



Programski određen radio

Tema 8

Karlo Graf
Anđela Jaković
Anamaria Vargić
Vanja Ljubobratović

Rijeka, 16.06.2023.

Projektni zadatak

- **Razviti programsku podršku koja će ostvariti sljedeće rezultate:**
 - rastaviti upisanu tekstualnu poruku na niz ASCII znakova
 - izvršiti istovremenu digitalnu modulaciju tri neovisne poruke pomoću 3 neovisna ortogonalna koda dužine minimalno 20 bitova CDMA metodom
 - digitalnu demodulaciju poruka
 - rekonstrukciju sve tri poruke iz ASCII formata
- **Razviti projektnu dokumentaciju:**
 - opisati problem
 - opisati metodologiju rada i koncept rješenja
 - opisati i demonstrirati razvijenu programsku podršku

Goldov kod

- **ortogonalan kod** -> unakrsna korelacija = 0
- vrlo lako razlikovanje dvije kodne sekvence u okruženjima s velikom bukom ili interferencijom
- **LSFR** (Linear shift feedback register)
- Odabir preferiranih parova

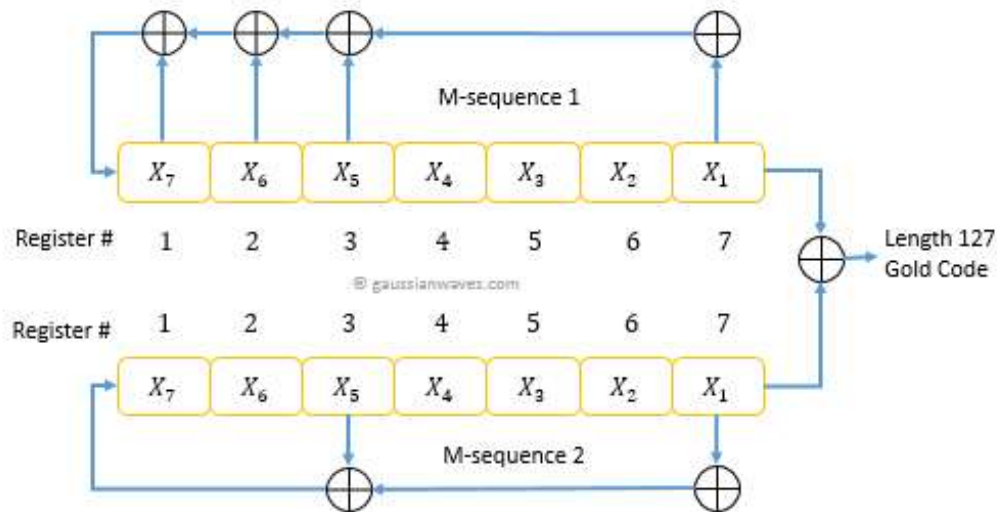


Figure 2: Length 127 – Gold sequence generator with odd n for the preferred pairs – $[7,3,2,1]$, $[7,3]$

Length of LFSR	Length of Gold code	Preferred pairs of m-sequences		Three valued Cross-correlation (unnormalized)		
n	$N = 2^n - 1$	m-sequence 1	m-sequence 2			
5	31	[5,4,3,2]	[5,2]	-9	-1	7
6	63	[6,5,2,1]	[6,1]	-17	-1	15
7	127	[7,3,2,1]	[7,3]	-17	-1	15
		[7,3,1]	[7,1]	-17	-1	15
		[7,5,4,3,2,1]	[7,3,2,1]	-17	-1	15
9	511	[9,6,4,3]	[9,4]	-33	-1	31
		[9,8,4,1]	[9,6,4,3]	-33	-1	31
10	1023	[10,9,7,6,4,1]	[10,9,8,7,6,5,4,3]	-65	-1	63
		[10,7,6,4,2,1]	[10,8,5,1]	-65	-1	63
		[10,9,7,6,4,1]	[10,8,7,6,5,4,3,1]	-65	-1	63
11	2047	[11,8,5,2]	[11,2]	-65	-1	63

