

Spis treści

Wstęp.....	9
1. Pierwszy mikroprocesor w układzie FPGA	19
1.1. Po co nam procesor w układzie FPGA?.....	20
1.2. Poznajemy NIOS II.....	21
1.3. Nasz pierwszy projekt – minimalny działający system	22
1.3.1. ...pierwsze kroki (kliknięcia) w <i>Platform Designer</i>	23
1.3.2. Z czego zbudujemy nasz procesor?.....	25
1.3.3. Czas urzeczywistnić nasz procesor i wgrać go do układu FPGA.....	37
1.4. Piszemy pierwszy program	40
1.4.1. Tworzymy projekt i generujemy BSP	40
1.4.2. Zaczynamy debugowanie	41
1.5. Podsumowanie	43
2. Wyjście na świat, czyli rzecz o GPIO	45
2.1. Czas na GPIO.....	46
2.2. Kolej na miganie – czyli piszemy pierwszy program	50
2.2.1. Jeszcze wejście.....	53
2.3. Programie, nie uciekaj – czyli jak sprawić, aby nasz program pozostał w układzie na dłużej.....	54
2.4. Zadania do wykonania	56
3. Timery i przerwania – część 1	57
3.1. Uzupełniamy system.....	58
3.2. Timer i przerwania	60
3.2.1. Timer	60
3.2.2. Przerwania	62
3.2.3. Konfiguracja przerwania.....	64
3.3. Wracamy do programowania	64
3.3.1. Modyfikujemy nasz program	66
3.3.2. Nieco modyfikacji	68
3.4. Podsumowanie	69
3.5. Podpowiedzi dla początkujących	70

4. Timery i przerwania – część 2	71
4.1. Przerwania z zewnątrz	72
4.1.1. Na początek trochę grzebania w sprzęcie	72
4.1.2. ...a potem programujemy!	73
4.2. Drgania styków – jak sobie z nimi poradzić	75
4.2.1. Rozwiązanie „na sterydach”	76
4.3. Ważne informacje o timerach i przerwaniach	77
4.3.1. Czas mierzymy precyzyjnie...	77
4.3.2. Jak wyłączyć przerwania (czyli o dostępie atomowym raz jeszcze)	78
4.4. Podsumowanie	80
5. UART	81
5.1. Krótkie wprowadzenie	82
5.2. Czas dodać moduł UART	83
5.3. Czas porozmawiać z komputerem przez koła	85
5.3.1. Odbieramy dane	87
5.3.2. Przerwania i bufor kołowy	88
5.3.3. Zmieniamy prędkość	92
5.3.4. Użyteczny przykład	93
5.4. Podsumowanie	93
6. Interfejs SPI	95
6.1. Krótkie wprowadzenie	96
6.1.1. Interfejs SPI	96
6.1.2. Przykładowe układy	98
6.2. Dodajemy moduł SPI do naszego systemu	100
6.3. Czas na poważną rozmowę „w języku SPI”	103
6.3.1. MCP23S08	104
6.3.2. BMP280	108
6.4. Podsumowanie	111
7. Magistrala I²C	113
7.1. Krótkie wprowadzenie	114
7.1.1. Interfejs I ² C	114
7.1.2. Przykładowe układy	117
7.2. Dodajemy moduł I ² C do naszego systemu	119
7.2.1. Tworzymy moduł bufora dla magistrali I ² C	120

7.3.	Czas na wytęsknione programowanie.....	125
7.3.1.	Modyfikujemy stare programy!.....	126
7.4.	Podsumowanie i uwagi	128
7.4.1.	Drobna uwaga na temat transmisji szeregowej po JATG	128
8.	Przetwornik analogowo-cyfrowy	129
8.1.	Krótkie wprowadzenie	130
8.1.1.	Przetwornik ADC	130
8.1.2.	Czujnik temperatury STLM20	131
8.1.3.	Wyświetlacz HD44780.....	131
8.1.4.	Klawiatura rezystancyjna, czyli jak podłączyć wiele przycisków do jednego wyprowadzenia procesora.....	133
8.2.	Czas wyruszyć na pomiary.....	134
8.2.1.	Dodajemy ADC.....	134
8.2.2.	Programujemy woltomierz FPGA	139
8.2.3.	Analogowy pomiar temperatury	141
8.2.4.	Analogowe przyciski	142
8.3.	Wyświetlacz HD44780.....	144
8.3.1.	Uzupełniamy sprzęt	145
8.3.2.	Biblioteka dla wyświetlacza.....	145
8.4.	Podsumowanie	147
9.	Własne moduły <i>Avalon-MM</i> – generator PWM.....	149
9.1.	Magistrala <i>Avalon-MM</i>	150
9.2.	Implementacja modułu PWM	153
9.2.1.	Przygotowania.....	153
9.2.2.	Jak działa PWM?.....	154
9.2.3.	Ruszamy do pracy – implementacja.....	155
9.2.4.	Ubieramy nasz opis w VHDL w „błoczek” <i>Platform Designer</i>	160
9.3.	Programujemy obsługę własnego modułu	162
9.4.	A może konfiguracja do tego?.....	164
9.5.	Podsumowanie	166
10.	Własne moduły <i>Avalon-MM</i> – enkoder i wyświetlacz 7-segmentowy.....	167
10.1.	Wstęp w skrócie telegraficznym	168
10.1.1.	Enkodery inkrementalne.....	168
10.2.	Sterownik do wyświetlacza raz poproszę!	169
10.2.1.	Oprogramowanie... ..	171

