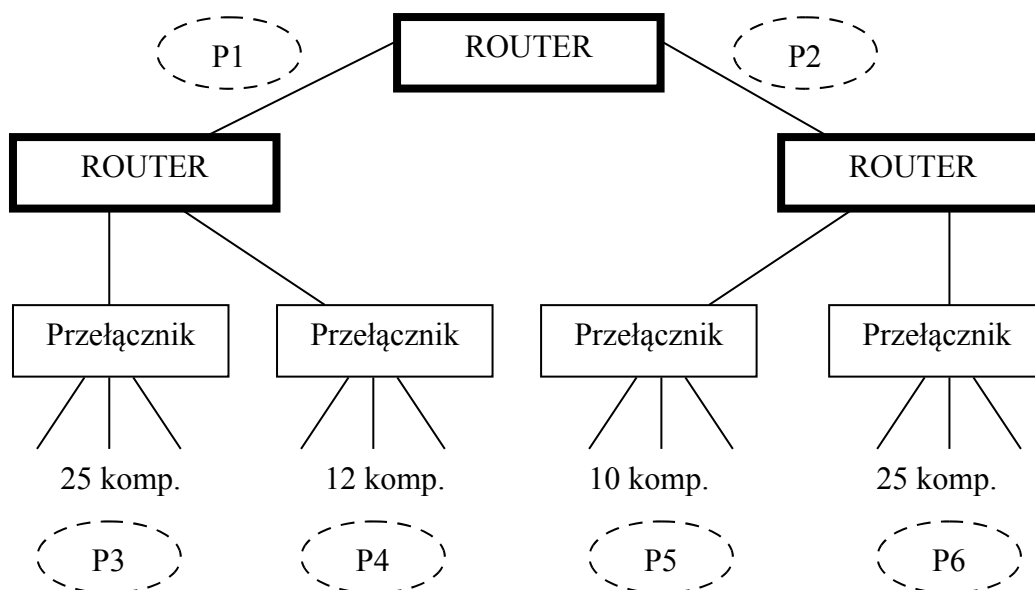


„EUROELEKTRA”
Ogólnopolska Olimpiada Wiedzy Elektrycznej i Elektronicznej
Rok szkolny 2011/2012
Rozwiązania zadań dla grupy teleinformatycznej na zawody II stopnia

ZADANIE 1

Administrator sieci komputerowej otrzymał niewielką pulę publicznych adresów IP, tj. 200.122.32.64/26 oraz 202.12.42.96/27. Należy podzielić posiadane adresy, tak aby każdy komputer w podsieciach od P1 do P6 otrzymał adres IP, przy czym dla sieci od P3 do P6 musi to być adres publiczny. W rozwiązaniu należy podać, dla każdej podsieci od P1 do P6, adres podsieci, maskę podsieci, liczbę adresów IP dostępnych dla hostów oraz pierwszy i ostatni dostępny adres IP.



Rozwiązanie

Posiadane adresy publiczne 200.122.32.64/26 oraz 202.12.42.96/27 należy podzielić pomiędzy podsieci od P3 do P6, natomiast do połączeń między routerami w podsieciach P1 i P2 należy użyć prywatnych adresów IP. Bez wykorzystania prywatnych adresów IP zadania nie da się rozwiązać.

Istnieje wiele możliwych rozwiązań, np. sieć 200.122.32.64/26 można podzielić pomiędzy P3 i P6, a 202.12.42.96/27 pomiędzy P4 i P5 albo 202.12.42.96/27 przydzielić do P3, a 200.122.32.64/26 podzielić między P4, P5 i P6 itd. Przykładowe rozwiązanie:

podsieć P1

- podsieć: 192.168.10.0 /24
- maska: 255.255.255.0
- liczba IP: 254
- adresy IP: od 192.168.10.1 do 192.168.10.254

podsieć P2

- podsieć: 192.168.20.0 /24
- maska: 255.255.255.0
- liczba IP: 254
- adresy IP: od 192.168.20.1 do 192.168.20.254

