

W artykule przedstawiono opis woltomierza zaimplementowanego w FPGA, realizującego ciągłe, automatyczne pomiary stałego napięcia elektrycznego (DC). Wyniki pomiarów są prezentowane na czteropozycyjnym, multipleksowanym wyświetlaczu siedmio-segmentowym, a także na każdym urządzeniu wyświetlającym, zgodnym ze standardem graficznym VGA.

# Implementacja cyfrowego woltomierza z wyświetlaniem VGA w FPGA

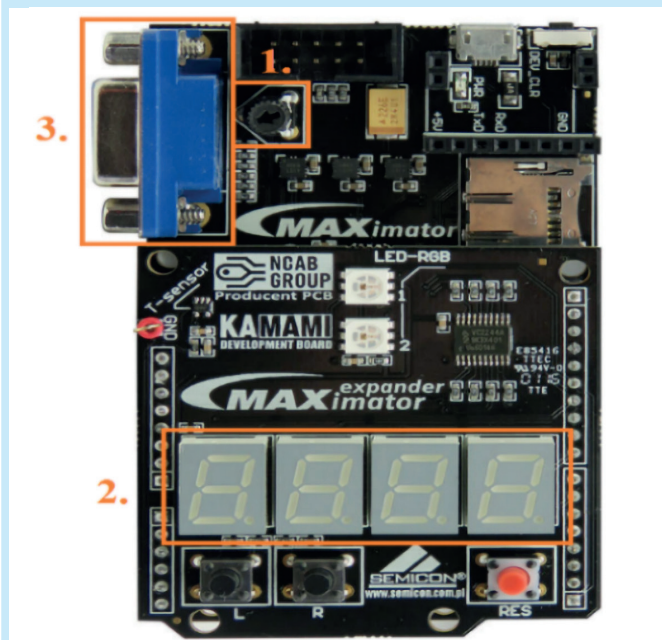
Opracowany woltomierz opiera się w swoim działaniu na dwunastobitowym przetworniku analogowo-cyfrowym (ADC), o częstotliwości taktowania równej 10 MHz oraz częstotliwości próbkowania równej 1 MHz (inaczej mówiąc: zwraca on milion próbek w ciągu jednej sekundy, funkcjonując z przepływnością 1 Msps (Megasamples per second)). Przetwornik ten wbudowany jest w układ FPGA obecny w zestawie maXimator (jest to układ z rodziny MAX10 firmy Intel (dawniej Altera), o oznaczeniu kodowym: 10M08DAF256C8GES) i przeznaczony jest do pracy z sygnałami o amplitudach z zakresu od 0 do 2,5 V (powyżej tego zakresu zwraca on, dzięki zastosowanej ochronie nadnapięciowej wyniki odpowiadające granicznemu, maksymalnemu napięciu tj. 2,5 V).

Jak można zauważyć, mamy do czynienia z urządzeniem, które oblicza wartości mierzonego napięcia (wyrażone w voltach), w oparciu o rezultaty zwracane przez wyżej wymieniony przetwornik (rezultaty te to poziomy kwantyzacji, będące całkowitymi wartościami liczbowymi, jednoznacznie powiązаны z określonymi przedziałami

wartości mierzonego napięcia). Co więcej pobiera ono je w obrębie tego samego układu, w którym ulokowany jest przetwornik, wobec czego jego funkcjonalność jest w całości zawarta w tym układzie (nie jest potrzebny tym samym odrębny, dodatkowy układ scalony). Oprócz tego zakres jego pracy obejmuje wartości mierzonego napięcia pokrywające się z przytoczonymi wcześniej poziomami amplitud. Samo napięcie natomiast do niego doprowadzane regulowane jest z użyciem dedykowanego potencjometru, przedstawionego na **rysunku 1**, na którym ukazane są również dwa pozostałe elementy woltomierza. Elementy te to:

- czteropozycyjny, multipleksowany wyświetlacz siedmio-segmentowy (zawarty na płycie rozszerzającej zestaw maXimator tj. maXimator Expander),
- gniazdo standardu graficznego VGA (zawarte w samym zestawie), umożliwiające dołączenie dowolnego urządzenia wyświetlającego z nim zgodnego.

Zatem jest to także urządzenie bazujące na niewielkiej liczbie periferiów, nieskomplikowane pod względem budowy fizycznej (biorąc



Rysunek 1. Zestaw maXimator (z zainstalowaną płytką rozszerzającą maXimator Expander). 1. Potencjometr przeznaczony do regulacji mierzonego napięcia; 2. Czterocyfrowy, multipleksowany wyświetlacz siedmiosegmentowy; 3. Gniazdo standardu graficznego VGA

szczególnie pod uwagę brak konieczności uprzedniego opracowania i zastosowania własnych torów wejściowych dla obecnego w zestawie źródła. Do stworzenia oraz implementacji woltomierza użyto darmowego środowiska projektowego o nazwie *Quartus II*, w wersji *16.1 Lite* (do pobrania ze strony <https://goo.gl/qP593Y>), która wbrew pozorom nie jest jedynie w stosunku do niego obowiązującą (może on bowiem zostać również uruchomiony na pozostałych wersjach wspomnianego środowiska, tj. *Standard* i *Prime*, zarówno o niższych (starszych) jak i wyższych (nowszych) numerach, pod warunkiem jednak, że obsługują one wspomniany wcześniej układ *FPGA*).

## Budowa woltomierza

Na **rysunku 2** przedstawiono ogólną (poglądową) budowę omawianego woltomierza. Na jego podstawie można wyróżnić łącznie pięć grup modułów go budujących. Są to:

- moduły generacji i podziałów sygnałów zegarowych, stosowanych w obrębie woltomierza (I),
- moduły obsługi przetwornika ADC (II),
- moduły wyznaczania wyników mierzonego napięcia (III),
- moduły przedstawiania wyników w oparciu o standard graficzny *VGA* (IV),
- moduły przedstawiania wyników na wyświetlaczu siedmio-segmentowym (V).

Podstawową ich cechą jest w pełni synchroniczna praca, tj. taka, która opiera się o zbocza sygnałów zegarowych do nich doprowadzonych (innymi słowy: funkcjonalność modułów realizowana jest

wraz z każdym kolejnym taktem wspomnianych sygnałów, przy czym momentem ją wyzwalającym są zbocza tych taktów, domyślnie narastające (przejścia od ich stanów niskich do wysokich). Tym samym również i woltomierz jest w pełni synchroniczny, co przekłada się bezpośrednio na jego architekturę, która jest potokowa (ang. *pipeline*). Jej istotą jest fakt, że moduły będące w stosunku do siebie następującymi po sobie stopniami pojedynczego

potoku (którym jest woltomierz) wykonują w tym samym czasie własne zadania, bazując na danych pochodzących z modułów wcześniej występujących. Czynią to w sposób przedstawiony na **rysunku 3**.

