UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

Tesina di Laurea Triennale in Ingegneria dell'Automazione

CIRCUITI LINEARI SEQUENZIALI E LORO APPLICAZIONI PER L'ELEBORAZIONE DI CODICI

Laureanda: Chiara Anselmi Relatore: Professore Mauro Bisiacco

23 febbraio 2012 Anno Accademico 2011/2012

Ringraziamenti

La strada che mi ha portato a questo momento non è stata facile e lo sarebbe stato ancor meno senza il supporto di tutti coloro che mi sono stati vicini in questi anni.

Un sentito ringraziamento va al Professor M. Bisiacco per la disponibilità mostrata in ogni occasione, in particolar modo nel periodo di stesura della tesi e nei mesi in cui è stato mio docente per il corso di Fondamenti di Automatica: grazie per tutto quello che mi ha insegnato e per aver contribuito a farmi amare questa disciplina.

Grazie ai miei genitori, il cui incoraggiamento e affetto sono stati preziosi e non solo nei momenti più difficili, ma in ogni circostanza.

Grazie agli amici che mi hanno sostenuto e sopportato, con e mie lune storte e i miei numerosi "Scusa ma oggi non posso".

Infine, grazie a mia sorella: 1200 o 8000 chilometri di distanza sono solo un fastidio, non un impedimento ad esserci, sempre.

Introduzione

In questi ultimi anni l'elettronica digitale, le telecomunicazioni e le altre discipline ad esse connesse hanno fatto enormi progressi, portando ad innovazioni tecnologiche di grande rilevanza. Tuttavia le fondamenta su cui si basano le teorie e le scoperte più recenti sono da ricercarsi negli studi svolti cinquanta-sessant'anni fa da ingegneri e matematici, quali C. Shannon e R. W. Hamming per citarne solo due, senza il cui contributo non si sarebbe arrivati ai traguardi raggiunti fin'ora.

Le tecniche sviluppate in quei decenni sono tutt'ora valide e vengono ancora oggi utilizzate in molte applicazioni nei campi più svariati, come la crittografia, la codifica e decodifica di segnali, le computazioni digitali e perfino la progettazione di strumentazione radar.

In questo contesto è stata sviluppata la teoria dei Circuiti Sequenziali Lineari, che si pone come punto di unione tra la Teoria dei Sistemi e lo studio degli automi, ovvero i circuiti sequenziali. Il fatto di poter sfruttare le metodologie proprie di entrambe le discipline, porta notevoli vantaggi, garantendo un approccio poliedrico alle problematiche da affrontare e un'ampia gamma di possibili applicazioni per questi circuiti.

I Circuiti Sequenziali Lineari, infatti, sono stati e sono tuttora impiegati nella realizzazione di numerosi dispositivi. Data la grande importanza assunta dai sistemi di codifica e decodifica, indispensabili per il funzionamento di qualsivoglia sistema di comunicazione, si è cercato in questa tesi di approfondire proprio questo aspetto, andando ad analizzare come i Circuiti Sequenziali Lineari possano essere utilizzati per progettare encoders e decoders. In un mondo che è sempre più dipendente dalle telecomunicazioni, diventa indispensabile avere a disposizione dei sistemi capaci di codificare, inviare, ricevere e decodificare segnali, individuando eventuali errori e possibilmente correggendoli. I Circuiti Sequenziali Lineari si pongono come una possibile soluzione a questi problemi.

Nelle pagine che seguono si è preferito, prima di descrivere le applicazioni sui sistemi di telecomunicazione, proporre un'introduzione matematica, per richiamare i concetti fondamentali e per delineare le caratteristiche principali dei Circuiti Sequenziali Lineari. Questo è proprio lo scopo del primo capitolo, dove vengono fornite alcune nozioni di algebra lineare, aritmetica modulare e teoria dei campi finiti, oltre alla presentazione generale dei Circuiti Sequenziali Lineari.
Nel secondo capitolo vengono presentate invece le prime applicazioni per la realizzazione di circuiti per operazioni polinomiali in campo digitale, indispensabili per la realizzazione dei circuiti progettati nel terzo capitolo.

L'ultimo capitolo rappresenta il fulcro di tutta la tesi. Sono descritti tre diversi tipi di encoders, ciascuno con le sue peculiarità, e due diversi decoders, facendo particolare attenzione al problema

dell'individuazione e correzione di errori in trasmissione. Per concludere, attraverso degli esempi numerici, si sono messe a confronto le varie tipologie di dispositivi, onde confrontarne le prestazioni.