Table 1: Preferred pairs of m-sequences

Goldov kod - implementacija

```
generate_gold_code <- function(seed){  
  # Generates codes up to 127 bits  
  g1 <- c(1, 1, 1, 0, 0, 0, 1)  
  g2 <- c(0, 0, 1, 0, 0, 0, 1)  
  last <- length(g1)  
  
  # Bit number 1 is left, bit number 7 is right  
  r1 <- c(0, 0, 0, 0, 0, 0, 1)  
  r2 <- seed  
  
  gold_code <- c()  
  
  for(i in 1:20) {  
    # Compute XOR of last bits in shift registers  
    output_bit <- int_xor(r1[last], r2[last])  
  
    in_r1 = NA  
    in_r2 = NA
```

```
    # Calculate new input bits via preferred pairs  
    for(j in seq(last, 1, -1)) {  
      if(g1[j] == 1) {  
        if(is.na(in_r1)) {  
          in_r1 <- r1[j]  
        } else {  
          in_r1 <- int_xor(in_r1, r1[j])  
        }  
      }  
  
      if(g2[j] == 1) {  
        if(is.na(in_r2)) {  
          in_r2 <- r2[j]  
        } else {  
          in_r2 <- int_xor(in_r2, r2[j])  
        }  
      }  
    }  
  }
```

```
    # Shift registers  
    r1 <- c(in_r1, r1[1:last-1])  
    r2 <- c(in_r2, r2[1:last-1])  
  
    # Store new output  
    gold_code <- c(gold_code, output_bit)  
  }  
  
  return (gold_code)  
}
```

Generacija jedne
kodne sekvence

```
# Generates n sequences depending on the number  
# of seeds given in argument  
generate_n_codes <- function(seeds) {  
  codes <- list()  
  for(i in 1:length(seeds)) {  
    # Turn int to binary array  
    seed_bin <- integer(7);  
    seed_bin[1:7] <- as.integer(intToBits(seeds[i]))[1:7]  
  
    codes[[i]] <- generate_gold_code(seed_bin)  
  }  
  
  return (codes)  
}
```



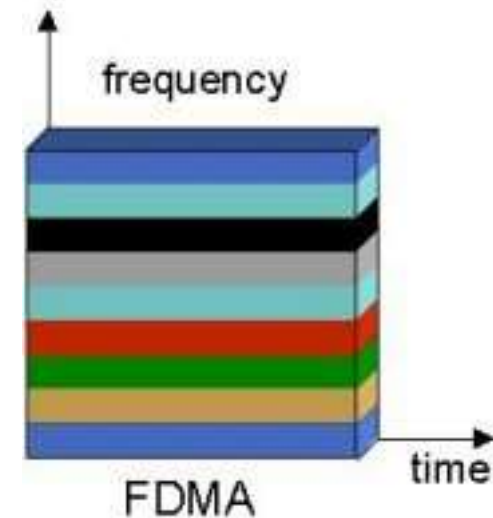
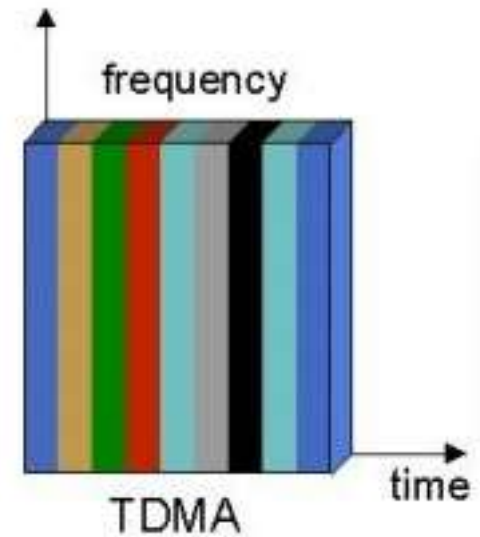
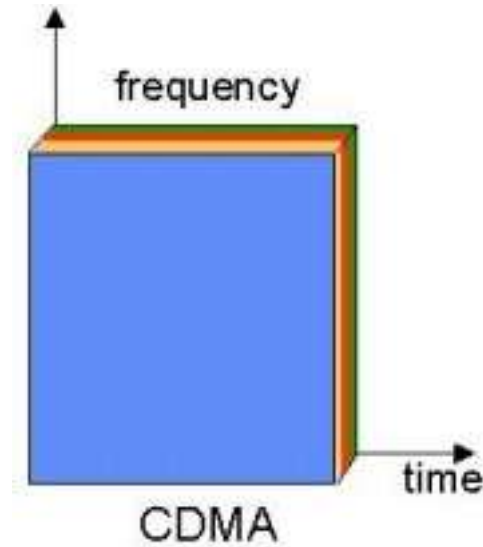
```
## [[1]]  
## [1] 1 0 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1  
##  
## [[2]]  
## [1] 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1  
##  
## [[3]]  
## [1] 1 0 1 1 0 1 1 1 0 0 1 0 0 0 0 1 0 0 1 0
```