Jak można zauważyć dane pobrane z przetwornika *ADC* są przekazywane do modułów obsługujących ten przetwornik (grupa II), po czym do modułów wyznaczających wyniki mierzonego napięcia (grupa III). Z kolei te wyniki są przekazywane do modułów odpowiedzialnych za ich prezentację, zarówno w oparciu o standard *VGA* (grupa IV), jak i w oparciu o 4-pozycyjny, multipleksowany wyświetlacz 7-osegmentowy (grupa V). Działania te powtarzane są dla każdego nowego danych, przy czym są one pobierane niezależnie od tych, które już podlegają przetwarzaniu (np.: w trakcie trzeciego taktu sygnału zegarowego dane oznaczone kolorem czerwonym przekazywane są do grup IV i V, podczas gdy dane oznaczone kolorem zielonym przekazywane są do grupy III. W tym samym czasie do grupy II są wprowadzane nowe dane, oznaczone kolorem niebieskim). Wszystko to realizowane jest na zboczach narastających sygnałów pochodzących z modułów grupy I. (nie należą one do wspomnianego wcześniej potoku, ze względu na to, że nie są one związane z przetwarzaniem danych). W dalszej części omówiono wszystkie wymienione wcześniej grupy modułów.

## Moduły generacji i podziału sygnału zegarowego

W skład tej grupy wchodzi:

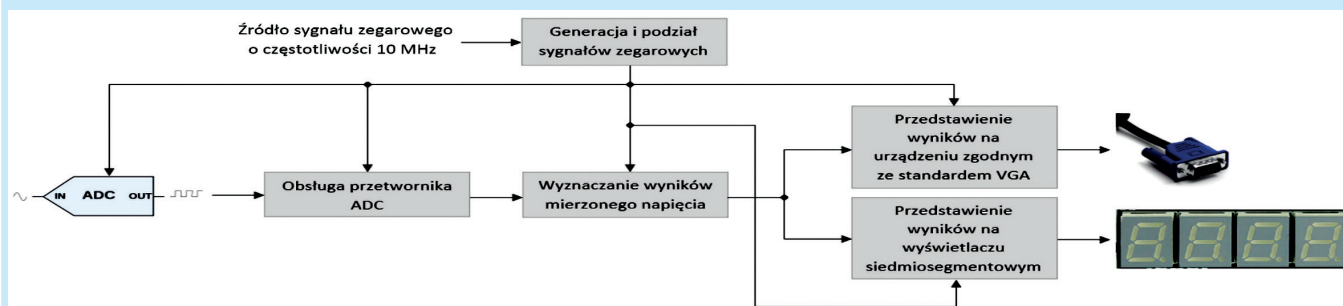
- pojedyncza pętla *PLL*,
- moduł *Generator\_opoznien\_impulsow*.

Pierwszy z przedstawionych elementów odpowiedzialny jest za generację sygnałów zegarowych, które w zależności od potrzeb albo są dostarczane bezpośrednio do dalszych modułów woltomierza, albo pośrednio, po uprzednim podzieleniu z zastosowaniem drugiego z nich. Sygnały te przedstawiono w **tabeli 1**.

Na jej podstawie można stwierdzić, że są to sygnały zegarowe o częstotliwościach 10 MHz i 50 MHz, tworzone w oparciu o wejściowy

Numer taktu sygnału zegarowego	Numer grupy modułów		
	II	III	IV i V
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Rysunek 3. Istota architektury potokowej omawianego woltomierza (oznaczenia grup modułów takie jak w tekście)



Rysunek 2. Ogólna (poglądowa) budowa woltomierza



sygnał zegarowy o częstotliwości 10 MHz (pochodzący ze źródła zawartego w zestawie *maXimator*). Obecność ich dwóch nie jest przydatkowa. Pierwszy z nich bowiem stosowany jest w odniesieniu do modułów obsługi przetwornika *ADC* (wymaga on do działania sygnału zegarowego pochodzącego z pętli *PLL*) i modułów wyznaczania wyników mierzonego napięcia. Drugi natomiast stosowany jest w przypadku modułów związanych z przedstawianiem wyników w oparciu o standard graficzny *VGA* (jedynym wyjątkiem jest wyświetlacz siedmiosegmentowy, do którego z kolei doprowadzony jest wspomniany wcześniej wejściowy sygnał zegarowy pętli). Oprócz nich generowany jest także dodatkowy sygnał oznajmiający o gotowości pętli *PLL* do działania (w sytuacji, gdy tworzone przez nią sygnały zegarowe są stabilne i nie podlegają zanikom). Sygnałem tym posługuje się przetwornik, w celu zapewnienia sobie niezawodnej pracy (na jego podstawie albo przerywa swoje działanie albo je kontynuuje lub wznowia, jeśli doszło do wspomnianych zaników). Standardowo obecny jest tu również sygnał resetu, domyślnie ustawiony na stan niski, w celu zapewnienia ciągłej pracy pętli.

Drugi element to nic innego jak generator impulsów o stałej szerokości, równej pojedynczemu taktowi wejściowego sygnału zegarowego. Impulsy te tworzone są co określony takt tegoż sygnału. Tym samym również co określony takt generowane są ich zbocza narastające, co z kolei przekłada się na obecne w nazwie modułu opóźnienie (im szybciej się one pojawiają, tym więcej czynności w określonej jednostce czasu jest w stanie wykonać dany moduł, im rzadziej natomiast, tym mniej). Jest to podejście równoważne dzielnikowi częstotliwości (preskalowaniu), przy czym różnica w stosunku do niego jest brak zapewnienia wypełnienia taktów wyjściowego sygnału zegarowego na poziomie 50% (nie jest to jednak tu istotne – w stosunku do opóźnienia bowiem liczą się jedynie wspomniane zbocza, a nie poziomy i czas trwania taktów sygnału zegarowego. Tym samym możliwe staje się zaniedbanie tego parametru i zastosowanie w sposób wystarczający jednostaktowych impulsów). Bez względu jednak na to, w stosunku do pętli *PLL* generator opóźnień impulsów jest modulem parametryzowanym tzn. takim, w którym w zależności od potrzeb można ustalić wartości wybranych jego wielkości (zwanymi parametrami). Wielkości te (wraz z sygnałami omawianego modułu) przedstawiono w **tabeli 2**.

