Podsystem dźwięku współpracuje z układem audio WM8731. Układ ten może pracować z maksymalną częstotliwością próbkowania 96 kHz i rozdzielczością 24 bity. Tryb pracy układu programowany jest przy użyciu magistrali I<sup>2</sup>C.

### 2.3.3. Pamięć SDRAM

Pamięć SDRAM o pojemności 32M słów 32-bitowych sterowana jest przez moduł sterownika, którego kod w języku Verilog zawarty jest na płycie CD znajdującej się w zestawie wraz z płytką uruchomieniową DE2-115. Moduł ten, pierwotnie przystosowany do współpracy z magistralą Avalon, został uzupełniony o elementy umożliwiające jego współpracę z pozostałymi częściami zaprojektowanego systemu.

### 2.3.4. Współpraca podsystemów z mikrokontrolerem

Mikrokontroler Propeller wyposażony jest w 32-bitowy, dwukierunkowy Port A, którego zadaniem jest współpraca z urządzeniami zewnętrznymi. Liczba ta nie pozwala na komfortową jego współpracę z opisanym powyżej systemem. W związku z tym rdzeń mikrokontrolera uzupełniony został o Port B. Port ten, rozszerzony do 96 bitów – 64 bitów wyjściowych i 32 bitów wejściowych – przeznaczony został w całości do wykorzystania jako główna magistrala systemowa.

Oryginalny mikrokontroler Propeller nie ma w swoim zestawie instrukcji operacji mnożenia. Możliwość szybkiego mnożenia jest jednak bardzo przydatna w zadaniach związanych z przetwarzaniem sygnałów i w związku z tym instrukcja ta została dodana do rdzenia wykorzystanego w systemie. Ponieważ układ FPGA Cyclone IV dysponuje szybkimi specjalizowanymi jednostkami mnożącymi, instrukcja mnożenia wykonuje się w tak zmodyfikowanym mikrokontrolerze w tym samym czasie (4 takty zegara systemowego), co pozostałe instrukcje arytmetyczno-logiczne.

W obu wspomnianych przypadkach rozszerzenia możliwości mikrokontrolera zostały wcześniej przewidziane przez producenta, a w związku z tym do programowania tak zmodyfikowanego mikrokontrolera można używać niezmodyfikowanych, oryginalnych narzędzi (*toolchain*) dostarczanych bezpłatnie przez firmę Parallax.

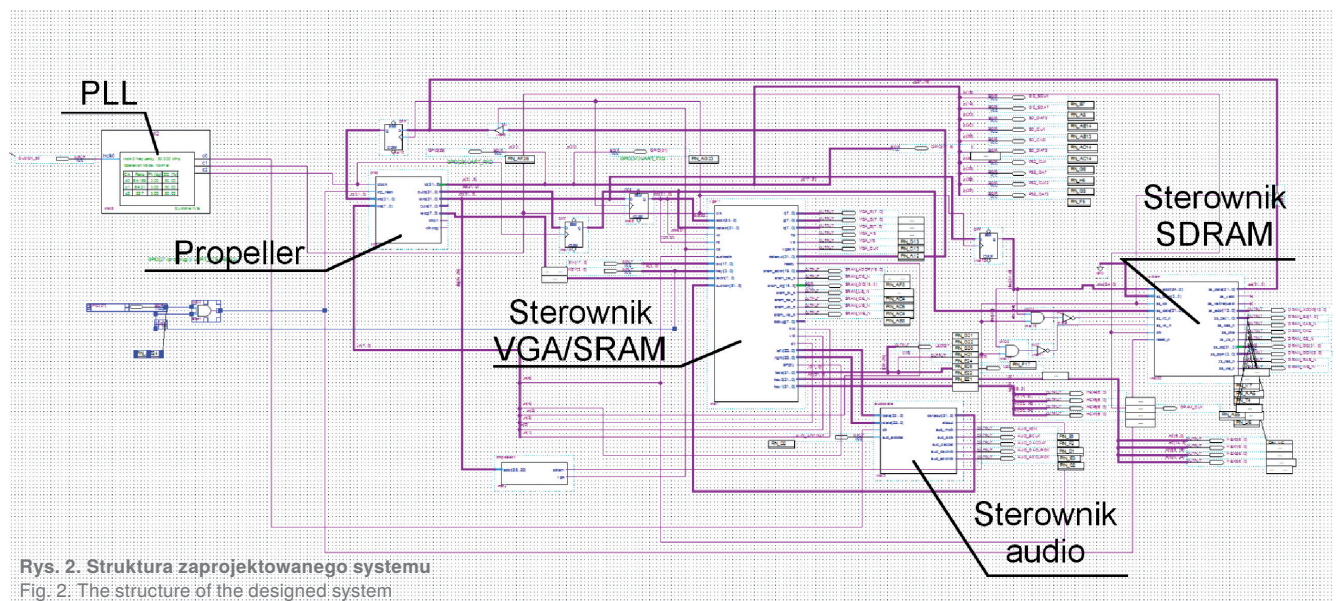
Ze względu na zaimplementowany w systemie rozbudowany podsystem graficzny, z rdzenia mikrokontrolera został usunięty niewykorzystywany układ generatora video, co zmniejszyło (o ponad 2000 jednostek logicznych) zapotrzebowanie systemu na zasoby FPGA.