CDMA

- Tehnologija za dijeljenje komunikacijskog kanala
- TDMA i FDMA vs CDMA
- Istovremeni prijenos različitih poruka koristeći kodove
- Digitalna modulacija signala
- Veći kapacitet mreže i sigurnost
- bežični telefonski sustavi, širokopojasne bežične pristupne mreže, satelitska navigacija

CDMA algoritam

- **Koraci CDMA algoritma:**
 - Generiranje ortogonalnih kodova
 - Učitavanje poruka
 - Pretvorba poruka u binarni oblik
 - Kodiranje poruka
 - Računanje kompozitnog signala
 - Dekodiranje kompozitnog signala
 - Pretvorba natrag u ASCII



CDMA - implementacija

```
ascii_to_binary = function(msg) {  
  # Konvertiranje ASCII vrijednosti u binarne vrijednosti  
  binary = sapply(msg, function(x) {paste0(rev(as.integer(intToBits(x)[1:8])), collapse = "")})  
  return(binary)  
}  
  
# Pretvaranje podataka iz ASCII u binarni  
  
stream <- list()  
k = 1  
for(d in data){  
  stream[[k]] <- ascii_to_binary(d)  
  k = k + 1  
}
```

```
# Funkcija za modulaciju poruka  
code_message = function(stream, code) {  
  coded_msg = c()  
  code_neg = ifelse(code == 0, 1, ifelse(code == 1, 0, code))  
  for(i in 1:length(stream)) {  
    symbol = strsplit(stream[i], "")[[1]]  
    for(j in 1:length(symbol)) {  
      if(symbol[j] == "1"){  
        coded_msg = c(coded_msg, code)  
      } else {  
        coded_msg = c(coded_msg, code_neg)  
      }  
    }  
  }  
  
  coded_msg = ifelse(coded_msg == 0, -1, 1)  
  return(coded_msg)  
}
```



```
## [1] -1 1 -1 -1 1 1 -1 -1 -1 1 -1 -1 -1 1 -1 -1 1 -1 1 -1 1 -1 1  
## [26] -1 1 1 1 -1
```

CDMA - implementacija

```
# Funkcija za kompoziciju signala
composite_signal = function(coded) {
  composite = 0
  for(c in coded){
    composite = composite + c
  }
  return(composite)
}
```



```
## [1] -3 3 -3 -3 3 -1 -1 -3 1 1 -3 1 -1 1 1 -1 3 1 1 -1 3 -3 3 3 -3
## [26] 1 1 3 -1 -1
```

```
# Funkcija za demodulaciju
decode = function(code, composite) {
  message = c()
  for(i in seq(1, length(composite), len)) {
    code_vect = composite[i:(i+len-1)]
    code_vect = code_vect * code
    code_sum = sum(code_vect)

    message = c(message, ifelse(code_sum < 0, 0, 1))
  }

  return(message)
}
```


CDMA - implementacija

```
# Funkcija za pretvorbu binarnog koda u poruke
convert_to_string = function(message) {
  ascii_str = ""
  for(i in seq(1, length(message), 8)) {
    binary_str = message[i:(i+7)]
    binary_str = paste(binary_str, collapse="")
    decimal = strtoi(binary_str, base=2)

    ascii_str = paste(ascii_str, intToUtf8(decimal), sep="")
  }

  return(ascii_str)
}
```

```
# Prikaz demoduliranih poruka
for(m in messages){
  print(convert_to_string(m))
}

#Primjer dekodiranja poruke s "krivim" kodom
print(convert_to_string(fakeMessage))
```

