

Spis treści

Wykaz skrótów.....	13
Przedmowa.....	15
Część I. Projektowanie pamięci.....	21
1. Bloki pamięci wbudowanej.....	22
1.1. Pamięć systemów wbudowanych.....	22
1.2. Pamięć wbudowana (pamięć FPGA)	23
1.2.1. Zalety i wady pamięci wbudowanej	23
1.2.2. Zakres zastosowania pamięci wbudowanej.....	24
1.3. Architektura pamięci wbudowanej	25
1.3.1. Typy pamięci wbudowanej	25
1.3.2. Typy pamięci wbudowanej układów FPGA rodziny Cyclone V	27
1.3.3. Tryby pracy pamięci wbudowanej.....	28
1.3.4. Bity parzystości w blokach pamięci wbudowanej	34
1.3.5. Tryby synchronizacji pamięci wbudowanej.....	34
1.3.6. Tryby działania wyjść pamięci	36
1.3.7. Konfiguracja portu mieszanego.....	38
1.3.8. Zezwolenie na zapis bajtów danych	40
1.3.9. Obsługa trybu spakowanego.....	40
1.3.10. Zezwolenie synchronizacji adresu.....	41
1.4. Zalecenia dotyczące projektowania pamięci wbudowanej.....	41
1.4.1. Wybór bloku pamięci.....	41
1.4.2. Zewnętrzne rozwiązywanie konfliktów	42
1.4.3. Konfigurowanie trybu działania wyjść pamięci	42
1.4.4. Stan i inicjalizacja pamięci po włączeniu zasilania	42
1.4.5. Sterowanie synchronizacją w celu zmniejszenia zużycia energii	43
1.5. Wnioski	43
2. Opis pamięci wbudowanej w języku Verilog.....	46
2.1. Metody opisu pamięci na przykładzie jednoportowej pamięci typu RAM (Single-Port RAM) z jednym sygnałem synchronizacji.....	46

2.1.1. Deklaracja macierzy pamięci	46
2.1.2. Tryby działania wyjść pamięci	47
2.1.3. Opis pamięci ze „starymi danymi” na wyjściu	47
2.1.4. Opis pamięci z „nowymi” danymi na wyjściu	52
2.1.5. Opis pamięci z wyjściami kombinacyjnymi.....	56
2.1.6. Wstawianie rejestrów na wejściach adresowych pamięci	58
2.2. Inicjalizacja pamięci.....	60
2.3. Jednoportowa pamięć z dwoma sygnałami synchronizacji	63
2.4. Prosta dwuportowa pamięć typu RAM (Simple Dual-Port RAM).....	64
2.5. Rzeczywista dwuportowa pamięć typu RAM (True Dual-Port RAM)	66
2.6. Pamięć typu ROM (Read-Only Memory).....	68
2.7. Implementacja rejestrów przesuwających w blokach pamięci wbudowanej FPGA	71
2.8. Pamięć typu FIFO.....	75
2.9. Pamięć typu LIFO.....	79
2.10. Używanie atrybutów syntezy podczas opisywania pamięci w języku Verilog	82
2.11. Wnioski	85
3. Projektowanie pamięci z wykorzystaniem bloków IP i edytora parametrów systemu Quartus	89
3.1. Katalog IP do tworzenia modułów pamięci.....	89
3.2. Tworzenie modułów pamięci typu RAM i ROM	90
3.3. Tworzenie modułu pamięci typu FIFO	99
3.4. Tworzenie rejestru przesuwającego w bloku pamięci wbudowanej FPGA	103
3.5. Wnioski	105
Część II. Projektowanie układów sterowania	107
4. Projektowanie automatów mikroprogramowanych	108
4.1. Mikroprogramowanie i automaty mikroprogramowane	108
4.2. Reprezentacja AM. Sieci działań	110
4.3. Projektowanie automatów mikroprogramowanych zgodnie z siecią działań (GSA).....	112
4.3.1. Metodyka projektowania AM zgodnie z GSA	112
4.3.2. Oznaczanie GSA	113
4.3.3. Tworzenie tablicy przejść.....	116
4.3.4. Dodatkowe oznaczenia GSA i pseudorównoważne AM.....	118
4.4. Implementacja AM na FPGA	119