1. Introduzione matematica

1.1. Richiami di algebra lineare e aritmetica modulare

Premessa. Considerati i continui riferimenti che saranno effettuati nei capitoli successivi a concetti di algebra lineare e aritmetica modulare, si è preferito introdurre brevemente tali argomenti in questo paragrafo introduttivo, che ha come unico scopo quello di richiamare i concetti base, non di esporre la materia in maniera completa e dettagliata. Per una trattazione esaustiva si rimanda ai testi di riferimento in bibliografia.

Definizione di gruppo. Un insieme G costituisce un gruppo (group) se è definita per ogni coppia di elementi in G l'operazione * ; devono inoltre valere i seguenti postulati:

1. Chiusura: per tutti gli elementi a e b appartenenti a G, $a*b$ è ancora in G
2. Proprietà associativa: per ogni a, b, c in G allora $a*(b*c) = (a*b)*c$
3. Esistenza dell'elemento neutro: G contiene un unico elemento e tale che per ogni a in G, $e*a = a*e = a$
4. Esistenza dell'inverso: per ogni a in G esiste ed è unico l'elemento a^{-1} tale che $a*a^{-1} = a^{-1}*a = e$

Se in G vale anche la proprietà commutativa, ovvero se per ogni a, b in G allora $a*b = b*a$, allora il gruppo viene detto gruppo Commutativo o Abeliano.
Un gruppo è detto gruppo additivo se l'operazione in esso definita è l'addizione e si indica $a+b$, gruppo moltiplicativo se l'operazione è la moltiplicazione e si indica con ab.
Un sottoinsieme C di G è un sottogruppo di G se soddisfa tutti i postulati del gruppo rispetto all'operazione *.

Definizione di anello. L'insieme R è un anello (ring) se in esso sono definite, per ogni coppia di elementi, l'addizione e la moltiplicazione; deve inoltre soddisfare i seguenti postulati:

1. R è gruppo Abeliano additivo
2. Chiusura per la moltiplicazione
3. Proprietà associativa per la moltiplicazione
4. Proprietà distributiva: per ogni coppia di elementi a, b e c in R, allora:
   $a(b+c) = ab + ac$
   $(b+c)a = ba + ca$

Se vale anche la proprietà commutativa per la moltiplicazione, allora l'anello è un anello commutativo.

Definizione di ideale. Un sottoinsieme I dell'anello commutativo R è detto ideale (ideal) se per ogni *a* in R e *b* in I, *ab* è in R. Un ideale formato da tutti i multipli di un elemento *a* dell'anello è detto ideale principale ed è indicato con (*a*). L'elemento *a* viene chiamato generatore dell'ideale; un anello in cui ogni ideale è ideale principale viene definito anello ad ideali principali.

Definizione di campo. Un anello commutativo F è detto campo (field) se soddisfa i seguenti due postulati:

1. Esistenza dell'elemento neutro della moltiplicazione;
2. Esistenza dell'inverso per la moltiplicazione: per ogni elemento *a* non nullo in F c'è un elemento a^{-1} in F tale che $aa^{-1} = a^{-1}a = 1$.

Pertanto, F è un gruppo Abeliano per l'addizione e un gruppo Abeliano per la moltiplicazione se si esclude lo {0}, elemento neutro dell'addizione.

Un esempio, che sarà molto utile nel prosieguo della trattazione è quello degli anelli commutativi, costituiti da un numero finito di interi {0, 1, …, p-1}, in cui le operazioni di addizione e moltiplicazione sono definite modulo p. Se p è un numero primo, questo anello è anche un campo ed si indica con GF(p) (Campo di Galois).

Prima di procedere con la descrizione delle proprietà e caratteristiche di questi particolari campi, si preferisce proporre un breve excursus sull'aritmetica modulare, al fine di chiarire alcuni concetti fondamentali.

Fondamenti di aritmetica modulare. Siano *a*, n appartenenti a Z due interi e siano *q* appartenente a Z e *r* appartenente all'insieme {0, 1,…, n-1} gli unici interi per cui valga la seguente:

$a = nq + r$

dove *r* è il resto della divisione di *a* per n e si indica con *a* mod n.

