



UNIVERSITÀ DI PISA

Dipartimento di Ingegneria Dell'Informazione

CDMA RECEIVER

Progetto di Electronics and Communications Systems

Amedeo POCHIERO

A.Y. 2019-2020

Indice

1	Introduzione	1
1.1	Descrizione del Problema	1
1.2	CDMA	1
1.3	Applicazioni	2
1.4	Esempio utilizzato	2
2	Architettura	3
2.1	Diagramma a Blocchi	3
2.2	Implementazione	3
2.3	Test-plan	3
3	Sintesi	3
3.1	Utilizzo	3
3.2	Massima frequenza di utilizzo	3
3.3	Cammino Critico	3
3.4	Consumo di Potenza	3
3.5	Warnings	3
4	Conclusioni	3

1 Introduzione

1.1 Descrizione del Problema

Il collegamento broadcast può avere più nodi trasmettenti e riceventi connessi allo stesso canale broadcast condiviso. In uno scenario del genere si pone il problema di come coordinare l'accesso di più nodi trasmettenti e riceventi in un canale broadcast condiviso, ossia il problema dell'accesso multiplo. Dato che tutti i nodi sono in grado di trasmettere frame, è possibile che due o più lo facciano nello stesso istante, per cui tutti i nodi riceveranno contemporaneamente più frame. Tra questi si genera una collisione a causa della quale nessuno dei nodi riceventi riuscirà a interpretare i frame.

1.2 CDMA

Il protocollo **CDMA** (*code division multiple access*) è un protocollo a suddivisione del canale in cui i vari utenti possono trasmettere contemporaneamente, causando quindi collisioni e interferenze tra loro, tuttavia il ricevente è in grado comunque di ricostruire il segnale trasmesso. A tale scopo, ogni utente modula il proprio segnale di periodo T_b (*symbol period*) con un codice, unico per ogni utente, di periodo T_c (*chip period*) dove $T_c \ll T_b$ come si può vedere in Figura 1.

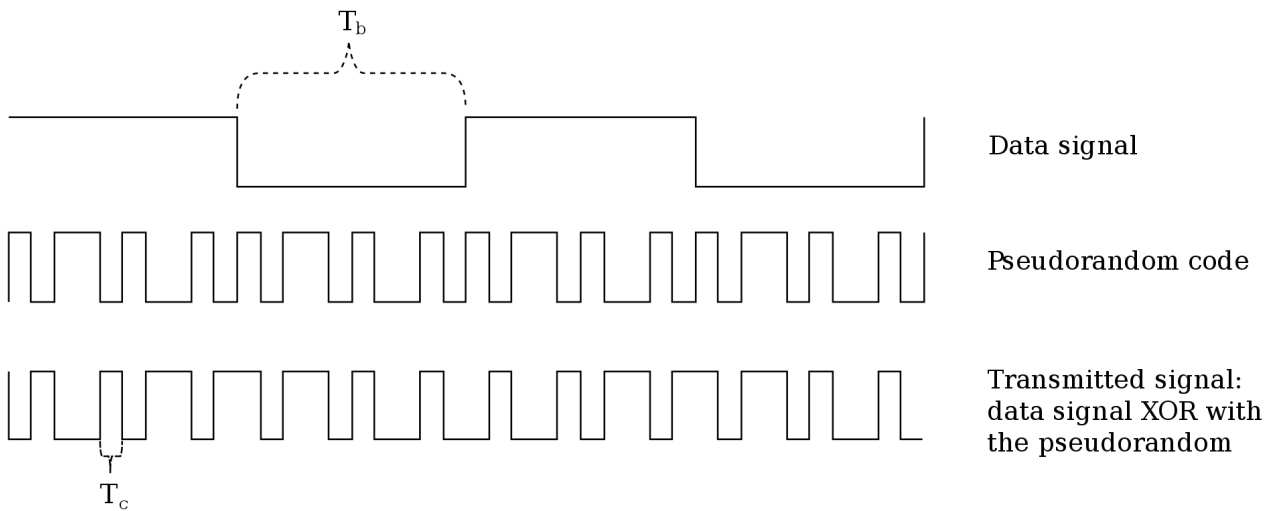


Figura 1: Generazione del segnale CDMA trasmesso

Il rapporto tra i due periodi è definito come *Spreading Factor* e come requisito è stato posto a 16:

$$\frac{T_b}{T_c} = 16$$

I codici devono essere scelti in modo tale che la correlazione tra i vari segnali dei diversi utenti sia il più vicino possibile a zero. Nel synchronous CDMA si può sfruttare la proprietà matematica di *ortogonalità* tra vettori che rappresentano stringhe di dati. Due vettori a e b si dicono ortogonali se vale la seguente relazione:

$$a \cdot b = 0$$

Ogni utente deve usare un codice ortogonale a quello di tutti gli altri. Nella tabella di seguito viene riportato un esempio di vettori $cw_1, cw_2 \in \mathbb{Z}^{16}$ ortogonali tra di loro, i bit 0, 1 vengono rappresentati rispettivamente dai simboli $-1, 1$:

vector	bit																Prodotto Scalare
cw_1	1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	-1	1	-
cw_2	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	-1	-1	-
$cw_{1,i} * cw_{2,i}$	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	-1	-1	0

Tabella 1: Vettori Ortogonali

1.3 Applicazioni

Il **CDMA** è il protocollo di accesso a canale condiviso più diffuso nelle reti wireless e nelle tecnologie cellulari. Deriva da una tecnologia usata per implementare il **GPS** (*Global Position System*) ed è stato usato in :

- *Globastar network*, con il nome di CDMA2000, ed altre compagnie telefoniche
- *UMTS 3G* come protocollo di accesso multiplo standard
- *OmniTRACS satellite system*, per trasporti logistici.

1.4 Esempio utilizzato

Al fine di verificare il corretto funzionamento del ricevitore, è stato seguito un caso reale di ricezione di due bit in un ricevitore CDMA. Facendo riferimento ai codici (*code words*) della tabella 1, si è considerato uno scenario in cui il primo utente trasmette il simbolo 1, mentre un secondo utente trasmette il simbolo -1 . Per le proprietà fisiche dell'interferenza, se 2 segnali interferenti sono in fase, essi si sommano e si crea un segnale di ampiezza doppia, altrimenti si sottraggono e creano un segnale con un'ampiezza pari alla differenza delle ampiezze. Nel mondo digitale, questo comportamento può essere rappresentato dalla somma componente per componente dei vettori trasmessi.

Nella seguente tabella 2 il simbolo trasmesso è modulato con la rispettiva *code word*, ottenendo un *chip stream* per ogni utente. Ogni chip stream trasmesso interferisce con gli altri trasmessi nello stesso istante, formando l'*interference pattern*, cioè il segnale realmente ricevuto dai ricevitori.

data	value															
d_1	1															
d_2	-1															
$cw_{1,i} * d_1$	1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	-1	1
$cw_{2,i} * d_2$	-1	1	-1	1	-1	1	1	1	-1	-1	1	1	1	-1	1	1
<i>interference pattern</i>	0	2	-2	2	-2	0	0	2	0	-2	0	0	2	0	0	2

Tabella 2: Interferenza tra trasmettitori

Per ricostruire il segnale, moltiplica ogni componente dell' *interference pattern* con la propria *code word*. Il vettore r ottenuto è dato in input ad un *Decisore Hard A Soglia* il quale decide il bit da porre in uscita. La decisione è presa nel seguente modo:

- se $\sum_{i=1}^{16} r_i \geq 0$ allora *bitstream* = 1
- se $\sum_{i=1}^{16} r_i < 0$ allora *bitstream* = 0

Nell'esempio considerato i ricevitori effettueranno le seguenti operazioni:

data	value															Somma	Bit deciso	
$r_i * cw_{1,i}$	0	2	2	2	2	0	0	2	0	2	0	0	2	0	0	2	16	1
$r_i * cw_{2,i}$	0	-2	-2	-2	-2	0	0	-2	0	-2	0	0	-2	0	0	-2	-16	0

Come si nota dalla tabella, i ricevitori sono in grado di ricostruire il bit trasmesso originariamente (primo utente 1, secondo utente 0) come specificato in 1.4 .

2 Architettura

2.1 Diagramma a Blocchi

2.2 Implementazione

2.3 Test-plan

3 Sintesi

3.1 Utilizzo

3.2 Massima frequenza di utilizzo

3.3 Cammino Critico

3.4 Consumo di Potenza

3.5 Warnings

4 Conclusioni