podsieć P3

- podsieć: 200.122.32.64 /27
- maska: 255.255.255.224

- liczba IP: 30
- adresy IP: od 200.122.32.65 do 200.122.32.94

podsieć P4

- podsieć: 202.12.42.96 /28
- maska: 255.255.255.240
- liczba IP: 14
- adresy IP: od 202.12.42.97 do 200.12.42.110

podsieć P5

- podsieć: 202.12.42.112 /28
- maska: 255.255.255.240
- liczba IP: 14
- adresy IP: od 202.12.42.113 do 200.12.42.126

podsieć P6

- podsieć: 200.122.32.96 /27
- maska: 255.255.255.224
- liczba IP: 30
- adresy IP: od 200.122.32.97 do 200.122.32.126

ZADANIE 2

Napisz podprogram w asemblerze mikrokontrolera AVR obliczający resztę z ilorazu dwóch liczb bez znaku. Wiadomo, że obie liczby są niezerowe, przy czym dzielna jest mniejsza, co do wartości, od liczby 128, ponadto dzielnik jest mniejszy niż dzielna. Przed wywołaniem podprogramu dzielna jest umieszczana w rejestrze R21, dzielnik zaś w R22. Przed instrukcją powrotu z podprogramu wynik należy umieścić w rejestrze R23, natomiast rejestry od R24 do R30 można swobodnie używać do zapisania tymczasowych zmiennych pomocniczych.

Uwaga: w rozwiązaniu należy podać wyłącznie kod podprogramu, bez elementów programu takich, jak inicjalizacja wektorów przerwań czy stosu. Przykładowy sposób wywołania podprogramu:

```
LDI R21, 64;
LDI R22, 10;
CALL reszta;
// wynik, w tym przypadku 4, powinien być teraz w R23
```

Wybrane polecenia asemblera AVR:

Nazwa	działanie	przykład
ADD Rd, Rr	dodawanie bez przeniesienia, $Rd \leftarrow Rd + Rr$	ADD R21, R22
ADC Rd, Rr	dodawanie z przeniesieniem, $Rd \leftarrow Rd + Rr + C$	
SUB Rd, Rr	odejmowanie bez przeniesienia, $Rd \leftarrow Rd - Rr$	SUB R21, R22
SBC Rd, Rr	odejmowanie z przeniesieniem, $Rd \leftarrow Rd - Rr - C$	
INC Rd	inkrementacja, $Rd \leftarrow Rd + 1$	INC R20
DEC Rd	dekrementacja, $Rd \leftarrow Rd - 1$	
LSL Rd	przesunięcie w lewo, w praktyce $Rd \leftarrow Rd \cdot 2$	LSL R15
LSR Rd	przesunięcie w prawo, w praktyce $Rd \leftarrow Rd / 2$	
CP Rd, Rr	porównanie wartości rejestrów, $Rd - Rr$	CP R21, R25

CPI Rd, K	porównanie wartości rejestru i stałej, Rd – K	CPI R22, 1
BREQ k	skok, jeżeli równe	BREQ gotowe
BRNE k	skok, jeżeli nie równe	
BRSH k	skok, jeżeli większe lub równe	
BRLO k	skok, jeżeli mniejsze	
JMP k	skok bezwarunkowy	JMP ponownie
RET	powrót z podprogramu	RET
MOV Rd, Rr	kopiowanie rejestru, Rd ← Rr	MOV R21, R22
LDI Rd, K	wpisanie wartości, Rd ← K	LDI R22, 1

Rd i Rr oznaczają rejestr (od R0 do R31), K liczbę, natomiast k jest etykietą.