4.4.1. Implementacja AM Mealy'ego	119
4.4.2. Implementacja AM Moore'a.....	120
4.4.3. Analiza funkcjonowania AM Mealy'ego i AM Moore'a.....	122
4.4.4. Implementacja AM Moore'a klasy C.....	124
4.5. Wykorzystanie AM przy realizacji synchronicznych układów mnożących	126
4.5.1. Implementacja układu mnożącego za pomocą algorytmu a.....	127
4.5.2. Projektowanie układu operacyjnego	128
4.5.3. Projektowanie układu sterującego	130
4.5.4. Implementacja układu mnożącego na FPGA	132
4.5.5. Porównanie różnych sposobów implementacji synchronicznych układów mnożących	137
4.6. Wnioski	139
5. Algorytmiczne automaty skończone (ASM)	142
5.1. Problemy GSA przy opisie algorytmów działania sprzętu.....	142
5.2. Język ASM.....	143
5.3. Metodyka budowy ASM	145
5.4. Przykład budowy ASM dla automatu Moore'a.....	146
5.5. Przykład budowy ASM dla automatu Mealy'ego	147
5.6. Implementacja automatów skończonych na FPGA zgodnie z ich opisem za pomocą ASM	148
5.6.1. Implementacja automatu Moore'a.....	149
5.6.2. Implementacja automatu Mealy'ego.....	150
5.7. Opis za pomocą ASM wspólnych modeli automatów Mealy'ego i Moore'a	152
5.8. Opis procesów równoległych za pomocą ASM	155
5.9. Opis układów kombinacyjnych za pomocą ASM	157
5.10. Wnioski.....	160
6. Automaty skończone ze ścieżką przetwarzania danych (FSMD).....	163
6.1. Sieć działań ASMD i automaty FSMD.....	163
6.2. Język ASMD	164
6.3. Wykorzystanie FSMD Moore'a do implementacji algorytmu mnożenia a.....	165
6.4. Wykorzystanie FSMD Mealy'ego do implementacji algorytmu mnożenia a.....	168
6.5. Zwiększenie wydajności synchronicznego układu mnożącego	173
6.6. Porównanie różnych sposobów implementacji synchronicznego układu mnożącego	176

6.7. Wnioski	182
Część III. Projektowanie procesorów wbudowanych	185
7. Podstawy projektowania procesorów wbudowanych.....	186
7.1. Architektury zestawów instrukcji procesora.....	186
7.2. Architektura zestawu instrukcji realizowanego procesora.....	188
7.2.1. Architektura zestawu instrukcji mikrokontrolera PIC16F84A	188
7.2.2. Projektowanie zestawu instrukcji procesora PIC.....	191
7.3. Podstawowe struktury procesorów RISC	194
7.3.1. Jednocyklowy procesor PIC	195
7.3.2. Wielocyklowy procesor PIC.....	196
7.3.3. Potokowy procesor PIC	198
7.3.4. Wykorzystanie pamięci podręcznej	201
7.3.5. Procesory superskalarne.....	202
7.3.6. Procesory wielowątkowe.....	203
7.3.7. Procesory wielordzeniowe.....	203
7.4. Metodyka projektowania procesorów na FPGA.....	204
7.5. Wnioski	206
8. Projektowanie jednocyklowego procesora PIC	209
8.1. Projektowanie układu operacyjnego	209
8.1.1. Elementy pamięci procesora	209
8.1.2. Określenie adresu następnej instrukcji.....	211
8.1.3. Realizacja instrukcji transferu danych (instrukcji movlw, movwf i movf).....	212
8.1.4. Realizacja operacji na bajtach (instrukcji formatu a)	213
8.1.5. Cykliczne operacje przesunięcia (instrukcje rlf i rrf)	213
8.1.6. Realizacja operacji inkrementacji i dekrementacji z pominięciem następnej instrukcji, jeśli wynik jest zerowy (instrukcje incfsz i decfsz)	215
8.1.7. Realizacja operacji bitowych (instrukcji formatu b)	216
8.1.8. Realizacja operacji na stałych (instrukcji formatu c).....	217
8.1.9. Realizacja instrukcji skoku do adresu (goto), wywołania podprogramu (call) i powrotu z podprogramu (return, retlw).....	217
8.1.10. Realizacja ładowania wartości akumulatora z pamięci danych (instrukcja lw) i zapisu wartości akumulatora w pamięci danych (instrukcja sw)	219
8.1.11. Realizacja skoku do adresu w zależności od wyniku poprzedniej operacji (instrukcje gotoz i gotonz).....	219
8.2. Projektowanie układu arytmetyczno-logicznego	222