Zegary taktujące dla poszczególnych podsystemów generowane są przez jednostkę PLL znajdującą się wewnątrz układu FPGA. Sygnałem wejściowym jest sygnał o częstotliwości 50 MHz z generatora kwarcowego. Częstotliwości wyjściowe zostały wybrane na podstawie następujących założeń:

- układ WM8731 pracuje na częstotliwości 16,9344 MHz, przy czym musi być ona jak najdokładniej odwzorowana, zależy bowiem od niej bezpośrednio częstotliwość próbkowania,
- rdzeń mikrokontrolera Propeller pracuje stabilnie w wybranym układzie FPGA z częstotliwością nie większą niż 120 MHz,
- standardowy zegar dla rozdzielczości 1920×1200 px wynosi 154 MHz,
- poszczególne częstotliwości powinny być ze sobą zsynchronizowane, aby zapewnić lepsze warunki współpracy poszczególnych podsystemów w jednym układzie.

W rezultacie zostały wybrane częstotliwości 16,93 MHz dla podsystemu dźwięku, 152,38 MHz dla podsystemu video oraz 114,29 MHz dla mikrokontrolera.

Bezpośrednio do portów mikrokontrolera podłączone zostały: gniazdo klawiatury i myszy PS2 (aby podłączyć jednocześnie oba urządzenia, wymagany jest dodatkowy przewód PS2 typu Y),



złącze karty pamięci SD oraz magistrala I<sup>2</sup>C do sterowania układem WM8731.

Całość systemu zajmuje ok. 26 000 jednostek logicznych (*logic elements, LE*), pozostawiając około 90 000 LE dla innych zastosowań. Jednostki te zostaną wykorzystane w celu implementacji zaawansowanych algorytmów przetwarzania sygnałów.

Strukturę zaprojektowanego systemu przedstawiono na rys. 2.

## 2.4. Oprogramowanie

Zaprojektowany system może być programowany przy użyciu jednego ze standardowych narzędzi dla mikrokontrolera Propeller.

Do obsługi urządzeń peryferyjnych, takich jak klawiatura, mysz, karta SD wraz z systemem plików FAT32 oraz magistrali I<sup>2</sup>C dostępne są gotowe biblioteki procedur.

W celu obsługi podsystemu audio i video napisana została w języku Spin i assemblerze biblioteka procedur pozwalająca na wykonywanie podstawowych operacji graficznych (czyszczenie ekranu, rysowanie podstawowych figur geometrycznych, wyprowadzanie tekstu), jak również obsługę podsystemu dźwiękowego, a także dostęp do pamięci RAM.