Rozwiązanie

Najprostsza realizacja polega na wielokrotnym zmniejszaniu dzielnej o dzielnik. Kiedy pozostałość dzielnej stanie się mniejsza od dzielnika – pozostałość ta będzie resztą z dzielenia. Rozwiązanie nie jest wydajne, ale łatwe w implementacji.

```

reszta:
MOV R23, R21;    // inicjalizacja wyniku

ponownie:
CP R23, R22;     // czy dzielna mniejsza od dzielnika?
BRLO mniejsze;
SUB R23, R22;    // nie, zmniejszamy dzielną o dzielnik
JMP ponownie;    // pętla

mniejsze:
RET;
```

Wydajniejsze będzie rozwiązanie z przesuwaniem dzielnika. Wykorzystujemy w nim informację, że dzielna jest mniejsza od 128: dzielnik jest mnożony razy 2 aż zrówna się z dzielną lub ją przewyższy (może być wtedy większy od 128, lecz nadal mniejszy od 255).

```

reszta:
MOV R23, R21;    // inicjalizacja wyniku
LDI R25, 1;      // mnożnik dzielnika

razy2:
LSL R22;         // dzielnik · 2
LSL R25;         // mnożnik · 2
CP R22, R21;     // czy dzielnik mniejszy od dzielnej?
BRLO razy2;     // tak, nadal zwiększaj

ponownie:
LSR R22;         // dzielnik / 2
LSR R25;         // mnożnik / 2
CP R23, R22;     // czy dzielna mniejsza od dzielnika?
BRLO mniejsze;
SUB R23, R22;    // nie, zmniejszamy dzielną o dzielnik
mniejsze:
CPI R25, 1;      // czy mnożnik wrócił do 1?
BRNE ponownie;  // nie, powtarzaj

RET;
```

ZADANIE 3

W celu ochrony transmisji przed błędami w systemach transmisyjnych stosuje się kodowanie kanałowe. Jedną z metod takiej ochrony jest stosowanie sposobu kodowania opracowanego przez Hamminga. Polega on na przesyłaniu, oprócz bitów informacyjnych, dodatkowych bitów kontroli parzystości. W porównaniu ze standardową kontrolą parzystości, w której występuje tylko jeden bit parzystości na końcu przesyłanej informacji, w kodzie Hamminga występuje tych bitów więcej i dodatkowo są one poumieszczane na różnych pozycjach wynikowego ciągu kodowego. Jeśli spojrzymy na taki wynikowy ciąg, to numery pozycji, na których pojawiają się bity parzystości, będą potęgami liczby 2 czyli 1, 2, 4, 8, ..., przy czym pozycja nr 1 jest rozumiana jako skrajna lewa w zapisie. Pozostałe pozycje będą obsadzone kolejno przez bity informacyjne.

Każdy bit parzystości określa parzystość wynikowego kodu na wybranych i ściśle określonych pozycjach. Bit parzystości znajdujący się na pozycji nr 1 naprzemiennie sprawdza i opuszcza po jednym bicie ciągu wynikowego aż do końca tego ciągu. Bit parzystości z pozycji nr 2 naprzemiennie sprawdza i opuszcza po 2 bity ciągu wynikowego aż do końca ciągu wynikowego. Bit parzystości z pozycji nr 4 sprawdza i opuszcza naprzemiennie po 4 bity ciągu wynikowego, aż do końca tego ciągu. Następne bity parzystości dokonują operacji kontroli parzystości odpowiednio zgodnie z powyższym schematem.

Upraszczając opis algorytmu kodowania, można powiedzieć tak: każdy bit parzystości określa parzystość według wyżej opisanego wzorca, poczynawszy od swojej własnej pozycji, zakładając, że początkowo na tej pozycji był ustawiony stan 0. Przykładowo, bit parzystości o numerze pozycji 2 rozpoczyna sprawdzanie parzystości wynikowego ciągu kodowego od bitu (pozycji) nr 2 i kontynuuje je na bitach 3, 6, 7, 10, 11, itd., przy założeniu, że na pozycji nr 2 było początkowo ustawione 0 (zero logiczne).

W powyższym algorytmie kodowania bit parzystości równy 1 odpowiada nieparzystej liczbie jedynek na sprawdzanych pozycjach.