Due interi *a* e *b* sono equivalenti modulo n e si indica $a \equiv b \pmod{n}$, se *a* mod n = *b* mod n, ovvero se esiste k tale che $(a - b) = nk$; detto in altri termini sono equivalenti modulo n se hanno lo stesso resto nella divisione tramite n.

L'operazione di modulo induce una partizione dell'insieme Z in classi di equivalenza, dove una classe ha per elementi tutti gli interi che divisi per n restituiscono il medesimo resto. Le classi di equivalenza sono n, una per ogni possibile resto della divisione per n.

Per operare l'addizione modulo n si può procede "per rappresentanti", ovvero sostituendo gli interi che si vogliono sommare con l'elemento più piccolo della classe corrispondente, che funge da rappresentante della classe; fatto ciò si sommano i rappresentanti e il risultato, se non è esso stesso un rappresentante di una classe, va sostituito con il rappresentante corrispondente. Alternativamente, si possono sommare gli interi e sostituire il risultato con il suo rappresentante.

Ad esempio, si supponga di operare nell'insieme GF(3) = {0, 1, 2}, che è l'insieme degli interi modulo 3:

(4 + 10) mod 3 = (14) mod 3 = 2

Infatti:

(4) mod 3 = 1

(10) mod 3 = 1

La somma modulo n possiede tutte le proprietà della somma di interi "tradizionale".

Allo stesso modo si procede per la moltiplicazione, riportando volta per volta i fattori, o direttamente il prodotto, ai corrispettivi rappresentanti.

Considerando sempre l'insieme GF(3) = {0, 1, 2} si ha per esempio:

(5 * 12) mod 3 = (60) mod 3 = 0

Infatti:

(5) mod 3 = 2

(12) mod 3 = 0

Proprietà dei campi di Galois. Si tornino ora a considerare le proprietà dei campi di Galois, ovvero come è stato detto prima, dei campi finiti con n elementi.

Si consideri quindi il campo GF(p), con p primo. L'insieme di polinomi a coefficienti in GF(p) viene indicato con GF(p)[x]. La domanda che sorge spontanea a questo punto è se è possibile applicare quanto detto per gli interi ai polinomi. La risposta è affermativa. In effetti, si può estendere la teoria sviluppata per gli interi modulo un intero, definendo il concetto di polinomio modulo un polinomio.

Sia quindi GF(p) un campo e A(x) e B(x) polinomi a coefficienti in GF(p); allora esistono e sono unici i polinomi Q(x) e R(x) tali che A(x) = B(x) Q(x) + R(x).

Come per i numeri interi, R(x) è il resto della divisione e si indica con R(x) = A(x) mod B(x). Inoltre due polinomi sono equivalenti modulo il polinomio Ψ(x) se danno lo stesso resto se divisi per Ψ(x). Anche in questo caso l'operazione di modulo è una relazione di equivalenza e divide l'insieme dei polinomi in classi di equivalenza. Ogni classe di equivalenza è descritta da un

rappresentante, dato dal resto della divisione per $\Psi(x)$: R(x) ha grado massimo n – 1, se n è il grado di $\Psi(x)$.

Viene in genere indicato con [GF(p)[x] mod $\Psi(x)$] l'anello di tutti i polinomi in x a coefficienti in GF(p) con addizione e moltiplicazione definite modulo $\Psi(x)$. Se $\Psi(x)$ ha grado n allora [GF(p)[x] mod $\Psi(x)$] è costituito dall'insieme di tutti i polinomi di grado massimo n – 1 del tipo:

$a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$ per ogni $a_0, \ldots, a_{n-1} \in GF(p)$.

Si sottolinea come questi polinomi siano proprio i rappresentanti delle classi di equivalenza sopra menzionate.

Se $\Psi(x)$ è un polinomio irriducibile, allora l'anello [GF(p)[x] mod $\Psi(x)$] è un campo e si indica con $GF(p^n, \Psi(x))$. $GF(p^n, \Psi(x))$ è quindi il campo di Galois associato al numero primo p e al polinomio irriducibile $\Psi(x)$, a coefficienti in GF(p) e grado n.

Un caso particolare si ha per n = 1. Si è visto che il campo $GF(p^n, \Psi(x))$ è costituito da tutti i polinomi di grado massimo n – 1, che nella situazione sotto esame sono i polinomi di grado 0, ovvero le costanti. Ma le costanti altro non sono che gli elementi di GF(p). Quindi, per n = 1, gli elementi di $GF(p^n, \Psi(x))$ vanno a coincidere con quelli di GF(p), perdendo la dipendenza dal polinomio $\Psi(x)$.