Wymienione moduły pozwalają na tworzenie programów użytkowych na tak opracowany system. Gotowy, skompilowany w komputerze PC program może być wprowadzony do mikrokontrolera poprzez łącze szeregowo. Płytkę uruchomieniową DE2-115 ma wejście RS-232; można także zaprogramować system poprzez łącze GPIO, używając podłączonego do portu USB komputera PC konwertera USB-RS-232.

Aby uniknąć konieczności programowania systemu przez łącze szeregowo po każdym jego włączeniu, opracowany został prosty program ładujący. Program ten został skompilowany, a następnie na podstawie pliku wynikowego wygenerowane zostały pliki definicji pamięci, które zostały dodane do projektu systemu w programie Quartus. Dzięki temu program ten uruchamia się bezpośrednio po włączeniu systemu, po czym wczytuje i uruchamia właściwy program użytkowy zapisany na karcie SD.

## 3. Uruchomienie i testy systemu

W celu testowania opracowanego systemu zostały napisane w języku Spin programy, testujące poszczególne jego elementy. Najwięcej problemów podczas uruchamiania wiązało się z pracą podsystemu graficznego. Podsystem ten, pracujący z wysoką częstotliwością (ponad 150 MHz) okazał się szczególnie czuły na błędy projektowania, prowadzące do zbyt długich czasów

propagacji sygnału wewnątrz układu FPGA. Poszczególne jego elementy musiały więc zostać zoptymalizowane tak, aby uzyskać poprawną i stabilną pracę tego podsystemu.

Podobne problemy wynikły również podczas opracowywania podsystemu obsługującego pamięć SRAM. Czas dostępu do tej pamięci, wynoszący nominalnie 10 ns w rzeczywistości jest przedłużony o czas propagacji sygnału wewnątrz układu FPGA i po uwzględnieniu tej poprawki wynosi ponad 16 ns. W zaprojektowanym systemie czas jednej operacji na pamięci SRAM wynosi ok. 20 ns.

Konieczność współpracy mikrokontrolera Propeller ze stosunkowo powolną magistralą systemową pozbawiła ten mikrokontroler części jego zalet związanych z wielozadaniowością – tylko jeden rdzeń może mieć dostęp do tej magistrali bez powodowania konfliktów. Pozostałe rdzenie mogą wykonywać inne zadania niezwiązane z jej obsługą. W takiej sytuacji niemożliwe jest jednak wykorzystanie wbudowanego mechanizmu oczekiwania na zdarzenia, a jednocześnie brak jest mechanizmu przerwania.

W tej sytuacji okazało się, że bufor audio, który w pierwszych wersjach systemu miał pojemność 256 próbek, musi zostać powiększony. Pozbawiony systemu przerwania mikrokontroler nie był w stanie dostarczyć na czas niezbędnych danych. W kolejnych wersjach systemu bufor ten został powiększony do 1024 próbek, co dało Propellerowi wystarczający margines czasu na jego obsługę.

W trakcie testowania dokonane zostały również inne poprawki, mające na celu poprawę działania i zwiększenie możliwości systemu.

## 4. Możliwości zastosowań

Przedstawiony system został opracowany jako podstawa większego systemu do przetwarzania sygnałów dźwiękowych. Ze względu na swoje cechy może być on jednak wykorzystany także do innych celów. Obsługa monitorów o wysokiej rozdzielczości pozwala na jego wykorzystanie wszędzie tam, gdzie potrzebna jest wizualizacja dużej liczby danych. Może on zostać również wykorzystany w celach hobbystycznych – mikrokontroler Propeller umożliwia emulację mikroprocesorów i układów towarzyszących używanych w mikrokomputerach produkowanych w latach osiemdziesiątych ubiegłego wieku. Dostępne są programy umożliwiające emulację na mikrokontrolerze Propeller mikroprocesorów 8080, Z80, 6502, a także układów dźwiękowych: MOS6581, znanego także jako SID oraz AY-3-8910. Program emulujący

układ SID został wykorzystany podczas testowania podsystemu dźwiękowego; układ ten, w jego emulowanej przez mikrokontroler wersji może zostać również wykorzystany w celu syntezy i filtracji sygnałów dźwiękowych.

