**Listing 1. Ustawienie czasów do generowania przebiegów**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 0**,** **(**NIOS2\_CPU\_FREQ **\*** 350LL**)** **/** 1000000000LL**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 4**,** NIOS2\_CPU\_FREQ **\*** 900LL **/** 1000000000LL**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 8**,** NIOS2\_CPU\_FREQ **\*** 900LL **/** 1000000000LL**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 12**,** NIOS2\_CPU\_FREQ **\*** 350LL **/** 1000000000LL**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 16**,** NIOS2\_CPU\_FREQ **\*** 60000LL **/** 1000000000LL**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 20**,** 0x00FF00**);**

IOWR\_32DIRECT**(**WS2812\_0\_BASE**,** 24**,** 0x0000FF**);**

**Listing 2. Dodanie sygnałów magistrali Avalon Master**

--avalon memory-mapped master

m\_address **:** **out** std\_logic\_vector**(**16 **downto** 0**);**

m\_byteenable **:** **out** std\_logic\_vector**(**3 **downto** 0**);**

m\_read **:** **out** std\_logic**;**

m\_readdata **:** **in** std\_logic\_vector**(**31 **downto** 0**);**

m\_write **:** **out** std\_logic**;**

m\_writedata **:** **out** std\_logic\_vector**(**31 **downto** 0**);**

**Listing 3. Zapis/odczyt danych do/z pamięci**

-- będziemy zawsze prowadzić odczyt całego słowa 32-bitowego

-- z pamięci - ustawiamy linie

m\_write **<=** '0'**;**

m\_writedata **<=** **(others** **=>** '0'**);**

m\_read **<=** '1'**;**

m\_byteenable **<=** "1111"**;**

-- nasze dane będą zawsze ostatnio pobranymi danymi z pamięci,

-- z pominięciem najstarszych 8 bitów (aby ułatwić adresowanie)

data **<=** m\_readdata**(**23 **downto** 0**);**

**Listing 4. Modyfikacja programu - użycie funkcji zapisu WS2812**

volatile uint8\_t WS2812Done **=** 0**;**

void WS2812Interrupt**(**void**\*** context**)**

**{**

//kasowanie flagi przerwania

IOWR\_32DIRECT**(**WS2812\_INT\_0\_BASE**,** WS2812\_STATUS\_REG**,** 0**);**

WS2812Done **=** 1**;**

**}**

void WS2812UpdateWaitInt**(**void**)**

**{**

IOWR\_32DIRECT**(**WS2812\_INT\_0\_BASE**,** WS2812\_CONFIG\_REG**,** WS2812\_CONFIG\_INT **|** WS2812\_CONFIG\_START**);**

**while(**WS2812Done **!=** 1**);**

WS2812Done **=** 0**;**

ALT\_USLEEP**(**5000**);**

**}**

**Listing 5. Użycie rejestru przerwań (bez przerwań)**

void WS2812UpdateWaitPoll**(**void**)**

**{**

IOWR\_32DIRECT**(**WS2812\_INT\_0\_BASE**,** WS2812\_CONFIG\_REG**,** WS2812\_CONFIG\_START**);**

**while(**IORD\_32DIRECT**(**WS2812\_INT\_0\_BASE**,** WS2812\_STATUS\_REG**)** **!=** 1**);**

IOWR\_32DIRECT**(**WS2812\_INT\_0\_BASE**,** WS2812\_STATUS\_REG**,** 0**);**

ALT\_USLEEP**(**5000**);**

**}**