8.3. Projektowanie układu sterowania.....	224
8.4. Realizacja procesora jednocyklowego PIC.....	227
8.4.1. Struktura realizowanego procesora.....	227
8.4.2. Opis komponentów układu operacyjnego w języku Verilog	228
8.4.3. Opis układu operacyjnego (datapath)	231
8.4.4. Opis procesora PIC	234
8.4.5. Opis pamięci danych.....	235
8.4.6. Opis pamięci instrukcji.....	236
8.4.7. Opis modułu najwyższego poziomu	237
8.5. Debugowanie procesora jednocyklowego PIC	238
8.5.1. Debugowanie komponentów układu operacyjnego.....	239
8.5.2. Debugowanie układu sterowania.....	239
8.5.3. Debugowanie pamięci danych i pamięci instrukcji	241
8.5.4. Debugowanie całego projektu procesora PIC.....	241
8.6. Ocena wydajności procesora jednocyklowego	246
8.7. Modyfikacja procesora PIC	250
8.8. Wnioski	250
9. Projektowanie procesorów wielocyklowych PIC	253
9.1. Tradycyjne projektowanie procesora wielocyklowego PIC	253
9.1.1. Projektowanie układu operacyjnego.....	253
9.1.2. Projektowanie układu sterującego.....	256
9.1.3. Opis komponentów procesora wielocyklowego w języku Verilog.....	263
9.1.4. Debugowanie procesora wielocyklowego.....	270
9.1.5. Analiza projektu procesora wielocyklowego.....	270
9.2. Projektowanie procesora dwucyklowego PIC	271
9.2.1. Projektowanie układu procesora dwucyklowego	271
9.2.2. Opis w języku Verilog komponentów procesora dwucyklowego PIC.....	273
9.2.3. Debugowanie procesora dwucyklowego	276
9.3. Projektowanie czterocyklowego procesora PIC	276
9.3.1. Projektowanie układu operacyjnego.....	276
9.3.2. Projektowanie układu sterującego.....	281
9.3.3. Opis w języku Verilog komponentów procesora czterocyklowego....	282
9.3.4. Debugowanie procesora czterocyklowego	283
9.4. Ocena kosztów realizacji i wydajności procesorów PIC.....	284
9.5. Wnioski	286

Część IV. Projektowanie bloków cyfrowego przetwarzania sygnałów	289
10. Projektowanie filtrów cyfrowych	291
10.1. Systemy cyfrowego przetwarzania sygnałów na FPGA	291
10.1.1. Wprowadzenie do cyfrowego przetwarzania sygnałów	291
10.1.2. Zastosowanie FPGA do realizacji systemów DSP	292
10.2. Filtry cyfrowe	294
10.2.1. Klasyfikacja filtrów cyfrowych według częstotliwości przepustowej	295
10.2.2. Filtry FIR i IIR	296
10.2.3. Wybór między filtrami IIR i FIR	297
10.2.4. Typy filtrów FIR	298
10.3. Opis filtrów cyfrowych w języku Verilog	298
10.4. Metodyka projektowania filtrów cyfrowych na FPGA	302
10.4.1. Definiowanie specyfikacji filtrów	303
10.4.2. Obliczanie współczynników filtra	304
10.4.3. Definiowanie struktury filtra	305
10.4.4. Analiza wpływu skończonej długości słowa	305
10.4.5. Realizacja filtra	306
10.5. Projektowanie filtrów cyfrowych w systemie MATLAB	307
10.6. Symulacja filtrów cyfrowych w systemie ModelSim	316
10.7. Realizacja filtrów cyfrowych w systemie Quartus	318
10.7.1. Realizacja w systemie Quartus filtrów cyfrowych opracowanych w systemie MATLAB	318
10.7.2. Badanie realizacji na FPGA różnych typów filtrów cyfrowych	319
10.8. Wnioski	322
Część V. Projektowanie podsystemu synchronizacji	325
11. Realizacja projektów synchronicznych	326
11.1. Zasady funkcjonowania projektów synchronicznych	326
11.2. Zalecenia firmy Intel dotyczące projektowania niezawodnych projektów synchronicznych	327
11.2.1. Unikanie pętli kombinacyjnych	327
11.2.2. Unikanie nieumyślnego wprowadzania zatrzaśków do projektu	328
11.2.3. Unikanie opóźnień w ścieżkach synchronizacji	329
11.2.4. Używanie układów synchronicznych do wygenerowania pojedynczego impulsu	329
11.2.5. Generacja zegara wewnętrznego	330
11.2.6. Wykorzystanie wewnętrznej logiki FPGA jako źródła sygnału zegarowego lub sygnałów sterujących rejestrami	331