Liczba bitów parzystości w kodzie wynikowym też jest ściśle określona. Zazwyczaj wykorzystuje się minimalną liczbę bitów parzystości, jednak taką, aby każdy bit informacyjny był sprawdzany przez co najmniej dwa bity parzystości.

Zadanie polega na wyznaczeniu ciągu kodowego N zakodowanego zgodnie z kodem Hamminga, jeżeli dany jest ciąg informacyjny $\{N_x\} = (1100100010)_b$.

Rozwiązanie

Należy zastosować podaną w treści zadania procedurę wyznaczania pozycji kontrolnych i całego ciągu kodowego. W wyniku tego uzyskuje się ciąg $\{N\} = (01101000100010)_b$.

ZADANIE 4

W systemie UMTS (3G) stosuje się transmisję z widmem rozproszonym DSSS (ang. *Direct Sequence Spread Spectrum*). Rozpraszanie bezpośrednio wykonywane jest w nadajniku systemu. Jego zasada została zilustrowana na rys. a) poniżej. Czas T_b zaznaczony na przebiegach czasowych, to czas trwania przesyłanego bitu natomiast czas T_c , to czas trwania pojedynczego impulsu rozpraszającego (chipu). Stosunek tych czasów T_b/T_c równy jest współczynnikowi rozpraszania SF (*Spectrum Factor*).

W odbiorniku systemu wykonywana jest operacja odwrotna do rozpraszania, zwana kompresją. Schemat blokowy układu kompresującego pokazano na rys. b) poniżej.

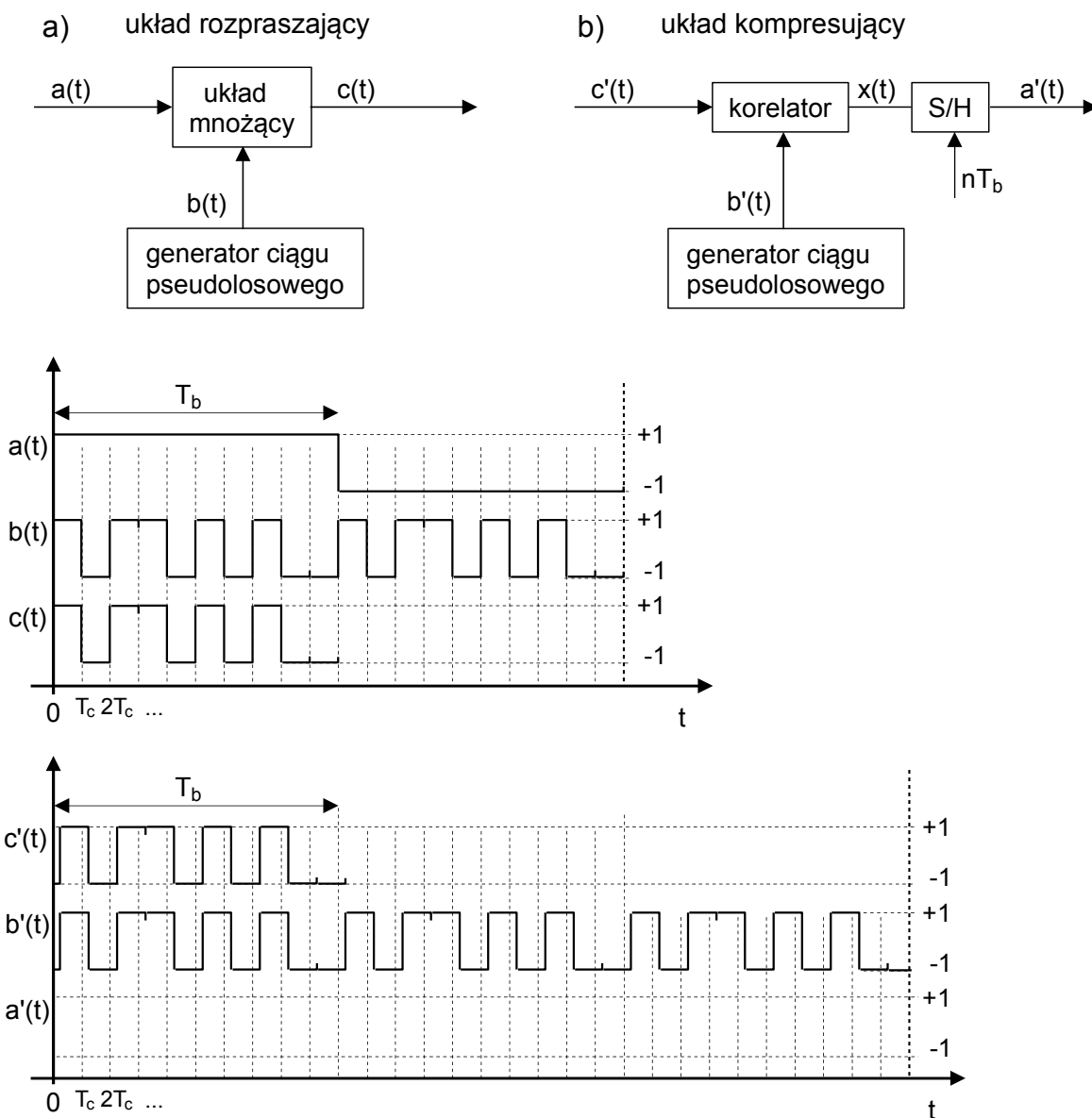
Do poprawnego działania systemu ważne jest, aby przesyłanemu bitowi „1” odpowiadał stan bipolarny +1, natomiast przesyłanemu bitowi „0” odpowiadał stan bipolarny -1. Ponadto układ próbkująco-pamiętający S/H powinien być taktowany sygnałem zegarowym nT_b , gdzie $n = 0, 1, 2, 3, \dots$