Si considerino i seguenti esempi per chiarire meglio quanto detto fin'ora.

Sia GF(2) il campo per gli interi e sia $\Psi(x) = x^3 + x + 1$.

Gli elementi delle classi di equivalenza di polinomi a coefficienti in GF(2) modulo il polinomio $\Psi(x)$ sono i polinomi a grado massimo 2 (n-1 con n = 3):

0	x+1	$x^2 + x$
1	x^2	$x^2 + x + 1$
x	x^2+1	

Questi sono ovviamente i possibili resti della divisione per $\Psi(x)$.

Sia ora invece GF(3) il campo di interi e sia $\Psi(x) = 1 + x + 2x^2$. I possibili resti della divisione per $\Psi(x)$ e quindi elementi del campo $GF(3^2, 1 + x + 2x^2)$ sono:

0	1	x
1 + x	1 + 2x	2
2x	2 + 2x	2 + x

Si può osservare che ciascuno dei polinomi non nulli sopra elencati coincide con una potenza x^i di x, per $0 \leq i \leq 7$. Si dice allora che x è elemento primitivo e che i nove polinomi elencati coprono

tutto il campo GF(3^2, $1 + x + 2x^2$). In effetti quelli riportati sono tutti i possibili polinomi di grado massimo n – 1. I casi $x^0 = 1$ e $x^1 = x$ sono immediati; per i = 2 si nota che:

$x^2 = 2\ (2x^2 + x + 1) + (x + 1)$

Questa espressione si ricava dalla divisione di x^2 per il polinomio $2x^2 + x + 1$, dove 2 è il quoziente e (x + 1) è il resto. Ricordando quanto detto precedentemente sulle classi di equivalenza, si ricava la relazione:

$x^2 = x + 1$

Per x^3 si pone la seguente:

$x^3 = x\ (x^2) = x\ (x + 1) = x^2 + x = x + 1 + x = 2x + 1$

Le altre uguaglianze si ricavano allo stesso modo, per cui si ha:

$x^4 = x\ (x^3) = x\ (2x + 1) = 2\ x^2 + x = 2\ (x + 1) + x = 2x + 2 + x = 3x + 2 = 2$

$x^5 = x\ (x^4) = x\ (2) = 2x$

$x^6 = x\ (x^5) = x\ (2x) = 2\ x^2 = 2\ (x + 1) = 2x + 2$

$x^7 = x\ (x^6) = x\ (2x + 2) = 2x^2 + 2x = 2x + 2 + 2x = 2 + 4x = x + 2$

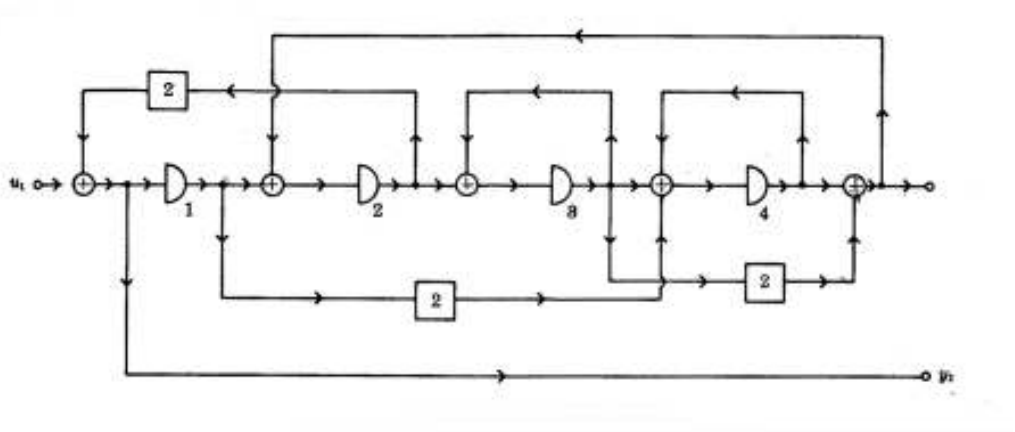
Nello sviluppare le catene di uguaglianze si è tenuto conto del fatto che le operazioni di addizione e moltiplicazioni tra polinomi si impostano esattamente come nel caso degli interi, riconducendosi alla fine sempre ai rappresentanti delle classi di equivalenza.