Jak można zauważyć jest tu obecny tylko jeden parametr, odpowiedzialny za określenie momentów wystąpienia generowanych impulsów wyjściowych. Wbrew nazwie nie jest to wartość liczbowa brana podwójnie (zależność ta znajduje pokrycie wyłącznie w stosunku do modułów przedstawiania wyników w oparciu o standard graficzny *VGA*, które wymagają do działania m.in. sygnału zegarowego o dwukrotnie niższej częstotliwości (o czym w dalszej części) – w pozostałych przypadkach obowiązuje funkcja wspomniana w tabeli 2). Wyprowadzenia natomiast związane są ściśle z sygnałami zegarowymi (obecny tu sygnał resetu ustawiony jest z kolei na stan niski w celu zapewnienia nieprzerwanej pracy modułów) Łącznie w ramach woltomierza występują trzy generatory opóźnień impulsów.

**Tabela 1. Wyprowadzenia pętli PLL**

Port	Funkcja
clk_in_10_mhz_clk	Wejściowy sygnał zegarowy o częstotliwości 10 MHz
reset_reset	Reset pętli PLL
clk_out_10_mhz_clk	Wyjściowy sygnał zegarowy o częstotliwości 10 MHz
clk_out_50_mhz_clk	Wyjściowy sygnał zegarowy o częstotliwości 50 MHz
locked_10_mhz_export	Sygnał gotowości pętli do działania (szczególnie w odniesieniu do wyjściowego sygnału zegarowego o częstotliwości 10 MHz)

**Tabela 2. Parametry i wyprowadzenia generatora opóźnienia impulsów**

Parametry	Funkcja
Stopień podziału razy dwa	Wartość powiększona o jeden, określająca, co który impuls wejściowego sygnału zegarowego ma występować impuls jednotaktowy na wyjściu generatora
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset generatora opóźnienia impulsów
Wyjscie zegarowe	Wyjściowy sygnał zegarowy będący ciągiem jednotaktowych impulsów występujących co określony takt wejściowego sygnału zegarowego

## Domeny zegarowe woltomierza, czyli wnioski wynikające z modułów generowania i podziału sygnału zegarowego

Dotychczas przedstawiony opis pozwala na zauważenie ważnego aspektu woltomierza. Mianowicie, składają się na niego dwie odrębne grupy modułów, pracujących na sygnałach zegarowych o dwóch różnych częstotliwościach (dla przypomnienia: chodzi o częstotliwość równą 10 i 50 MHz). Grupy, o których mowa to domeny zegarowe (ang. *clock domains*). W ogólności dąży się do uzyskania jednej z nich w obrębie całego urządzenia. Takie, a nie inne podejście wynika z faktu, że przy sygnałach zegarowych o różnych częstotliwościach zawsze zachodzi sytuacja, w której zbocza narastające jednego sygnału będą nieznacznie wyprzedzały zbocza narastające drugiego sygnału, przyczyniając się w ten sposób do pojawienia się niepożądanych, metastabilnych stanów przerzutników (są to stany, w trakcie których przerzutniki zachowują się nieprzewidywalnie, co przekłada się na całkowicie losowe dane na ich wyjściach w określonych momentach czasowych. Prowadzi to w efekcie do ich przekłamania w trakcie wymiany pomiędzy wspomnianymi grupami, o czym szerzej można poczytać na stronie pod adresem <https://go.gl/8bZgZY>). W przypadku woltomierza problem ten jednak nie zachodzi – sygnały zegarowe bowiem są między sobą zsynchronizowane (są one tworzone na bazie wspólnego sygnału, którego jakiegokolwiek zmiany automatycznie przekładają równomiernie na zmiany w obydwu sygnałach). Możliwe staje się w ten sposób zmniejszenie użycia zasobów logicznych układu *FPGA* (nie trzeba bowiem uwzględniać mechanizmów chroniących wymieniane dane przed przekłamaniami – jest to zarazem drugi możliwy do zauważenia aspekt opisywanego woltomierza).

## Moduły związane z obsługą ADC

W przypadku tej grupy obecne są dwa poniższe moduły:

- *ADCSerwer*,
- *RejestrRownolegloRownolegly*.

Pierwszy z nich odpowiedzialny jest za bezpośrednią obsługę przetwornika, przy czym chodzi tu o wysyłanie do niego w sposób ciągły żądań odczytu poziomów kwantyzacji z kanału, do którego dołączone jest źródło mierzonego napięcia wraz z regulowanym potencjometrem (kanałem tym jest kanał 0 przetwornika). W **tabeli 3** przedstawiono sygnały wspomnianego modułu.

Jak można zauważyć są tu jak najbardziej obecne sygnały wymienione przy okazji omawiania pętli *PLL*, a także oprócz nich sygnały związane ze zwracaniem przez moduł poziomami kwantyzacji, oznajmiające ich zwrot i resetujące sam moduł (w przypadku ostatniego z nich występuje na stałe poziom wysoki, zapewniający ciągłą pracę przedstawionego modułu).

Drugi z modułów natomiast odpowiedzialny jest za zwracanie jedynie wybranych poziomów kwantyzacji. Nie chodzi jednak o świadome ingerowanie w zwracane wyniki, a pomijanie zbyt dużej ich ilości (zgodnie z wcześniejszym opisem przetwornik zwraca aż milion poziomów w ciągu sekundy, co sprawia, że zmieniają się one nadzwyczaj szybko

**Tabela 3. Wyprowadzenia modułu ADCSerwer**

Port	Funkcja
Wejscie_zegarowe_ADC	Wejściowy sygnał zegarowy „rdzenia” przetwornika ADC
Wejscie_zegarowe_PLL	Wejściowy sygnał zegarowy modułów obsługujących „rdzeń” przetwornika ADC
Wejscie_PLL_Locked	Wejście sygnału gotowości pętli do działania
Wejscie_resetu	Reset modułu
Wyjscie_gotowosci_slowa_danych_ADC	Sygnał oznajmiający, że w danej chwili czasowej został zwrócony konkretny poziom kwantyzacji
Slowo_danych_ADC	Poziom kwantyzacji na wyjściu modułu zwrócony przez przetwornik ADC

i nie jest możliwe sensowne ich odczytanie). Tym samym realizuje on proces zwany decymacją, w oparciu o prosty rejestr równoległo-równoległy (PIPO), pod wpływem jednotaktowych impulsów wytwarzanych przez pierwszy z trzech wspomnianych wcześniej generatorów opóźnień impulsów. W tabeli 4 przedstawiono parametry i sygnały omawianego modułu.

