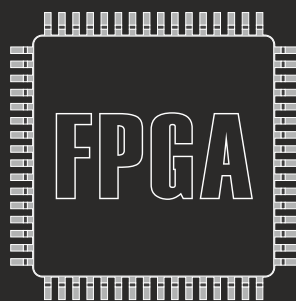




Projektowanie systemów wbudowanych w układach FPGA

Valery Salauyou
Adam Klimowicz



Przedmowa

Książka omawia projektowanie bloków funkcjonalnych na *układach programowalnych* (field programmable gate array – FPGA), które są najczęściej używane w *systemach wbudowanych* (embedded systems). Wszystkim prezentowanym tematom towarzyszy duża liczba przykładów opisanych w języku Verilog. Do zademonstrowania wyników syntezy i symulacji rozpatrywanych przykładów został wykorzystany system Quartus Prime w wersji 18.1, a płyta ewaluacyjna Intel DE1-SoC została wykorzystana do fizycznej implementacji. Jednak materiał w książce nie koncentruje się na FPGA konkretnej firmy lub konkretnym systemie projektowania. Aby opanować materiał zaprezentowany w monografii, wystarczy żeby użyte narzędzie projektowe wspierało język wejściowy Verilog przy opisie projektów. Wszystkie prezentowane przykłady można także zaimplementować na układach FPGA Xilinx przy użyciu systemów ISE lub Vivado.

Osobliwością książki jest to, że każdy rozważany problem jest kompleksowo badany i proponowane są najskuteczniejsze metody jego rozwiązania. Omówienie wielu zagadnień podsumowane jest badaniami eksperymentalnymi, na podstawie których formułowane są zalecenia dotyczące najbardziej efektywnego wykorzystania różnych podejść do projektowania konkretnej jednostki lub bloku. Realizacja zadań zaproponowanych w książce przyczynia się do lepszego przyswojenia materiału, a także do zdobycia nieocenionego doświadczenia w projektowaniu bloków systemów wbudowanych na układach FPGA.

Książka oparta jest na kursach wykładowych prowadzonych przez autora przez wiele lat dla studentów Politechniki Białostockiej i dla profesjonalnych projektantów systemów wbudowanych oraz na specjalnych seminariach, przeznaczonych dla profesjonalnych projektantów systemów wbudowanych na specjalnych seminariach.

Monografia składa się z 12 rozdziałów, które są tematycznie połączone w pięć części.

Pierwsza część poświęcona jest projektowaniu pamięci systemów wbudowanych. Żaden z nowoczesnych systemów wbudowanych nie może obejść się bez pamięci. Ograniczyliśmy się do rozpatrzenia problemów projektowania tylko pamięci wbudowanej zaimplementowanej w układach FPGA. Pamięć w systemach wbudowanych jest szeroko stosowana do przechowywania kodu programu, generowania złożonych funkcji matematycznych w postaci dużej *tablicy przeglądowej* (look-up table – LUT), pamięci podręcznej procesora, pamięci typu FIFO do buforowania danych sieciowych itp.

Rozdział 1 omawia główne typy pamięci systemów wbudowanych, podaje ich klasyfikację i pokazuje miejsce w tej klasyfikacji pamięci na układzie FPGA (pamięci wbudowanej). W rozdziale opisano również architekturę pamięci wbudowanej i przedstawiono zalecenia firmy Intel dotyczące jej projektowania.

Rozdział 2 omawia projektowanie pamięci wbudowanej za pomocą języka Verilog. Różne sposoby opisu pamięci podano na przykładzie pamięci jednoportowej. Pokazano cechy opisu pamięci dwuportowej i pamięci typu ROM. Osobno została rozpatrzona kwestia inicjalizacji pamięci. Bloki pamięci wbudowanej FPGA umożliwiają wydajną implementację rejestrów przesuwających z odczepami stosowanymi w aplikacjach do *cyfrowego przetwarzania sygnałów* (digital signal processing – DSP). Podano metody projektowania w języku Verilog pamięci, takich jak FIFO (First-In, First-Out) i LIFO (Last-In, First-Out). Przedstawiono również atrybuty syntezy używane do opisu pamięci w języku Verilog.

Moduły pamięci w systemach wbudowanych wykorzystujących układy FPGA można również budować przy użyciu bloków IP (Intellectual Property Core) i edytora parametrów systemu Quartus. W tym przypadku wykorzystywane są specjalne megafunkcje, opracowane przez firmę Intel. Megafunkcje wykorzystują wszystkie właściwości architektur zintegrowanych bloków pamięci FPGA. Rozdział 3 omawia tworzenie przy użyciu bloków IP systemu Quartus modułów pamięci typu RAM, ROM i FIFO, a także rejestrów przesuwających.