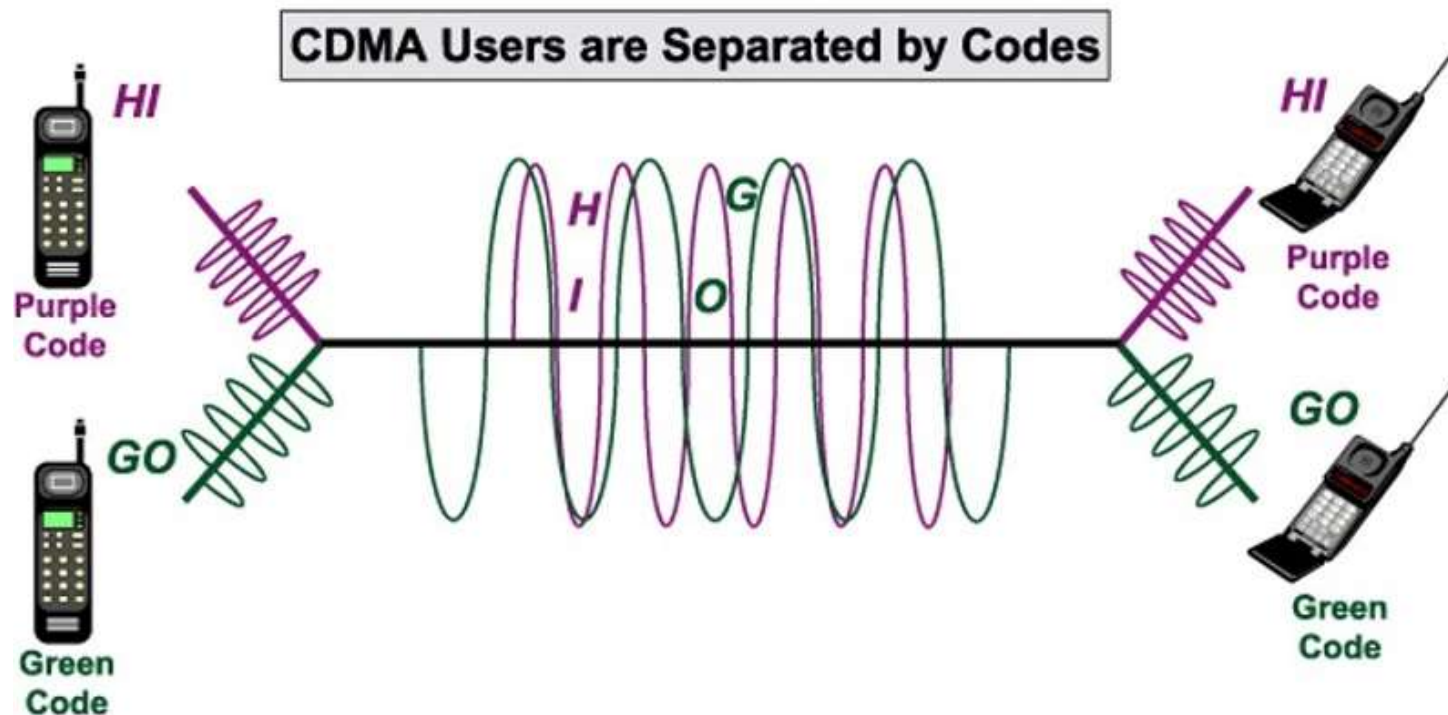


```
## [1] "Never gonna give you up; Never gonna let you down; Never gonna run around and desert you"
## [1] "Az, v ime Otca i Sina i Svetago Duha. Az opat Drziha pisah se o ledine, juze da Zvanimir kralj hrvatskij v dni svoje v
Svetuju Luciju i svedomi: zupan Desimira Krbave, Mratin v Lice, Pribineza posal Vinodole, Jakov v otoce."
## [1] "Some random text, doesn't really matter."

## [1] "Coler aodnm gape qmoaqh#pfetafladmha$da~ i`a Drj) @ par ` bd dd abe$ ` `d ` Rd`d iip"
```

Zaključak

- CDMA metoda i njene prednosti
- Generiranje ortogonalnih kodova
- Gold kod
- Demonstracija CDMA metode



Izvori

- Mathuranathan Viswanathan (2015.) - Gold code generator using LFSRs. <https://shorturl.at/bpGKY>
- Jean-Paul M.G. Linnartz, Frank Kamperman (1995.) - Code Sequences for Direct Sequence CDMA. <https://shorturl.at/cjG57>
- David M. Harris (2014.) - E11 Lecture 7: Gold Codes. <https://shorturl.at/ntFT3>
- Wikipedia - Code-division multiple access. <https://shorturl.at/grU78>
- Renato Filjar (2023.) – Programski određen radio V9, V10 CDMA