Nietrudno domyślić się, że jest to w rzeczywistości rejestr, który wyprowadza dane w momencie jego aktywacji (odbywa się ona w momencie pojawienia się sygnału oznajmiającego o zwrocie poziomów kwantyzacji przez moduł ADCSerwer). Z kolei w przypadku liczby stopni rejestru chodzi o wprowadzenie dodatkowego, opcjonalnego opóźnienia, jeśli zachodzi taka potrzeba (opcja ta nie została jednak wykorzystana w przypadku woltomierza, wobec czego parametr ten jest na stałe ustawiony na wartość równą 1, a tym samym w module występuje wyłącznie jeden rejestr równoległo-równoległy – nie oznacza to jednak braku możliwości jej zmiany, szczególnie gdy priorytetem jest uzyskanie możliwie jak najmniejszego stopnia podziału sygnału zegarowego, doprowadzonego do modułu). Oczywiście i tu występuje sygnał resetu, który podobnie jak w przypadku pętli PLL ustawiony jest na stan niski (w celu zapewnienia ciągłej pracy modułu).

## Moduły wyznaczania wyników mierzonego napięcia

Również i tu występują dwa moduły – są to jednak tym razem:

- *JednostkaObliczeniowa*,
- *KonwerterBinToBcd*.

Najważniejszym z nich jest pierwszy, który zamienia zwracane przez moduł ADCSerwer poziomy kwantyzacji na konkretne wartości mierzonego napięcia, wyrażone w woltach. Zamiana ta odbywa się zgodnie ze wzorem 1, o postaci:

**Tabela 4. Parametry i wyprowadzenia modułu RejestrRownolegloRownolegly**

Parametry	Funkcja
Dlugosc_slowa	Liczba bitów, którymi odznaczają się dane wejściowe (tj. poziomy kwantyzacji zwracane przez moduł ADCSerwer)
Liczba_stopni_rejestru	Liczba rejestrów równoległo-równoległych połączonych ze sobą w sposób szeregowy (kaskadowy)
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset modułu
Wejscie_aktywacji	Aktywacja modułu
Wejscie_rejestru	Dane wejściowe (poziomy kwantyzacji) modułu
Wyjscie_rejestru	Dane wyjściowe (poziomy kwantyzacji) modułu

**Wartość napięcia = (Wartość z przetwornika ADC/4096)·2,5 V (1)** gdzie: *Wartość z przetwornika ADC* przyjmuje wartości z przedziału całkowitych wartości liczbowych od 0 do 4095 (czyli wszystkie możliwe poziomy kwantyzacji przetwornika ADC).

Wzór ten (pochodzący z oficjalnej dokumentacji przetwornika, dostępnej na stronie pod adresem: <https://goo.gl/uMHK8B>, strony 9 i 10) wyznaczany jest w oparciu o logikę stałoprzecinkową, przy czym na część całkowitą przeznaczone są 2 bity (ze względu na fakt, że maksymalne mierzone napięcie wynosi 2,5 V, a tym samym maksymalną liczbą jego części całkowitej jest liczba 2, wymagająca do opisu jedynie dwóch bitów – 2 bowiem w zapisie binarnym to „10”), natomiast na część po przecinku łącznie 66 bitów (jest to ilość pozwalająca w sposób praktycznie jednoznaczny (z pomijalnie małym błędem) na zapisanie części ułamkowej dowolnej liczby, z dokładnością równą  $1 \cdot 10^{-4}$ ). Tym samym użytych jest łącznie 68 bitów, co wbrew pozorom nie jest znacząco dużą liczbą (jest to zarazem uzasadnienie zastosowania logiki stałoprzecinkowej zamiast zmiennoprzecinkowej, która z natury wymaga do zapisu mniejszej ilości bitów). Niezależnie jednak od tego wyznaczany w ten sposób wynik rozdzielany jest na dwie odrębne liczby – jedną obejmującą część całkowitą wyniku oraz drugą obejmującą jej część ułamkową. O ile w pierwszym przypadku wystarczające jest odrzucenie części ułamkowej wyniku, o tyle w drugim poza odrzuceniem jego części całkowitej konieczne jest jeszcze przemnożenie jego części całkowitej przez 10000, dzięki czemu możliwe jest jej potraktowanie jako łatwiejszej w operowaniu liczby całkowitej (jak można zauważyć będzie to zawsze liczba rzędu  $10^3$ , co da w ostateczności wynik pomiaru określony z dokładnością do  $1 \cdot 10^{-4}$  V). W tabeli 5 przedstawiono parametry oraz sygnały modułu.

Jak można zauważyć, moduł ten odznacza się wieloma parametrami, które w zależności od potrzeb można zmienić (np.: napięcie referencyjne przetwornika ADC, liczba poziomów kwantyzacji jakie może on zwrócić, czy też to przez jaką potęgę liczby 10 ma zostać wymnożona stałoprzecinkowa część ułamkowa obliczonego wyniku). Krótko mówiąc: moduł ten może być z powodzeniem zastosowany w przypadku innych niż zawarty w układzie FPGA zestawu *maXimator* przetworników ADC. Oczywiście również i on może zostać w każdej chwili zresetowany, co nie jest jednak czynione (domyślnie występuje w jego ramach stan niski, zapewniający ciągłą pracę przedstawionego modułu).

Niezależnie od tego zarówno rozdzielona część całkowita wyniku, jak i jego rozdzielona część ułamkowa wyrażone są w naturalnym kodzie binarnym, co nie jest wygodnym sposobem ich dalszej prezentacji. W związku z tym wymagają one zamiany na wygodniejszą postać, przy czym najlepszym z rozwiązań jest ich zamiana na kod BCD wyrażającą każdą dowolną liczbę dziesiętną w postaci ciągu cyfr dziesiętnych, które je budują, a które to zapisane są odrębnie w postaci binarnej. Zadanie to realizuje drugi z przytoczonych modułów, realizując prosty algorytm przesuwania liczby wyrażonej w naturalnym kodzie binarnym i dodawania trójki do każdej tetrady bitów liczby konwertowanej, gdy jest ona równa co najmniej 5 (wyjaśnienie algorytmu można znaleźć na stronie pod adresem <https://goo.gl/fFMWe9>). Parametry oraz sygnały przedstawionego modułu zawiera tabela 6.