Druga część poświęcona jest projektowaniu układów sterowania. Twórca systemów wbudowanych staje przed zadaniem projektowania układów sterowania za każdym razem, gdy powstaje nowy projekt. W niektórych przypadkach, cały system wbudowany może składać się z jednego układu sterującego lub kilku systemów sterowania współpracujących ze sobą. Jako urządzenie sterowania tradycyjnie używane są *automaty skończone* (finite state machine – FSM).

Wiele urządzeń i systemów sterowania opiera się na zasadach sterowania mikroprogramowego. Szeroko stosowane w systemach wbudowanych *automaty mikroprogramowane* (AM) są zbudowane na tych samych zasadach. Języki sterowania logicznego służą do opisu działania AM, a jednym z nich jest język sieci działań. Rozdział 4 omawia metodę projektowania AM zgodnie z siecią działań, a także implementację na FPGA AM Mealy'ego, AM Moore'a i AM klasy C. Jako przykład zastosowania AM przedstawiono sprzętową implementację na FPGA synchronicznych układów mnożenia.

Rozdział 5 omawia nowy język graficzny do opisu automatów skończonych, zwany *algorytmicznym automatem skończonym* (algorithmic state machine – ASM) lub siecią działań algorytmu. Główną zaletą sieci działań algorytmu ASM w porównaniu z siecią działań jest to, że w języku Verilog automat skończony można bezpośrednio opisać poprzez jego reprezentację jako sieć działań ASM, bez szeregu formalnych etapów syntezy wykonywanych ręcznie, które często są źródłem trudnych do wykrycia błędów. Za pomocą sieci działań ASM można również opisać funkcjonowanie wspólnych modeli automatów skończonych, procesów równoległych i układów kombinacyjnych.

Rozdział 6 przedstawia nową technikę projektowania układów cyfrowych opartą na *sieci działań ASM ze ścieżką przetwarzania danych* (algorithmic state machine with datapath – ASMD) i *automatów skończonych ze ścieżką przetwarzania danych* (finite state machine with datapath – FSMD), która została nazwana techniką ASMD-FSMD. W przeciwieństwie do tradycyjnego podejścia, technika ASMD-FSMD umożliwia natychmiastowe opisanie całego układu w języku Verilog bez dzielenia go na urządzenie operacyjne i urządzenie sterowania (kontroler – controller), co znacznie skraca czas i koszty projektowania, a także zwiększa niezawodność opracowanych projektów.

Trzecia część poświęcona jest projektowaniu procesorów wbudowanych. Nowoczesne systemy wbudowane często zawierają procesory zwane procesorami wbudowanymi (embedded processors), służące do wykonywania programów i algorytmów.

Rozdział 7 omawia ogólne problemy projektowania procesorów wbudowanych na układach FPGA: architekturę tradycyjną dla różnych list rozkazów procesorów, architekturę oraz sposób projektowania listy rozkazów dla procesorów PIC, podstawowe struktury procesorów RISC (reduced instruction set computer), a także metodologię projektowania procesorów na układach FPGA.

Rozdział 8 przedstawia proces projektowania jednordzeniowego procesora PIC oraz szczegółowo opisuje metodykę projektowania układu operacyjnego. Osobno rozważa się projekt jednostki arytmetyczno-logicznej i układu sterującego. Kody źródłowe Verilog są pokazane dla komponentów układu procesora, a także modułu najwyższego poziomu. Szczególną uwagę zwraca się na debugowanie procesora PIC i ocenę jego wydajności; podano zalecenia dotyczące sposobu zwiększenia wydajności procesora.

Rozdział 9 przedstawia trzy metodyki projektowania procesorów wielocyklowych. Pierwsza technika polega na tradycyjnym podejściu do projektowania procesorów wielocyklowych i obejmuje sekwencyjne wprowadzanie stanów układu sterującego: do wybierania i dekodowania polecenia, implementacji niektórych grup instrukcji i poszczególnych instrukcji. Druga technika maksymalnie wykorzystuje układ operacyjny i urządzenie sterowania procesorem jednocyklowym. Aby to zrobić, automat skończony, który określa etapy wykonywania instrukcji, jest dodawany do układu procesora. Trzecia technika ma na celu zwiększenie liczby etapów wykonawczych procesora wielocyklowego. Jest to konieczne, aby poprawić wydajność w przypadku przetwarzania danych potokowych.