Prima di concludere si vuole dare un'ultima definizione, relativa alla funzione minima in un campo di Galois GF(p^n, $\Psi(x)$). Sia d un elemento di GF(p^n, $\Psi(x)$); la funzione minima di d è l'unico polinomio monico di grado minimo $m_d(x)$ tale che $m_d(d) = 0$; $m_d(x)$ è irriducibile e di grado massimo n ed ovviamente dipende in generale dalla scelta fatta per d.

1.2. I Circuiti Sequenziali Lineari

Prima di iniziare la descrizione dei Circuiti Sequenziali Lineari, si vuole sottolineare come questo paragrafo non sia inteso come una trattazione approfondita su tali circuiti, ma come un'introduzione necessaria per fornire le nozioni indispensabili alla comprensione degli argomenti che saranno presentati nei prossimi capitoli. Si rimanda al libro "Linear sequential circuits. Analysis, synthesis, and applications" di Arthur Gill (vedi bibliografia) per approfondimenti sui Circuiti Sequenziali Lineari e ai testi "Teoria dei sistemi dinamici" e "Appunti di Teoria dei Sistemi", rispettivamente di M. Bisiacco S. Braghetto e E. Fornasini G. Marchesini, per quanto riguarda Teoria dei Sistemi.

I Circuiti Sequenziali Lineari (Linear Sequential Circuits o Machines, in breve LSC) sono circuiti costituiti da un numero finito di terminali di ingresso e di uscita, per l'applicazione e l'osservazione di segnali digitali appartenenti ad un campo finito GF(p) per qualche p fissato, e da un numero finito di blocchi elementari (ritardatori, moltiplicatori per una costante, nodi sommatori)
Un esempio di LSC è riportato in figura



Il numero di blocchi ritardatori presenti nel circuito costituisce la dimensione dell'LSC e l'uscita a tali blocchi rappresenta lo stato del sistema. In quanto circuiti lineari, gli LSC possono essere descritti sfruttando la teoria e le metodologie di analisi proprie della Teoria dei Sistemi. Le relazioni che intercorrono tra ingresso e uscita sono pertanto le note espressioni matriciali:

$$x(t+1) = A\,x(t) + B\,u(t) \qquad (1.0)$$

$$y(t) = C\,x(t) + D\,u(t)$$

dove

$A = [a_{ij}]_{nxn}$ $\quad C = [c_{ij}]_{mxn}$

$B = [b_{ij}]_{nxp}$ $\quad D = [d_{ij}]_{mxp}$

La matrice A viene detta matrice caratteristica dell'LSC.

Facendo riferimento alla figura, tenendo presente che si sta operando su GF(3), le matrici che descrivono quel particolare sistema sono:

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Oltre alle equazioni (1.0), che esprimono genericamente il legame ingresso-uscita, si possono facilmente ricavare le equazioni che descrivono l'evoluzione nel tempo dello stato e dell'uscita, particolarmente utili per la descrizione del sistema, ovvero le:

$$x(t) = A^t x(0) + \sum_{i=0}^{t-1} A^{t-1-i} Bu(i) = x_l(t) + x_f(t) \tag{1.1}$$

$$y(t) = CA^t x(0) + \sum_{i=0}^{t-1} CA^{t-1-i} Gu(i) + Du(t) = y_l(t) + y_f(t) \tag{1.2}$$

dove con x_l e x_f si indica rispettivamente l'evoluzione libera e forzata dello stato e con y_l e y_f l'analoga evoluzione per l'uscita.

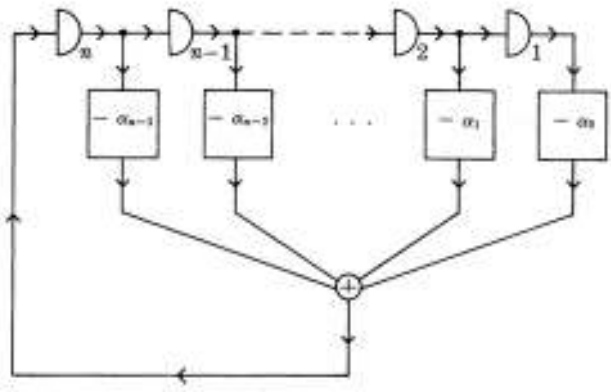
Infine, come per i sistemi lineari, anche per gli LSC è possibile ricavare l'espressione della matrice di trasferimento, data dalla nota equazione:

$$W = C(zI-A)^{-1}B + D \tag{1.3}$$

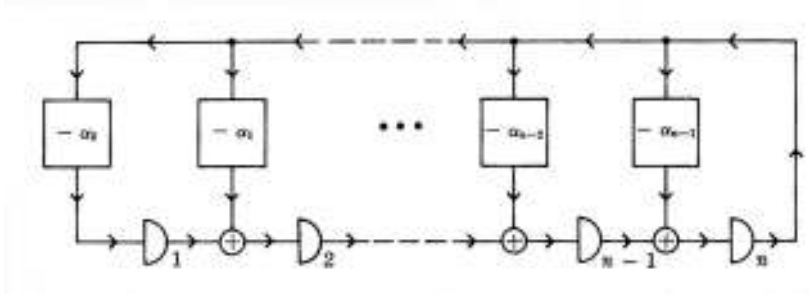
Un caso particolare, che tornerà utile nelle applicazioni che verranno studiate nei prossimi capitoli, è dato dai circuiti di shift register, ovvero LSC la cui matrice caratteristica A è in forma compagna. Si ricorda che una matrice è in forma compagna se è del tipo:

$$M = \begin{bmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ -a_0 & -a_1 & \cdots & \cdots & -a_{n-1} \end{bmatrix} \tag{1.4}$$

dove a_0, a_1..a_{n-1} sono i coefficienti del polinomio caratteristico della matrice. Se la matrice caratteristica è la trasposta di una matrice compagna, allora il circuito è detto multi adder shift register. In figura è riportato un esempio rispettivamente di un shift register e di un multi adder shift register.



Shift register



Multi adder shift register

Dalle equazioni (1.1) e (1.2) si è visto come la dinamica di stato ed uscita dipenda tanto dallo stato iniziale quanto dall'ingresso applicato. Un LSC, in cui lo stato iniziale sia nullo e che quindi evolva solo in evoluzione forzata sia per lo stato che per l'uscita, è detto QLSC (Quiescent LSC). Un LSC in cui l'ingresso sia nullo e che evolva solo in evoluzione libera è detto ALSC (Autonomus LSC).

In quel che resta di questo capitolo introduttivo sugli LSC, verranno presentate alcune proprietà tipiche degli ALSC e verranno introdotti alcuni elementi che saranno necessari alla descrizione dei circuiti per i sistemi di codifica e decodifica.

Si consideri innanzitutto un ALSC la cui matrice caratteristica A sia in forma compagna. Ad essa è associato un polinomio, detto polinomio di feedback, che ha per coefficienti i parametri dei blocchi di scalamento dello shift register corrispondente alla matrice A; il polinomio di feedback è esprimibile nella seguente forma:

$$\varphi(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + x^n \qquad (1.5)$$

Se $\varphi(x)$ è non divisibile per x, allora l'ALSC viene definito non singolare, mentre se $\varphi(x)$ è divisibile per x allora l'ALSC è definito singolare.

Se all'ALSC è associato il polinomio $\varphi(x)$, allora la sequenza $y(t) = y_0y_1y_2\ldots$ su GF(p), ovvero una sequenza di simboli di GF(p) definita per tempi interi non negativi, è una sequenza di output per il sistema se e solo se per ogni $t \geq 0$ vale:

$$y_{t+n} = - a_0y_t - a_1y_{t+1} - \ldots - a_{n-1}y_{t+n-1} \quad (1.6)$$

L'equazione (1.6) viene detta equazione ricorsiva e ogni sua soluzione è detta sequenza lineare ricorsiva corrispondente a $\varphi(x)$; pertanto il teorema appena enunciato afferma che una sequenza è una sequenza di output se e solo se è una soluzione dell'equazione (1.6).

Si consideri ora un polinomio g(x), di grado massimo T – n; esso genera nell'anello di polinomi a coefficienti in GF(p) modulo il polinomio $1 - x^T$, ovvero in [GF(p)[x] mod $(1 - x^T)$], un ideale che verrà indicato con $(g(x))_T$. Tale ideale è uno spazio vettoriale su GF(p), con base $\{g(x), xg(x), x^2g(x), \ldots, x^{n-1}g(x)\}$.