Również i tu istnieje możliwość dokonania resetu, co jednak nie jest czynione z powodów takich jak w przypadku modułu *JednostkaObliczeniowa*. Niemniej w obrębie woltomierza zastosowane zostały łącznie dwa tego rodzaju moduły (jeden przypadający na rozdzieloną część całkowitą wyniku, drugi natomiast na jego rozdzieloną część ułamkową).

## Rozdzielczość i niepewność woltomierza – wnioski płynące z modułów wyznaczania wyników mierzonego napięcia

Spoglądając jeszcze raz na wzór (1) można zauważyć, że podstawiając kolejne wartości poziomów kwantyzacji obliczone napięcie będzie rosło z krokiem równym  $610,35 \mu\text{V}$  (źródło <https://goo.gl/Q7V1zX>, strona 9). Jeśli weźmiemy pod uwagę przyjętą we wcześniejszych

Tabela 5. Parametry i wyprowadzenia modułu JednostkaObliczeniowa

Parametry	Funkcja
Liczba_bitow_slowa_przetwornika_ADC	Liczba bitów, którymi odznaczają się dane wejściowe (tj. poziomy kwantyzacji zwracane przez moduł ADCSerwer)
Liczba_bitow_czesci_calkowitej	Liczba bitów przypadająca na część całkowitą wydzieloną z wyniku obliczanego przez moduł
Liczba_bitow_czesci_ulamkowej	Liczba bitów przypadająca na część ułamkową wydzieloną z wyniku obliczanego przez moduł
Wejscie_wyboru_trybu	Decyzja czy obliczony wynik ma być zwrócony na zboczach narastających zegara czy niezależnie od nich (stan wysoki jeśli tak, stan niski jeśli nie)
Napiecie_referencyjne	Napięcie odniesienia przetwornika ADC (domyślnie równe 2,5 V)
Lewa_granica_napiecia_referencyjnego	Lewy graniczny indeks stałoprzecinkowego zapisu napięcia referencyjnego
Prawa_granica_napiecia_referencyjnego	Prawy graniczny indeks stałoprzecinkowego zapisu napięcia referencyjnego
Liczba_poziomow_kwantyzacji_przetwornika_ADC	Całkowita liczba poziomów kwantyzacji możliwych do zwrócenia przez przetwornik ADC (domyślnie równa $2^{12} = 4096$ )
Liczba_miejsc_po_przecinku	Potęga liczby 10, decydująca o tym o ile należy wymnożyć otrzymaną część ułamkową wyniku (zgodnie z opisem w tekście równa 4)
Liczba_bitow_dla_miejsc_po_przecinku	Liczba bitów przypadająca na wymnożoną część (domyślnie przez 10000)
Liczba_bitow_dla_obliczen_miejsc_po_przecinku	Liczba bitów przypadająca na zapis stałoprzecinkowy części ułamkowej obliczanego wyniku
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset jednostki obliczeniowej
Wejscie_slowa_przetwornika_ADC	Daje wejściowe (poziomy kwantyzacji)
Wyjscie_liczby_obliczonej	Wyjście zawierające rozdzieloną część całkowitą i wymnożoną ułamkową wyniku (obydwie liczby ułożone obok siebie)

rozważaniach dokładność pomiaru równą  $1 \cdot 10^{-4}$  V możemy zaakceptować ten krok, uzyskując w efekcie wartość równą  $6 \cdot 10^{-4}$  V, która jest zarazem ostateczną rozdzielczością pomiarową opracowanego woltomierza. Należy jednak mieć przy tym na względzie to, że wynik pomiaru nią określony obciążony jest niepewnością, którą dla uproszczenia można przedstawić zgodnie ze wzorem 2.

**Błąd kwantyzacji woltomierza =  $\pm 0,5 \cdot \text{pojedynczy krok napięcia przed zaokrągleniem} = \pm 0,5 \cdot 610,35 \mu\text{V} = 305,175 \mu\text{V}$  (2)**

W praktyce niepewność ta zależy od znacznie większej ilości czynników (np. od zniekształceń charakterystyki przetwornika czy zjawisk zachodzących w dołączonym do niego torze wejściowym). Jednak z racji braku stosownych w tym względzie danych czynniki te zostały pominięte, a tym samym nie zostały w jej ramach uwzględnione.

## Moduły odpowiedzialne za przedstawienie wyników w oparciu o standard graficzny VGA

W przeciwieństwie do dotychczas przedstawionych grup obecne są tu trzy moduły. Są to:

- WyświetlaczVGA,
- ROM,
- SterownikVGA.

Pierwszy odpowiedzialny jest za generację dwuwymiarowej (2D) grafiki, związanej z przedstawianiem wyników pomiarów mierzonego napięcia. Grafika to nic innego jak biały, pięciopozycyjny wyświetlacz alfanumeryczny o czarnych znakach,



0,8038V

Rysunek 4. Pięcipozycyjny wyświetlacz alfanumeryczny, generowany przez moduł WyświetlaczVGA

zawarty na ciemnoniebieskim tle. Wyświetlacz ten przedstawiono na **rysunku 4**.

Jak można zauważyć wyniki przedstawiane są z dokładnością do czterech miejsc po przecinku, zgodnie z przytoczoną wcześniej dokładnością pomiarową. Co więcej poza samymi cyframi obecny jest znak przecinka rozdzielający część całkowitą wyniku od jego części ułamkowej oraz znak jednostki napięcia tj. „V” (również zgodnie z wcześniejszymi ustaleniami).

W **tabeli 7** przedstawiono parametry i wyprowadzenia omawianego modułu.