W zadaniu należy uzupełnić brakujące przebiegi chipów $c(t)$ i $c'(t)$ otrzymane w wyniku rozpraszania bezpośredniego oraz przebieg $a'(t)$ na wyjściu odbiornika przy następujących założeniach:

- sygnał $c'(t)$ jest kopią sygnału $c(t)$,
- korelator w układzie kompresującym opisany jest zależnością:

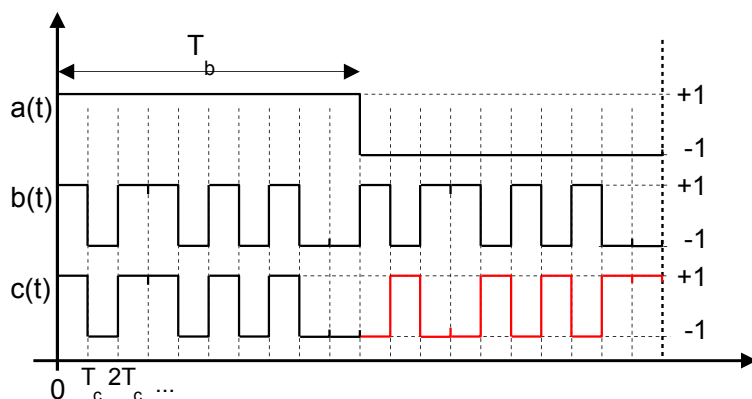
$$x(t) = \frac{1}{SF} \sum_{m=0}^{SF-1} c'(mT_c) b'(mT_c) .$$

Dla uproszczenia założono, że sygnały $c'(t)$ i $b'(t)$ są nieco opóźnione w czasie względem sygnałów $c(t)$ i $b(t)$, co widać na przebiegach.