11.2.7. Domeny synchronizacji	331
11.2.8. Dzielniki częstotliwości	332
11.2.9. Korzystanie z liczników kaskadowych (ripple counters)	332
11.2.10. Zastosowanie sieci synchronizacji FPGA.....	334
11.3. Multipleksowanie sygnałów zegarowych.....	335
11.4. Bramkowanie sygnałów zegarowych	338
11.4.1. Bramkowanie sygnałów zegarowych w projektach na FPGA.....	338
11.4.2. Metody bramkowania sygnałów zegarowych	338
11.4.3. Realizacja rejestrów z bramkowaniem sygnału zegarowego.....	343
11.4.4. Badanie metod bramkowania sygnałów zegarowych	348
11.4.5. Zastosowanie opcji Auto Gated Clock Conversion	349
11.4.6. Bramkowanie sygnałem asynchronicznym.....	350
11.5. Zalecenia dotyczące tworzenia szybkich i złożonych projektów.....	351
11.5.1. Planowanie realizacji fizycznej	351
11.5.2. Planowanie zasobów FPGA	352
11.5.3. Spełnienie ograniczeń czasowych	353
11.6. Optymalizacja podsystemu synchronizacji.....	353
11.6.1. Retiming	353
11.6.2. Duplikacja rejestrów	354
11.6.3. Synteza fizyczna logiki kombinacyjnej	354
11.6.4. Opcje systemu Quartus wpływające na optymalizację synchronizacji	355
11.6.5. Optymalizacja ścieżek krytycznych	358
11.7. Projektowanie sygnałów resetowania	359
11.7.1. Synchroniczne resetowanie rejestrów	359
11.7.2. Asynchroniczne resetowanie rejestrów	363
11.7.3. Badania metod resetowania rejestrów	368
11.8. Atrybuty syntezy wpływające na synchronizację projektu	369
11.9. Wnioski	371
12. Bloki pętli synchronizacji fazowej PLL	374
12.1. Zasady funkcjonowania bloku PLL.....	374
12.2. Specyfika architektury bloków PLL w rodzinie FPGA Cyclone V	376
12.3. Cechy funkcjonowania bloków PLL w układach FPGA z rodziny Cyclone V	378
12.4. Tryby synchronizacji sprzężenia zwrotnego	379
12.4.1. Tryb synchronizacji źródłowej.....	379
12.4.2. Tryb kompensacji LVDS.....	380

12.4.3. Tryb bezpośredniej kompensacji.....	381
12.4.4. Tryb kompensacji normalnej.....	381
12.4.5. Tryb bufora z zerowym opóźnieniem ZDB	382
12.4.6. Tryb zewnętrznego sprzężenia zwrotnego	382
12.5. Ułamkowy i całkowity tryb pracy bloków PLL	383
12.6. Przełączanie zegarów referencyjnych.....	383
12.7. Konfigurowanie bloków PLL za pomocą blików IP i Edytora parametrów systemu Quartus	384
12.8. Rekonfiguracja bloków PLL	390
12.9. Rekonfiguracja PLL za pomocą interfejsu Avalon-MM.....	392
12.10. Rekonfiguracja bloków PLL za pomocą strumieniowego przesyłania pliku MIF ze wbudowanego bloku pamięci FPGA	393
12.11. Dynamiczne przesunięcie fazowe bloku PLL	394
12.11.1. Wykonywanie dynamicznego przesunięcia fazowego przy użyciu Bloku IP do rekonfiguracji PLL (Altera PLL Reconfig).....	394
12.11.2. Wykonywanie dynamicznego przesunięcia fazowego przy użyciu Bloku IP Altera PLL.....	394
12.12. Wnioski.....	397
Zakończenie	401
Literatura	403
Indeks tematyczny.....	405
Spis tabel.....	417
Spis rysunków	419
Streszczenie	429
Abstract.....	431