W oparciu o nią można stwierdzić, że odznacza się on dużą funkcjonalnością, który w zależności od potrzeb można dowolnie

Tabela 6. Parametry i wyprowadzenia modułu KonwerterBinToBcd

Parametry	Funkcja
Liczba_bitow_slowa_wejscowego	Liczba bitów, przypadająca na dane wejściowe
Liczba_cyfr_dziesietnych_slowa_wyjsciowego	Liczba cyfr dziesiętnych, którą odznacza się liczba zamieniona z naturalnego kodu binarnego na kod BCD
Wejscie_wyboru_trybu	Decyzja czy obliczony wynik ma być zwrócony na zboczach narastających zegara czy też niezależnie od nich (stan wysoki jeśli tak, stan niski jeśli nie)
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset konwertera
Slowo_wejscowe	Dane wejściowe konwertera (tj. liczba przed zamianą z naturalnego kodu binarnego na kod BCD)
Slowo_BCD	Dane wyjściowe konwertera (tj. liczba zamieniona z naturalnego kodu binarnego na kod BCD)



modyfikować (można np.: określić dokładne jego położenie na ekranie, można określić wielkość znaków jaka ma być wyświetlana, czy też można ustawiać dowolne kolory zarówno samych znaków, jak i wyświetlacza oraz tła go otaczającego). Co więcej, jest on dołączony do drugiego z wymienionych wcześniej modułów, tj. do pamięci ROM, która przechowuje wszystkie jego znaki. Pamięć ta (przeznaczona domyślnie wyłącznie do odczytu), opiera się w swej pracy na adresach jednoznacznie identyfikujących każdy jej bit (mamy tu zatem do czynienia z adresacją po bitach, zwaną inaczej: adresacją bitową). Na tej podstawie zwraca ona na każdym kolejnym zboczku następującym doprowadzonego do niej sygnału zegarowego dane, będące pojedynczymi pikselami wspomnianych wcześniej znaków. Innymi słowy odznacza się ona sygnałami takimi jak w tabeli 8.

Samych danych w niej jest tyle ile wynosi liczba pikseli przypadająca na jeden znak, pomnożona przez liczbę przechowywanych w omawianej pamięci znaków (w przypadku wyświetlacza znaki odznaczają się wielkością 60×100 pikseli (czcionka CourierPS, rozmiar 72), co w połączeniu z tym, że jest ich łącznie dwanaście daje łączną ilość 72000 bitów, jaką odznacza się opisywana pamięć. Mówiąc jeszcze inaczej: przechowuje ona 72000 pikseli związanych ze znakami wyświetlacza alfanumerycznego, a tym samym każdy pojedynczy jej bit opisuje jeden piksel znaku, przy czym, jeśli jest on równy zeru dynamicznie rysowany jest właściwy znak, natomiast jest on równy zeru rysowane jest jego tło, które jest białe, podobnie jak wyświetlacz). Wiemy zatem już gdzie są ulokowane znaki wyświetlacza. Jak są one natomiast w nim przedstawiane? Nietrudno nie zauważyć, że muszą one zostać pierw odczytane ze wspomnianej pamięci, zanim

**Tabela 7. Parametry i wyprowadzenia modułu WyświetlaczVGA**

Parametry	Funkcja
Początek_x	Położenie w poziomie lewego górnego rogu wyświetlacza alfanumerycznego
Początek_y	Położenie w pionie lewego górnego rogu wyświetlacza alfanumerycznego
Szerokosc_jednej_cyfr	Szerokość pojedynczej cyfry wyświetlacza alfanumerycznego (w pikselach)
Wysokosc_jednej_cyfr	Wysokość pojedynczej cyfry wyświetlacza alfanumerycznego (w pikselach)
Liczba_cyfr_przed_przecinkiem	Liczba cyfr wyświetlanej liczby przed przecinkiem
Liczba_cyfr_po_przecinku	Liczba cyfr wyświetlanej liczby po przecinku
Szerokosc_wejscia_cyfr	Liczba bitów przypadająca na dane wejściowe modułu (tj. liczby przeznaczone do wyświetlenia)
Szerokosc_adresu_pamieci_znakow	Liczba bitów przypadająca na adres pamięci ROM, zawierającej znaki wyświetlacza alfanumerycznego
Szerokosc_wyjscia_danych_pamieci_znakow	Liczba bitów przypadająca na dane zwracane przez pamięć ROM, zawierającą znaki wyświetlacza alfanumerycznego
Kolor_tla_znakow_RGB	Kolor tła znaków wyświetlacza alfanumerycznego (trójbitowa wartość oznaczająca kolejno barwy: czerwoną, zieloną i niebieską)
Kolor_tla_ekranu_RGB	Kolor tła otaczającego wyświetlacz alfanumeryczny (trójbitowa wartość oznaczająca kolejno barwy: czerwoną, zieloną i niebieską)
Kolor_znakow_RGB	Kolor znaków wyświetlacza alfanumerycznego (trójbitowa wartość oznaczająca kolejno barwy: czerwoną, zieloną i niebieską)
Polozenie_przecinka_w_pamieci	Położenie pierwszego bitu znaku przedstawiającego przecinek w wyświetlaczu alfanumerycznym w pamięci ROM
Polozenie_jednostki_w_pamieci	Położenie pierwszego bitu znaku przedstawiającego jednostkę napięcia w wyświetlaczu alfanumerycznym w pamięci ROM
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset wyświetlacza
Wejscie_polozenia_w_pionie	Położenie w pionie aktualnie wyświetlanego na urządzeniu piksela
Wejscie_polozenia_w_poziomie	Położenie w poziomie aktualnie wyświetlanego na urządzeniu piksela
Wejscie_obszaru_aktywnego	Sygnał aktywacji modułu (informacja o tym, że można generować grafikę wyświetlacza alfanumerycznego w danej chwili)
Wyjscie_koloru_czerwonego	Wyjście barwy czerwonej
Wyjscie_koloru_zielonego	Wyjście barwy zielonej
Wyjscie_koloru_niebieskiego	Wyjście barwy niebieskiej
Adres_rom	Wyjście adresu pamięci ROM
Odczyt_rom	Sygnał odczytu danych z pamięci ROM
Wejscie_liczb	Dane wejściowe wyświetlacza (tj. liczba jaka ma zostać na nim przedstawiona)

Tabela 8. Wyprowadzenia modułu ROM

Port	Funkcja
Address	Adres
Clock	Wejściowy sygnał zegarowy
Q	Dane zwracane przez moduł (tj. pojedyncze piksele znaków wyświetlacza alfanumerycznego)

dojdzie do ich docelowego wyświetlenia. Przekładając to na sygnały zegarowe: odczyt z niej musi nastąpić przed zboczami narastającymi sygnału zegarowego, doprowadzonego do ostatniego z przedstawionych modułów (tj. *SterownikVGA*). Jak można to poczynić? Otóż wystarczy doprowadzić do modułu *WyświetlaczVGA* sygnał zegarowy o częstotliwości dwukrotnie większej od częstotliwości wspomnianego wcześniej sygnału zegarowego (wówczas na jednym zboczach wysyłamy adres związany z danym pikselem znaku, natomiast na drugim odbieramy odczytany piksel z pamięci ROM i przesyłamy go do modułu *SterownikVGA*). Nie jest to jednak idealne rozwiązanie – odczytywany piksel bowiem wyświetlany jest z opóźnieniem, równym jednemu taktowi sygnału zegarowego modułu *SterownikVGA* (ma to związek z tym, że na zboczach, na którym dochodzi do przesłania piksela do wspomnianego modułu pamięć ROM dopiero odczytuje nadesłany adres, a tym samym zwrot oczekiwanego piksela następuje już w momencie przeznaczonym na wyświetlanie następnego piksela na urządzeniu wyświetlającym). Nie jest to jednak aż tak istotne, szczególnie w przypadku grafiki statycznej, którą wyświetlacz alfanumeryczny w istocie jest. Bez względu jednak na to zarówno odczyt pikseli jak i ich przesłanie następuje

na zboczach narastających sygnału zegarowego, o częstotliwości 50 MHz (sygnał pochodzący jak wiemy z pętli PLL).