Sia $y(t) = y_0y_1\ldots y_{T-1}y_0\ldots$ una sequenza puramente periodica su GF(p) con periodo T; questa sequenza può essere associata ad un elemento dell'anello [GF(p)[x] mod $(1 - x^T)$], ovvero:

$$Y(x) = y_0\, x^{T-1} + y_1\, x^{T-2} + \ldots + y_{T-2}\, x + y_{T-1} \quad (1.7)$$

In questo caso si dice che Y(x) rappresenta y(t) nell'anello [GF(p)[x] mod $(1 - x^T)$].

Questa proprietà è molto utile se applicata agli ALSC. Infatti sia $\mathcal{A}$ un ALSC non singolare con polinomio di feedback $\varphi(x)$; in quanto l'ALSC è non singolare, $\varphi(x)$ non è divisibile per x, pertanto esiste un intero i tale per cui $\varphi(x)$ divide $1 - x^i$. Questo intero prende il nome di esponente di $\varphi(x)$ e sarà indicato con T. Viene definito polinomio generatore dell'ALSC, il polinomio:

$$g(x) = (1 - x^T) / \varphi(x) \quad (1.8)$$

Se un ALSC ha polinomio di feedback $\varphi(x)$ e polinomio generatore g(x), allora tutte le sue sequenze di output sono puramente periodiche di periodo T ; inoltre una sequenza è una sequenza di output se e solo se è rappresentata nell'anello [GF(p)[x] mod $(1 - x^T)$] da un elemento dell'ideale $(g(x))_T$, che è detto ideale di output e rappresenta univocamente tutte le sequenze di uscita generate dall'ALSC. L'insieme di tutte le sequenze di uscita producibili da un ALSC non singolare prende il nome di spazio ciclico.

A partire dal polinomio generatore e dal polinomio di feedback, è possibile definire la matrice generatrice dell'ALSC. Sia dunque $\varphi(x)$ di grado n e:

$$g(x) = (1 - x^T) / \varphi(x) = a_0\, x^{T-n} + a_1\, x^{T-n-1} + \ldots + a_{T-1-n}\, x + a_{T-n} \quad (1.9)$$

La matrice generatrice è la matrice G nxT seguente:

$$G = \begin{bmatrix} a_0 & a_1 & \dots & a_{T-n-1} & a_{T-n} & & & \\ & a_0 & a_1 & \dots & a_{T-n-1} & a_{T-n} & & \\ & & \dots & & & & & \\ & & & \dots & & & & \\ & & & & a_0 & \dots & a_{T-n-1} & a_{T-n} \end{bmatrix} \quad (1.10)$$

Si è visto che una base per l'ideale $(g(x))_T$ è data da $\{g(x), xg(x), x^2g(x), \dots, x^{n-1}g(x)\}$; ma allora la solita sequenza y(t) è una sequenza di output se e solo se il vettore composto dalle sue prime T-1 componenti appartiene allo spazio delle righe della matrice generatrice G.

Si è detto fin'ora che lo spazio ciclico, che è l'insieme di tutte le sequenze di uscita producibili da un ALSC non singolare, è univocamente determinato dal polinomio di feedback, dal polinomio generatore e dalla matrice generatrice. Oltre a questi elementi esiste una seconda matrice, che individua univocamente lo spazio ciclico ed è la matrice di base Z. Una matrice di base per uno spazio ciclico è ogni matrice Z per cui vale la seguente affermazione: la sequenza y(t) è una sequenza di output se e solo se il vettore dato dalle sue prime T-1 componenti appartiene allo spazio delle righe di Z; ovvero una matrice di base è ogni matrice equivalente per righe alla matrice generatrice G.

Prima di concludere questa breve trattazione, verrà sviluppato un altro argomento che sarà utile nella presentazione dei circuiti nei prossimi capitoli, ovvero come sia possibile descrivere uno spazio ciclico con r elementi distinti in un campo di Galois $GF(p^n, \Psi)$.
Siano $k_1 \dots k_r$ r elementi distinti nel campo $GF(p^n, \Psi)$, sia $m_i(x)$ la funzione minima di k_i e sia g(x) il $mcm(m_i(x))$, il tutto con $1 \leq i \leq r$. Sia inoltre T_i il grado di k_i e sia T il $mcm(T_i)$.
Ricordando la definizione di $(g(x))_T$ data precedentemente, si ha che Y(x) è elemento di $(g(x))_T$ se e solo se $Y(k_i) = 0$ per ogni $i = 1, \dots r$.
Questa affermazione si dimostra ricordando le proprietà della funzione minima e la definizione data di g(x).
$Y(k_i) = 0 \Leftrightarrow Y(x) = \alpha\, m_i(x) = \beta\, g(x)$
Ma $(g(x))_T$ è dato dagli Y(x) tali che per qualche a(x) e b(x) valga la seguente:
$Y(x) = a(x)\, g(x) + b(x)\, (1 - x^T)$
e considerando che g(x) è divisore di $1-x^T$ (in quanto non divisibile per x), allora $(g(x))_T$ è dato dagli Y(x) multipli di g(x).
Tenendo conto di quanto appena detto, si può specificare uno spazio ciclico a partire da un insieme di r elementi appartenenti a $GF(p^n, 1 - x^T)$.