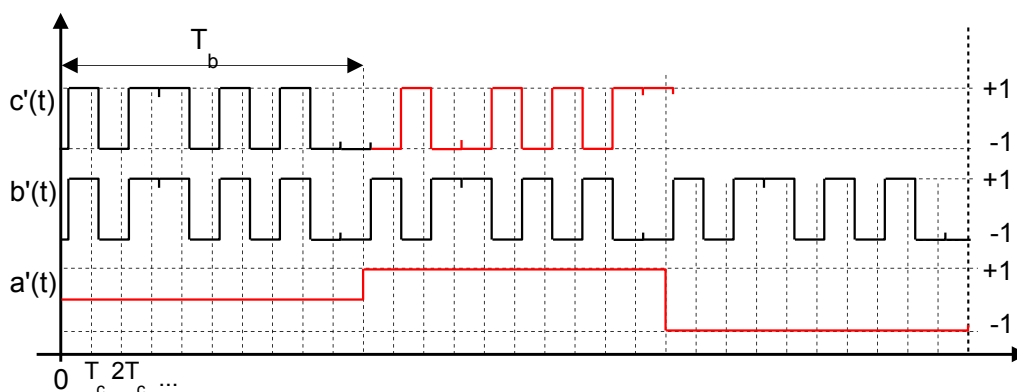


Rozwiązanie

W celu uzyskania drugiej części przebiegu $c(t)$ wystarczy zgodnie ze schematem z rysunku a) pomnożyć drugą część przebiegu $b(t)$ przez sygnał $a(t)$, czyli w tym przypadku przez wartość -1 . Rozwiązanie zaznaczono kolorem czerwonym na poniższym rysunku.



W celu uzyskania przebiegu $a'(t)$ należy poddać standardowej korelacji przebiegi $c'(t)$ i $b'(t)$ po przedziałach czasowych równych T_b . Funkcja korelacji, którą należy zastosować, została podana w treści zadania. Należy ponadto zauważyć, że wynik obliczenia funkcji korelacji jest „zatraskiwany” w układzie pamiętająco-próbkującym dopiero po czasie T_b . W związku z tym dopiero po tym czasie pojawi się na wyjściu odbiornika pierwszy wynik obliczenia korelacji sygnałów. W przedziale czasowym $(0, T_b)$ można przyjąć, że $a'(t) = 0$ przy założeniu, że przebiegi $c'(t)$ i $b'(t)$ przed chwilą $t = 0$ były równe zeru. Można również oznaczyć sygnał $a'(t)$ w tym przedziale jako nieokreślony. Rozwiązanie zaznaczono kolorem czerwonym na poniższym rysunku.



ZADANIE 5

Poziom sygnału na wejściu kanału komunikacyjnego wynosi 500 mV, natomiast poziom sygnału szumu na wyjściu tego kanału 20 mV (wartość średnia) – niezależnie od tego, czy sygnał użyteczny jest transmitowany w kanale, czy też nie. Tłumienność kanału wynosi 1dB/km. Oblicz maksymalną długość kanału przy warunku, że poziom sygnału do szumu nie spada w nim poniżej 20 dB.

Rozwiązanie

Oznaczmy poziom sygnału użytecznego i poziom sygnału szumu (w rozumieniu napięcia) na wyjściu kanału komunikacyjnego, odpowiednio, symbolami V_{SO} i V_{NO} . Z treści zadania wynika, że $V_{NO} = 20$ mV. Z powodu tłumienia sygnału użytecznego jego poziom na wyjściu kanału będzie znacznie mniejszy niż na jego wejściu. W stosunku do niezmiennego się poziomu szumu będzie on zatem najbardziej krytyczny właśnie na wyjściu kanału. Przyjęto w zadaniu, że stosunek mocy sygnału użytecznego do mocy szumu (zwany krótko stosunkiem sygnał/szum i oznaczany symbolem SNR) nie może spadać poniżej wartości 20 dB. W naszym przypadku zatem napiszemy

$$10 \log_{10} \left(\frac{V_{SO}}{V_{NO}} \right)^2 \geq 20, \text{ skąd po przekształceniu wynika, że musi być spełniona nierówność}$$

$\frac{V_{SO}}{V_{NO}} \geq 10$. Podstawiając $V_{NO} = 20 \text{ mV}$, otrzymuje się $V_{SO} \geq 200 \text{ mV}$. Teraz możemy obliczyć tłumienie sygnału użytecznego od wejścia do wyjścia kanału. Będzie ono wynosić $10 \log_{10} \left(\frac{500 \text{ mV}}{200 \text{ mV}} \right)^2 \approx 8 \text{ dB}$. Skoro tłumienność kanału wynosi 1 dB/km , 8 dB tłumienia sygnału użytecznego będzie na długości 8 km . Zatem maksymalną długość kanału wynosi 8 km .