Pozostaje jeszcze do omówienia wstępnie już napomknięty moduł *SterownikVGA*, który jak się dało zauważyć odpowiada za przesyłanie danych graficznych do urządzenia wyświetlającego otrzymywanych z modułu *WyświetlaczVGA* (jest to tym samym zgodnie z nazwą pełnoprawny sterownik graficzny). Co więcej jest on związany ze wspomnianym już wcześniej standardem graficznym VGA, w przypadku którego odznacza się poniższymi parametrami:

- rozdzielczość przesyłanego obrazu: 640×480 pikseli,
- częstotliwość jego odświeżania: 59 lub 60 Hz,
- głębokość kolorów: 8-bitowa (po jednym bicie na barwę czerwoną, zieloną i niebieską).

W celu ich spełnienia (zwłaszcza dwóch pierwszych) omawiany moduł funkcjonuje w oparciu o sygnał zegarowy o częstotliwości równej 25 MHz, która to pozyskiwana jest z sygnału zegarowego o częstotliwości równej 50 MHz, z zastosowaniem drugiego z przedstawionych wcześniej generatorów opóźnień impulsów (jak można zauważyć generacja jednotaktowych impulsów co drugi impuls wejściowego sygnału zegarowego jest tożsame z zastosowaniem dzielnika częstotliwości (preskalera) przez dwa – stąd też wynika pierwotnie użyta nazwa parametru związanego ze stopniem podziału)). Co więcej pierwsze dwa parametry zostały dobrane arbitralnie (według uznania autora), przy czym w przypadku drugiego z nich nie chodzi o możliwość wyboru wymienionych przy nim wartości przez użytkownika – zamiast tego chodzi o wspomniany już sygnał zegarowy o częstotliwości równej 25 MHz, która w rzeczywistości jest zaokrąglona i nie jest w pełni zgodna z oficjalnymi wartościami przewidzianymi dla rozdzielczości 640×480 pikseli (w praktyce

Tabela 9. Parametry i wyprowadzenia modułu SterownikVGA

Parametry	Funkcja
Długosc_slowa_koloru_czerwonego	Liczba bitów przypadająca na dane wejściowe, związane z barwą czerwoną
Długosc_slowa_koloru_zielonego	Liczba bitów przypadająca na dane wejściowe, związane z barwą zieloną
Długosc_slowa_koloru_niebieskiego	Liczba bitów przypadająca na dane wejściowe, związane z barwą niebieską
Rozmiar_obrazu_w_poziomie	Szerokość wyświetlanego na urządzeniu obrazu
Rozmiar_obrazu_w_pionie	Wysokość wyświetlanego na urządzeniu obrazu
Dolna_granica_marginesu_w_poziomie	Długość marginesu występującego przed impulsem synchronizacji poziomej
Gorna_granica_marginesu_w_poziomie	Długość marginesu występującego po impulsie synchronizacji poziomej
Dolna_granica_marginesu_w_pionie	Długość marginesu występującego przed impulsem synchronizacji pionowej
Gorna_granica_marginesu_w_pionie	Długość marginesu występującego po impulsie synchronizacji pionowej
Długosc_impulsu_synchronizacji_w_pionie	Długość impulsu synchronizacji pionowej
Długosc_impulsu_synchronizacji_w_poziomie	Długość impulsu synchronizacji poziomej
Tryb_dolaczenia_modulu_zewnetrznego	Decyzja czy moduł generujący wyświetlany obraz jest dołączony do sterownika synchronicznie czy asynchronicznie ('1' oznacza dołączenie synchroniczne)
Port	Funkcja
Wejscie_zegarowe	Wejściowy sygnał zegarowy
Wejscie_resetu	Reset wyświetlacza
Wejscie_koloru_czerwonego	Wejście danych związane z barwą czerwoną
Wejscie_koloru_zielonego	Wejście danych związane z barwą zieloną
Wejscie_koloru_niebieskiego	Wejście danych związane z barwą niebieską
Wyjscie_synchronizacji_w_pionie	Wyjście synchronizacji pionowej
Wyjscie_synchronizacji_w_poziomie	Wyjście synchronizacji poziomej
Wyjscie_koloru_czerwonego	Wyjście danych związane z barwą czerwoną
Wyjscie_koloru_zielonego	Wyjście danych związane z barwą zieloną
Wyjscie_koloru_niebieskiego	Wyjście danych związane z barwą niebieską
Wyjscie_polozenia_w_pionie	Wyjście położenia pionowego aktualnie wyświetlanego na urządzeniu piksela
Wyjscie_polozenia_w_poziomie	Wyjście położenia poziomego aktualnie wyświetlanego na urządzeniu piksela
Wyjscie_obszaru_aktywnego	Wyjście sygnału informującego o momencie, w którym można przesyłać do urządzenia generowaną grafikę

**Tabela 9. Parametry i wyprowadzenia modułu WyświetlaczSiedmiosegmentowy**

Port	Funkcja
Clk	Wejściowy sygnał zegarowy
Hex_data_in	Dane wejściowe wyświetlacza
Hex_data_out	Dane wyjściowe wyświetlacza
Hex_switch_out	Aktywacja segmentów wyświetlacza

bowiem potrzebny jest sygnał zegarowy o częstotliwości 25,175 MHz, który nie jest łatwy w uzyskaniu, nawet za pomocą pętli PLL). Tym samym niektóre urządzenia wyświetlające mogą pod jej wpływem różnie po prostu zinterpretować częstotliwość odświeżania obrazu (nie ma to jednak większego znaczenia, szczególnie, że po pierwsze ludzkie oko nie dostrzeże różnicy, a po drugie natomiast większość współczesnych urządzeń radzi sobie bez problemu zarówno z jedną, jak i z drugą częstotliwością odświeżania obrazu). Ostatni z parametrów natomiast ma związek z tym, że w zestawie *maXimator* obecny jest trzybitowy przetwornik, dołączony do gniazda standardu VGA (sprawia to, że każdą z barw przestrzeni RGB przypada wyłącznie jeden bit, co w sumie daje  $2^3 = 8$  możliwych do otrzymania kolorów). W tabeli 9 przedstawiono parametry i sygnały związane ze sterownikiem.

