



**Olimpiada Wiedzy Elektrycznej i Elektronicznej EUROELEKTRA
Rok szkolny 2015/2016**

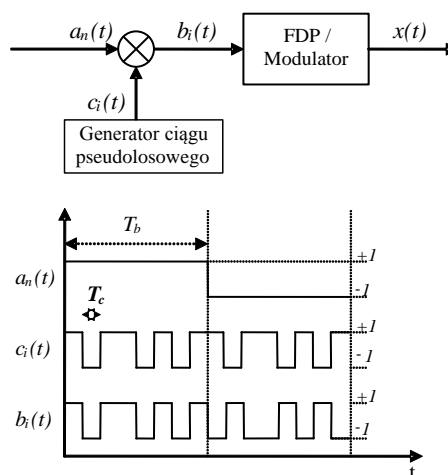
**Zadania z rozwiązaniami teleinformatyki
na zawody II stopnia**

1. Czas trwania zawodów: 120 minut.
2. II stopień olimpiady zawiera 6 zadań otwartych.
3. Należało podać poprawną odpowiedź wraz tokiem rozwiązania.
4. **Za każdą prawidłową odpowiedź uzyskuje się maksymalnie 10 punktów. Maksymalna liczba punktów za 6 zadań do zdobycia to 60 punktów.**

Lp.	Zadanie
1.	<p>Do rejestru szesnastobitowego (czyli słowa) wpisano wartość 7. Jak jest wartość dziesiętna tego słowa po przesunięciu bitów tego rejestru o 2 pozycje w lewo?</p>
Odpowiedź	<p>Objaśnienie: Po wpisaniu liczby 7, zapisane są trzy najmłodsze bity o wagach 1, 2, 3, razem 7. Po przesunięciu o jedną pozycję w lewo, zapisane są bity czwarty, trzeci i drugi, czyli bity o wagach 2, 4, 8, razem daje to wartość dziesiętną 14. Po kolejnym przesunięciu o jedną pozycję, wagi wzrosną ponownie dwukrotnie, co da wartość słowa równą 28.</p>
2.	<p>Oblicz wynik dodawania trzech wyrażeń podanych poniżej w różnych kodach liczb. L1 = 111 (dwójkowa), L2 = 3 (dziesiętna), L3 = B (szesnastkowa).</p>
Odpowiedź	<p>(wynik= 7+3+11=21)</p>
3.	<p>Opracować algorytm obliczania wartości funkcji eksponencjalnej e^x na podstawie wzoru</p> $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots, \quad x \in R.$ <p><i>Uwaga:</i> kolejne składniki szeregu $= x^n / n!$ monotonicznie maleją dla $n > x$, aż stają się pomijalnie małe, co można wykorzystać obliczając sumę szeregu. Algorytm można przedstawić w postaci schematu blokowego lub funkcji języka C, C++, C# o nagłówku <code>double exp(double x);</code> która zwraca obliczoną wartość funkcji eksponencjalnej e^x. Algorytm powinien być zoptymalizowany pod kątem minimalnego czasu obliczeń. Określić maksymalną wartość argumentu x dla której algorytm da poprawny wynik, przy założeniu że obliczenia wykonywane są na liczbach typu double o zakresie od $\pm 5.0 \times 10^{-324}$ do $\pm 1.7 \times 10^{308}$ i precyzji 15-16 cyfr znaczących.</p>

	Odpowiedź	<p>Kolejne składniki szeregu $y_n = x^n / n!$ można obliczać za pomocą wzoru rekurencyjnego $y_n = y_{n-1}x/n$. Poniższa przykładowa funkcja działa poprawnie dla nieujemnych argumentów, do największego, dla którego wynik mieści się w zakresie liczb double, $x_{\max} = \ln(1,7 \times 10^{308}) \approx 709$.</p> <pre>double exp(double x) { double y, suma; y = x; suma = 1 + y; for (int n = 2; true; n++) { y = y * x / n; if (suma + y == suma) break; suma = suma + y; } return suma; }</pre> <p>Jeśli liczyć osobno licznik i mianownik składników szeregu, to licznik lub mianownik przekroczy zakres liczb double dla $x \approx 300$. Błędem jest ewentualne liczenie silni jako liczby typu int z uwagi na mały zakres tych liczb. Funkcja (algorytm) nie działa poprawnie dla ujemnych argumentów, ale można skorzystać ze wzoru $e^{-x} = 1/e^x$.</p>
4.	Odpowiedź	<p>Opracować algorytm zliczający liczbę bitów, na których różnią się dwie 32 bitowe liczby całkowite a i b. Algorytm można przedstawić w postaci schematu blokowego lub funkcji języka C, C++, C# o nagłówku <code>int hamming(int a, int b)</code>, która zwraca zliczoną liczbę bitów, na których różnią się liczby a i b.</p> <p>Operatory logiczne w języku C, C++, C#: AND: & XOR: ^, OR: .</p> <pre>int hamming(int a, int b) { int licznik = 0, aXORb = a ^ b; int maska = 1; for (int i = 0; i < 32; i++) { if ((maska & aXORb) != 0) licznik++; maska *= 2; } return licznik; }</pre> <p>Przesuwając bit 1 w masce, zamiast mnożenia przez 2, można oczywiście posłużyć się operatorem przesunięcia bitowego, ale nie wszyscy go pamiętają.</p>

5. W systemie UMTS (3G) stosuje się transmisję z widmem rozproszonym DSSS (Direct Sequence Spread Spectrum). Zasadę rozpraszania bezpośredniego ilustruje rys.1, gdzie T_b - czas trwania bitu, T_c – czas trwania chipu a współczynnik rozpraszania SF (Spectrum Factor) $SF=8$. Stanowi logicznemu bitu 1 (jeden) odpowiada stan bipolarny +1, natomiast stanowi logicznemu bitu 0 (zero) odpowiada stan bipolarny -1. Dla przykładu z rys. 1 wyznaczyć ciąg logiczny chipów otrzymany w wyniku rozpraszania bezpośredniego bitu o stanie logicznym 1 (jeden).



Rys. 1. Schemat nadajnika i zasada rozpraszania bezpośredniego w dziedzinie czasu, FDP – filtr dolnoprzepustowy

Odpowiedź Wartość ciągu chipów należy odczytać bezpośrednio z rys.1, jako ciąg $\{b_i\}$ dla bitu $a_n=1$
Odp. Ciąg $\{10110101\}$

6. Wyznaczyć napięcie czułościowe U_c [μV] odbiornika jeżeli moc czułościowa odbiornika wynosi $P_c = -70dBm$ a $R_{we} = 50\Omega$.

Odpowiedź Uwzględniając $P_c = -70 dBm = -100dBW$ i definicję skali decybelowej otrzymamy $10\log_{10}P_c[W] = -100$, stąd $P_c=10^{-10} [W]= 100 pW$.
Ponieważ $U_c = (P_c R_{we})^{1/2}$ stąd $U_c = 70,7\mu V$
Odp. $U_c = 70,7\mu V$

Opracowali:
dr inż. Andrzej Łuksza
dr inż. Stanisław Lindner
dr inż. Krzysztof Kamiński

Sprawdził:
dr inż. Jacek Majewski

Zatwierdził:
Przewodniczący Rady Naukowej Olimpiady
dr hab. inż. Sławomir Cieślik