ZADANIE 6

Założmy, że dane są transmitowane w bezszumnym kanale komunikacyjnym o paśmie 4 kHz z szybkością jeden symbol w czasie $0,5 \text{ ms}$ z wykorzystaniem transmisji wielopoziomowej. Oblicz maksymalne przepustowości tego kanału dla dwóch przypadków, gdy każdy z symboli jest kodowany za pomocą 4 bitów i 8 bitów. Czy liczbę bitów, na których można kodować symbol danych może wzrastać do nieskończoności? Odpowiedź uzasadnij. (Podpowiedź: Ponieważ rozpatrywany kanał komunikacyjny jest bezszumny, można do niego zastosować wzór Nyquista, który ma postać $r_m = 2 B \log_2 V$, gdzie r_m oznacza maksymalną przepustowość kanału bezszumnego o paśmie B natomiast V jest liczbą poziomów, jakie może przyjmować sygnał.)

Rozwiązanie

W wielopoziomowym systemie transmisyjnym, jeżeli symbol jest kodowany za pomocą 4 bitów, to daje to $2^4 = 16$ poziomów w sygnale. Natomiast w przypadku kodowania symbolu za pomocą 8 bitów mamy $2^8 = 256$ poziomów w sygnale. Zatem podstawiając te wartości do podanego w treści zadania wzoru Nyquista, otrzymujemy maksymalne przepustowości kanału w powyższych przypadkach tj.:

$$r_{m1} = 2 \cdot 4000 \log_2 2^4 = 32000 \text{ bit/s} = 32 \text{ kbit/s}$$

oraz

$$r_{m2} = 2 \cdot 4000 \log_2 2^8 = 64000 \text{ bit/s} = 64 \text{ kbit/s}.$$

Dane są transmitowane w tych przypadkach z szybkościami odpowiednio:

$$r_1 = \frac{1 \text{ symbol}}{0,5 \text{ ms}} = \frac{4 \text{ bity}}{0,5 \text{ s}} 1000 = 8000 \text{ bit/s} = 8 \text{ kbit/s}$$

oraz

$$r_2 = \frac{1 \text{ symbol}}{0,5 \text{ ms}} = \frac{8 \text{ bitów}}{0,5 \text{ s}} 1000 = 16000 \text{ bit/s} = 16 \text{ kbit/s}.$$

Oczywiście w każdym z tych przypadków szybkość transmisji jest mniejsza od maksymalnej przepustowości obliczonej odpowiednio jako r_{m1} i r_{m2} . Zauważmy dalej że, jeżeli zakodujemy symbol danych za pomocą większej liczby bitów, to naszkicowany powyżej schemat powtórzy się i nigdy szybkość transmisji w bitach na sekundę nie przekroczy maksymalnej przepustowości przy danej liczbie bitów użytej do zakodowania symbolu. Wydaje się, że tak można postępować w nieskończoność. Nie jest to jednak prawda, gdyż nie można w nieskończoność zmniejszać czasu trwania bitu oraz nie można w nieskończoność zwiększać liczby poziomów w sygnale. Na zmniejszanie czasu trwania bitu nie pozwalają skończone możliwości układów elektronicznych. Na zwiększanie liczby poziomów nie pozwala sygnał szumu, który w układach elektronicznych występuje zawsze i sprawia, że przy dużych liczbach poziomów sąsiednie poziomy sygnałów przestają być rozróżnialne.