Podobnie jak w przypadku wyświetlacza alfanumerycznego jest to moduł o dużych możliwościach konfiguracyjnych i tym samym można go dostosować w zależności od własnych potrzeb (parametry związane z poszczególnymi rozdzielczościami, wliczając w to częstotliwości wymaganych sygnałów zegarowych można znaleźć na stronie pod adresem: <http://bit.ly/2pTKl9c>). Należy jednak pamiętać, aby nie zapomnieć w tym względzie zmienić częstotliwości sygnału zegarowego, generowanego przez pętlę PLL!).

## Moduły odpowiedzialne za przedstawienie wyników na wyświetlaczu 7-segmentowym

W przypadku ostatniej z grup woltomierza mamy do czynienia wyłącznie z jednym modułem, którym jest *WyświetlaczSiedmiosegmentowy*. Moduł ten jest zbiorem trzech następujących elementów:

- rejestr przesuwany z wartością początkową binarną równą „1000”,
- dekodery przekazujący w danej chwili dane bity danych wejściowych wyświetlacza,
- dekodery zamieniający wartość bitów danych wejściowych wyświetlacza na kod wyświetlacza 7-segmentowego.

Oprócz tego zawiera on trzeci i ostatni generator opóźnień impulsów, poprzez który następuje jego odświeżanie (zgodnie z wcześniejszym opisem do woltomierza dołączony jest wyświetlacz multipleksowany, czyli taki, w którym dochodzi do cyklicznego przełączania wszystkich jego segmentów z poziomu dołączonych

**Tabela 10. Szczegóły dotyczące implementacji woltomierza**

Aspekt implementacji	Dane związane z aspektem
Liczba użytych elementów logicznych	888 na 8064 dostępnych w układzie FPGA (11% wszystkich dostępnych elementów)
Liczba użytych przerzutników D	265
Liczba użytych układów mnożących	8 na 48 (17% wszystkich dostępnych układów)
Liczba użytych pętli PLL	1 na 2 (50% wszystkich dostępnych pętli)
Pojemność pamięci ROM	72000 bitów na 387082 dostępnych (19% wszystkich dostępnych bitów bloków pamięciowych w układzie FPGA)

do niego katod. Tym samym wspomniane odświeżanie sprowadza się do szybkości przełączania segmentów wyświetlacza na tychże katodach). Zasada jego działania jest następująca: rejestr przesuwany „przerzuca” na każdym narastającym zboczku sygnału zegarowego zawartą w jego wartości początkowej binarną „jedynekę”, która decyduje o tym, który segment ma zostać w danej chwili uaktywniony (tym samym ich identyfikacja odbywa się na podstawie położenia „jedynki” w wartości początkowej). Następnie określane jest to, które bity danych wejściowych powinny być w danej chwili pobrane (czynione jest to w oparciu o drugi z wymienionych elementów wyświetlacza, na podstawie aktualnego położenia przesuwanej w rejestrze „jedynki” – dla przykładu: jeśli w rejestrze jest obecna wartość równa „1000” to odczytane zostają pierwsze cztery bity danych wejściowych, jeśli „0100” to następne cztery bity itd.). Ostatecznie dane które są w danej chwili pobierane zamieniane są na określony kod wyświetlacza siedmiosegmentowego, po uprzednim ich podaniu na anody wyświetlacza. W efekcie na danym segmencie pojawia się oczekiwana cyfra (i tak w przypadku pozostałych segmentów). W tabeli 9 przedstawiono parametry i sygnały przedstawionego modułu.

Warto zauważyć, że w stosunku do urządzenia wyświetlającego wynik pomiaru napięcia prezentowany jest jedynie z dokładnością do trzech miejsc po przecinku (ze względu na obecność zgodnie z nazwą dołączonego do woltomierza wyświetlacza jedynie czterech segmentów. W ogólności jednak nie ma to znaczenia, ponieważ zmiana dotyczy wyłącznie samego wyświetlania, a nie sposobu prowadzonych obliczeń). Domyślnie też nie jest wyświetlana kropka rozdzielająca część całkowitą wyniku od jego części ułamkowej (nie jest to na dłuższą metę potrzebne, szczególnie, jeśli wiemy, że na część całkowitą będzie zawsze składać się wyłącznie jedna cyfra, natomiast na część ułamkową trzy cyfry). Dodatkowo również nie jest do niego dołączony sygnał zegarowy z pętli PLL, zamiast którego doprowadzony jest sygnał pochodzący bezpośrednio ze źródła zegarowego, zawartego w zestawie *maXimator* (uzasadnieniem jest fakt, że jest to moduł końcowy, który nie przekazuje danych do kolejnych modułów, w celu ich dalszego przetwarzania – nie jest wobec tego wymagana synchronizacja ze strony pętli PLL).

## Dane praktyczne

Szczegóły dotyczące implementacji woltomierza zawarto w tabeli 10, natomiast w tabeli 11 umieszczono zbiorcze zestawienie parametrów woltomierza.

Jakub Tyburski, WAT

**Tabela 11. Zbiorcze zestawienie parametrów woltomierza**

Wielkość	Wartość
Częstotliwość wejściowego sygnału zegarowego	10 MHz
Rodzaj mierzonego napięcia elektrycznego	stałe
Zakres pomiarowy	od 0 do 2,5 V
Regulacja napięcia	Poprzez dedykowany potencjometr
Rozdzielczość pomiarowa	$6 \cdot 10^{-4}$ V
Błąd kwantyzacji	$\pm 305,175 \mu\text{V}$
Rozdzielczość obrazu VGA	640 x 480 pikseli
Częstotliwość odświeżania obrazu VGA	59 Hz lub 60 Hz
Głębokość kolorów obrazu VGA	8-kolorowa
Częstotliwość odświeżania czteropozycyjnego, multipleksowanego wyświetlacza siedmiosegmentowego	2000 Hz