Czwarta część poświęcona jest projektowaniu bloków cyfrowych przetwarzania sygnałów. Jedną z wyróżniających cech nowoczesnych systemów wbudowanych jest możliwość realizacji szerokiego zakresu zadań z dziedziny cyfrowego przetwarzania sygnałów DSP. Wynika to z faktu, że metody DSP są coraz częściej stosowane w różnych obszarach działalności człowieka.

Czwartą część przedstawiono tylko w jednym rozdziale – 10, który omawia projektowanie filtrów cyfrowych na układach FPGA. Podano krótkie wprowadzenie do cyfrowego przetwarzania sygnałów i wskazano możliwości FPGA przy implementacji systemów DSP. Określono główne parametry filtrów cyfrowych oraz metody opisywania filtrów cyfrowych w języku Verilog. Przedstawiono ogólną metodykę projektowania filtrów cyfrowych, rozważono przykład opracowania filtra cyfrowego

w systemie MATLAB, oraz symulację filtrów w systemie ModelSim i implementację filtrów w systemie Quartus. W podsumowaniu, zbadano implementację różnych typów filtrów dla różnych rodzin FPGA.

Piąta część poświęcona jest projektowaniu podsystemu synchronizacji. Problem skutecznej implementacji podsystemu synchronizacji staje się szczególnie istotny przy opracowywaniu szybkich i złożonych projektów. W przeciwieństwie do projektów na układach ASIC (application-specific integrated circuit), projekty na układach FPGA nie muszą obejmować całego podsystemu synchronizacji. W FPGA wiele elementów podsystemu synchronizacji zostało już zaprojektowanych i wdrożonych na poziomie sprzętowym przez projektantów FPGA. Elementy te obejmują sieci synchronizacji, pętle synchronizacji fazowej PLL (phase-locked loop) i bloki sterowania sygnałem zegarowym. Ponadto kompilator systemu Quartus automatycznie rozwiązuje wiele problemów związanych z synchronizacją projektu.

Rozdział 11 omawia różne problemy, przed którymi stoi inżynier podczas projektowania podsystemu synchronizacji złożonego projektu na FPGA. Realizacja projektów synchronicznych na FPGA ma swoją własną charakterystykę, której znajomość jest niezbędna do skutecznego wykorzystania możliwości sprzętowych FPGA. W tym rozdziale omówiono zalecenia firmy Intel dotyczące projektowania projektów synchronicznych na układach FPGA, multipleksowania i bramkowania sygnałów zegara, zalecenia dotyczące opracowywania szybkich i złożonych logicznie projektów, projektowania sygnałów „reset”, a także atrybutów syntezy wpływających na synchronizację projektu.

Rozdział 12 opisuje bloki pętli synchronizacji fazowej PLL. Bloki PLL są zaprojektowane do tworzenia wysokiej jakości wewnętrznych sygnałów zegarowych o zadanej częstotliwości, współczynniku wypełnienia i przesunięciu fazowym na podstawie referencyjnych sygnałów zegarowych. W rozdziale omówiono zasady działania bloków PLL, architekturę i funkcjonowanie bloków PLL w FPGA rodziny Cyclone V, a także sposób konfiguracji bloków PLL w projektach FPGA.

Każdy rozdział kończy się wnioskami. Wnioski podsumowują główne punkty rozważane w danym rozdziale i prezentują najważniejsze rezultaty badań oraz skupiają czytelnika na najważniejszych zagadnieniach, które mogły zostać pominięte podczas czytania tego rozdziału. W książce zaprezentowano dużą liczbę przykładów oraz wyników ich syntezy i symulacji. Wszystkie przykłady są sprawdzone i mogą być używane w projektach systemów wbudowanych.

Monografia przeznaczona jest przede wszystkim dla studentów odpowiednich specjalności uczelni technicznych. Materiał w niej zawarty może być wykorzystywany przez nauczycieli do prowadzenia wykładów, ćwiczeń laboratoryjnych i zajęć praktycznych. Wiele zagadnień poruszonych w książce nie zostało jeszcze w pełni zbadanych, dlatego też mogą zainteresować młodych naukowców. Książka może być również wykorzystana przez inżynierów jako samouczek do studiowania różnych problemów związanych z projektowaniem systemów wbudowanych na FPGA. Niektóre tematy przedstawione w tej publikacji są całkiem nowe i mogą zainteresować nawet doświadczonych projektantów.

Badania zostały zrealizowane w ramach pracy nr WZ/WI-IIT/4/2020 w Politechnice Białostockiej i sfinansowane z subwencji badawczej przekazanej przez Ministra Edukacji i Nauki.