

FPGA programowanie w języku C

C oraz C++ to nadal jedne z najczęściej stosowanych języków programowania. Ich popularność jest tak duża, iż mimo rozwoju nowych technologii nadal są one powszechnie wykorzystywane do tworzenia programów dla nowych rozwiązań sprzętowo-programowych. Takim rozwiązaniem, istotnie różniącym się od typowego procesora, z jakim stykamy się w codziennej pracy, jest technologia FPGA.

Przez długi czas do tworzenia programów dla FPGA, a raczej do opisu zachowania się takiego układu stosowano tzw. języki opisu sprzętu (HDL – hardware description language), tj. głównie VHDL oraz Verilog. Jednakże od kilku lat, ze względu na powszechną znajomość składni języka C, pojawiła się też jego odmiana o nazwie SystemC, która jest odpowiednikiem VHDLa oraz Veriloga.

Proces adaptacji języka C do technologii FPGA postępuje nadal i obecnie możemy już względnie swobodnie przenosić typowy kod w języku C do układów FPGA. Wykorzystanie języka C w ten sposób jest nazywane syntezą wysokiego poziomu (ang. HLS – high level synthesis). W wielu przypadkach tworzymy niemal tradycyjny kod C/C++ (choć są też pewne ważne ograniczenia, np. brak rekurencji), który jest przekształcany do języków opisu sprzętu, np. do języka VHDL.

Prezentowane w tym artykule rozwiązanie o nazwie Vivado HLS nie jest pierwszym tego typu narzędziem do syntezy wysokopoziomowej (zostało oparte o wcześniejszy pakiet o nazwie AutoESL), jednakże firma Xilinx odpowiedzialna za rozwój środowiska Vivado zdecydowała się, aby ich środowisko syntezy wysokiego poziomu – Vivado HLS – stało się dostępne dla wszystkich. Od końca 2015 roku, w wersji darmowej Vivado HL WebPACK™ Edition, pakiet Vivado HLS stał się dostępny już dla wszystkich użytkowników środowiska Vivado.

ZAŁOŻENIA DO PROJEKTU

Przed wykonaniem jakiegokolwiek projektu jak zawsze warto przygotować założenia, które chcemy zrealizować. Nasze zadanie polega na tym, iż chcemy przygotować niewielki system z mikroprocesorem oraz układem generującym obraz o rozdzielczości 320 na 240. Przy czym niska rozdzielczość to konsekwencja użycia nieco starszego układu FPGA do implementacji naszego projektu. Mamy stosunkowo małą ilość tzw. pamięci BRAM. Ponieważ jednak nie chcemy zbytnio skomplikować naszego projektu, będziemy stosować tylko pamięć BRAM, zamiast np. pamięci typu DDR (o znacznie wyższych pojemnościach, ale i bardziej skomplikowanej obsłudze).

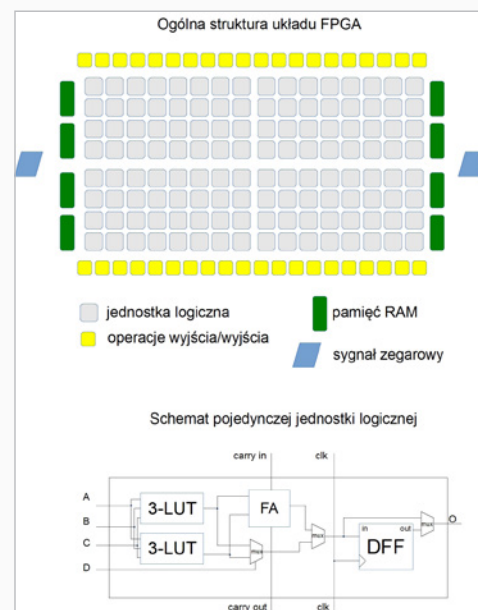
Wykonanie tego typu projektu w językach opisu sprzętu VHDL czy Verilog byłoby dość czasochłonne, szczególnie w przypadku implementacji mikroprocesora. Wymagałoby to od nas większego wysiłku, a w szczególności sporej ilości czasu, aby uzyskać właściwą implementację. Dlatego sam mikroprocesor zaprojektujemy za pomocą narzędzia Vivado HLS. Sprowadzi się to do napisania kilku funkcji w języku C, które będą implementować nieskomplikowany procesor.

Generację obrazu zgodnego ze standardem VGA będzie łatwiej zrealizować w języku typu HDL, dlatego tę część napiszemy w VHDL. Będzie trzeba też dopisać jeszcze jeden moduł w VHDLu, który połączy system generacji obrazu z mikroprocesorem, lecz ograniczy się to tylko do deklaracji odpowiednich komponentów.

Ostatnim założeniem jest wybór układu FPGA, dla którego przygotowujemy implementację. Będzie to układ Spartan 3 produkowany przez

Czym są układy FPGA?

Podstawowym elementem w układzie FPGA jest tzw. jednostka logiczna (JK). Każda JK zawiera kilka typów wejść oraz wyjść. Obecne są sygnały wejściowe (np. dla układu Spartan 3 JK posiada cztery wejścia, ale nowsze wersje układów FPGA mogą posiadać JK o sześciu, a nawet o ośmiu wejściach). Dostępne jest też wejście/wyjście zegarowe. Jednak sygnał wyjściowy może być tylko jeden, choć wyjściem może też być tzw. wartość przeniesienia. Zadaniem JK jest realizacja pewnej funkcji logicznej, przy czym postać tej funkcji określa się w momencie programowania (konfiguracji) układu. Schemat pojedynczej jednostki oraz całego układu FPGA przedstawia się następująco:



Rysunek 1. Ogólna budowa układów FPGA oraz schemat pojedynczej jednostki logicznej

Oprócz jednostek logicznych wszystkie współczesne układy FPGA posiadają pewną ilość pamięci RAM (nazywaną pamięcią BRAM), obecnie są to już nawet megabajty, ale używany w artykule układ ma około 52 kB dostępnej pamięci BRAM.

W każdym układzie FPGA dostępne są też bloki realizujące operacje wejścia/wyjścia oraz bloki związane z przetwarzaniem sygnału zegarowego. Obecność tych wszystkich elementów oznacza, iż można zmienić przeznaczenie układów FPGA. Możliwe staje się tworzenie prototypów innych układów scalonych – i to wielokrotnie za pomocą jednego układu FPGA. Obecnie układy FPGA są szeroko wykorzystywane także w innych celach. Ważną cechą układów FPGA jest ich wydajność, dzięki elastycznej strukturze można uzyskać implementację cechującą się małym zużyciem energii oraz wysoką wydajnością, przy stosunkowo niskich taktowaniach zegara. Ta ostatnia własność spowodowała, iż układy FPGA są wykorzystywane nawet jako akceleratorzy obliczeniowe, oferujące wyższą wydajność niż wyżej taktowane układy CPU oraz GPU z kart graficznych.