Rdzeń mikrokontrolera Propeller zajmuje około 16 000 jednostek logicznych układu Cyclone IV. Możliwa jest więc implementacja tego mikrokontrolera również na znacznie mniejszych platformach, takich jak DE0-nano. Ta płytkę, wielkości karty kredytowej, cenowo zbliżoną do płytek uruchomieniowych z mikrokontrolerami, po zaimplementowaniu kodu mikrokontrolera Propeller może służyć jako uniwersalny sterownik o nieporównywalnej z układami typu AVR czy PIC elastyczności. Możliwe jest również obniżenie zapotrzebowania na zasoby FPGA przez usunięcie zbędnych z punktu widzenia danego projektu podjednostek (liczniki, generator video) lub obniżenie liczby rdzeni.

System, w nieco zmienionej wersji, był również testowany na płytce uruchomieniowej DE1-SoC. Wyposażona w układ FPGA nowszej generacji – Cyclone V – pozwalała na pracę rdzenia mikrokontrolera Propeller z częstotliwością do 180 MHz, a jej układy PLL były w stanie dostarczyć żądanych częstotliwości ze znacznie większą precyzją.

## 5. Podsumowanie

Opracowany system został zaimplementowany i przetestowany. Wyniki testów wskazują na jego przydatność do założonych celów, wskazując także na możliwość projektowania takich systemów również do innych zadań.

Dostępność kodu mikrokontrolera, który jest wciąż produkowany i wspierany przez producenta, jak również posiada bogatą bibliotekę oprogramowania, wraz ze spadkiem cen i wzrostem dostępności układów FPGA otwiera nowe możliwości projektowania urządzeń opartych na logice programowalnej. Urządzenia takie cechują się ogromną elastycznością i możliwością dokładnego dostosowania ich parametrów do potrzeb docelowego użytkownika.

## Podziękowania

Publikację sfinansowano w ramach pracy własnej W/WE/8/2013.

## Bibliografia

1. Majewski J., Zbysiński P., *Układy FPGA w przykładach*, BTC, Warszawa 2007.
2. Pawluczuk A., *Układy programowalne dla początkujących*, BTC, Warszawa 2010.
3. Zbysiński P., Pasierbiński J., *Układy programowalne*, BTC, Warszawa 2004.
4. Chu P.P., *FPGA Prototyping by Verilog Examples*, Wiley, 2008.
5. Tala D.K., *Verilog Tutorial*, dokument elektroniczny, www.asic-world.com.
6. *Propeller Manual, version 1.2*, Parallax Semiconductor, Rocklin 2011.
7. *Propeller P8X32A Datasheet, rev. 1.4*, Parallax Semiconductor, Rocklin 2011.

# System with a Programmable Microcontroller Propeller

**Abstract:** The paper presents the system on FPGA programmable chip using the Propeller P8X32A microcontroller. The microcontroller code was published in the August 2014 on the GPL v. 3 open source license. The system contains graphics and sound drivers, as well as the keyboard, mouse, and storage drivers. It is designed for applications involving the processing of audio signals. The paper presents the structure of the designed system and its possible applications.

**Keywords:** FPGA, microcontroller, SoPC, HDL, Verilog

## mgr inż. Piotr Kardasz

pik@we.pb.edu.pl

Doktorant na Wydziale Elektrycznym Politechniki Białostockiej. Zajmuje się badaniami związanymi z inteligentnymi algorytmami kompresji, rekonstrukcji i przetwarzania sygnałów.