10.3.	Czas zdekodować enkoder	172
10.3.1.	Eliminacja drgań styków	172
10.3.2.	Enkoder we własnej osobie.....	173
10.3.3.	Parę linijek programu do sukcesu!.....	177
10.4.	Podsumowanie	178
11.	Własne moduły <i>Avalon-MM</i> – diody WS2812, czyli czas na kolor.....	179
11.1.	Wstęp w kolorze.....	180
11.2.	Czas znów pomagać diodami	181
11.2.1.	Zużycie zasobów – czas na RAM.....	185
11.2.2.	Projektujemy i używamy modułu z portem <i>Avalon Master</i>	185
11.3.	Przerwania od podszewki.....	190
11.4.	Podsumowanie	192
12.	Więcej pamięci, czyli karty SD.....	193
12.1.	Karty pamięci, systemy plików – jak się nie pogubić.....	194
12.1.1.	Przygotowania.....	195
12.2.	Przygotowanie systemu.....	195
12.3.	Oprogramowanie.....	197
11.4.	Współdzielenie modułów	202
11.4.1.	Jeszcze więcej porządku.....	203
11.5.	Podsumowanie	203
13.	VGA, czyli czas na duże wyświetlacze.....	205
13.1.	Od czego zacząć?	206
13.1.1.	Złącze VGA	206
13.1.2.	Sposób przesyłania danych	207
13.2.	Generator znaków – czyli co zbudujemy	209
13.2.1.	Generator sygnałów sterujących.....	209
13.2.2.	Dobry plan to podstawa!	210
13.2.3.	Czas wcielić w życie nasz plan.....	212
13.2.4.	...jeszcze czcionka!	216
13.2.5.	Szczypta oprogramowania – wersja demo	217
13.3.	Podsumowanie	218
14.	HDMI, czyli duży „cyfrowy” wyświetlacz.....	219
14.1.	Od czego zacząć?	220
14.1.1.	Złącze HDMI	220
14.1.2.	Transmisja różnicowa – o co chodzi?.....	221

14.1.3. Transmisja TMDS	222
14.1.4. Przypomnijmy sobie naszą kartę graficzną VGA	223
14.2. Implementacja HDMI	223
14.2.1. Zaczniemy od założeń	223
14.2.2. Powtórka	224
14.2.3. Ostatnie połączenia	228
14.2.4. Oprogramowania linijek kilka	229
14.3. Podsumowanie	229
15. Parę drobiazgów, które nigdzie nie pasują, czyli... uzupełnienie.....	231
15.1. Symulacja mikroprocesora NIOS II	232
15.2. Symulacja trochę prościej	235
15.3. A może NIOS będzie tylko komponentem?	238
15.4. A jeśli jednak trzeba sprawdzić coś na żywo?	239
15.5. Podsumowanie... całej przygody	242
16. O VHDL słów kilka	243
16.1. Część teoretyczna – informacje na temat VHDL	245
16.1.1. Wprowadzenie	245
16.1.1.1. Etymologia	245
16.1.1.2. Definicja	247
16.1.1.3. Przeznaczenie	249
16.1.1.4. Istota VHDL	251
16.1.2. Wybrane kwestie VHDL	253
16.1.2.1. Dlaczego VHDL nie jest językiem?	253
16.1.2.2. Dlaczego VHDL nie jest językiem programowania?	254
16.1.2.3. Znaczenie słowa „język” w nazwie VHDL	255
16.1.2.4. HDL a FHL	255
16.1.3. Informacje poboczne	256
16.2. Część praktyczna – projektowanie na bazie VHDL	259
16.2.1. Podstawy kodu VHDL	259
16.2.1.1. Główne cechy kodu	259
16.2.1.2. Typy danych	260
16.2.1.3. Rodzaje danych	264
16.2.1.4. Komentarze	265
16.2.2. Układy cyfrowe w ujęciu kodu VHDL	266
16.2.2.1. Blok <i>entity</i> (czyli wyprowadzenia i parametry układów cyfrowych)	266
16.2.2.2. Typy wyprowadzeń układów cyfrowych	267
16.2.2.3. Blok <i>architecture</i> (czyli zachowanie układów cyfrowych)	268

16.2.3. Konstrukcje kodu VHDL.....	268
16.2.3.1. Wprowadzenie	268
16.2.3.2. Wybrane konstrukcje współbieżne	269
16.2.3.3. Wybrane konstrukcje sekwencyjne.....	275
16.2.4. Inne stosowane konstrukcje kodu VHDL	279
16.2.4.1. Komponenty.....	279
16.2.4.2. Wykrywanie zboczy sygnału zegarowego	280
16.2.4.3. Funkcje konwersji między typami.....	281
16.2.4.4. Funkcje logiczne	283
16.2.4.5. Operatory	283
16.2.5. Przykłady układów cyfrowych opisanych z użyciem VHDL	285
16.2.5.1. Istotna rzecz na początek kodu.....	285
16.2.5.2. Układy kombinacyjne	285
16.2.5.3. Układy sekwencyjne.....	287
16.2.5.4. Style opisu w praktyce	288
16.3. Podsumowanie	290