Infatti sia $y_0, y_1, \ldots, y_{T-1}$ una sequenza appartenente allo spazio ciclico e sia $Y(x)$ il rappresentante della sequenza degli y_i nell'anello $[GF(p)[x], 1 - x^T]$:

$Y(x) = y_0 x^{T-1} + y_1 x^{T-2} + \ldots y_{T-2} x + y_{T-1}$

Ricordando che $Y(k_i) = 0$:

$$Y(k_i) = y_0 k_i^{T-1} + y_1 k_i^{T-2} + \ldots y_{T-2} k_i + y_{T-1} \qquad (1.11)$$

Sia in $GF(p^n, 1 - x^T)$:

$k_i = k_{1i} + k_{2i} x + \ldots + k_{ni} x^{n-1}$

allora si ottiene dalla (1.11):

$y_0 k_{hi}^{T-1} + y_1 k_{hi}^{T-2} + \ldots y_{T-2} k_{hi} + y_{T-1}$

con:

$h = 1, 2, \ldots, n$; $I = 1, 2, \ldots, r$.

Il tutto si può esprimere in forma matriciale, definendo:

$$\Gamma = \begin{bmatrix} k_{11}^{T-1} & k_{11}^{T-2} & \ldots & k_{11} & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ k_{n1}^{T-1} & k_{n1}^{T-2} & \ldots & k_{n1} & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ k_{r1}^{T-1} & k_{1r}^{T-2} & \ldots & k_{1r} & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ k_{nr}^{T-1} & k_{nr}^{T-2} & \ldots & k_{nr} & 1 \end{bmatrix}$$

da cui:

$(y_0, y_1, \ldots, y_{T-1}) \Gamma^T = 0$.

2. LSC per operazioni polinomiali

1.1. Introduzione

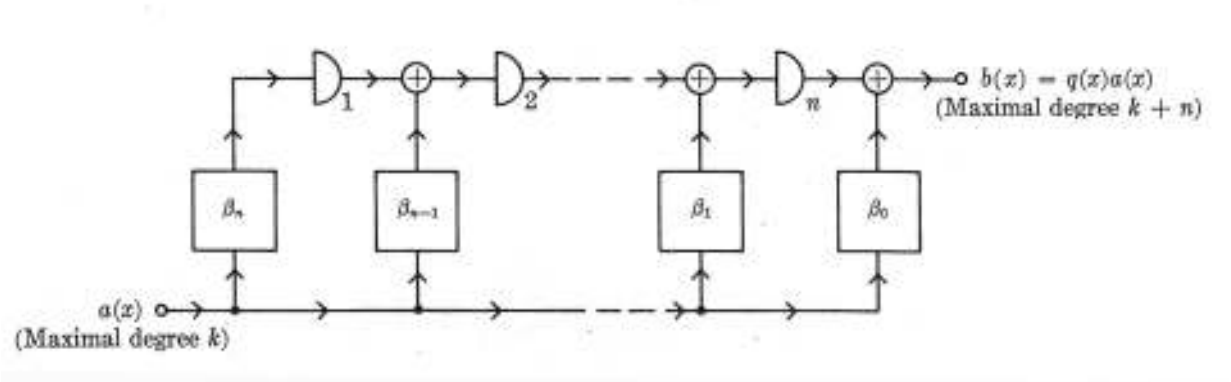
Di pari passo con lo sviluppo dell'elettronica digitale, delle telecomunicazioni e dell'informatica è sorta la necessità di realizzare circuiti che possano svolgere in maniera autonoma operazioni di moltiplicazione e divisione, indispensabili per il funzionamento di calcolatori, encoders, decoders e altri dispositivi, divenuti ormai comuni in campo ingegneristico. Di norma per svolgere tali operazioni si fa ricorso alla teoria dei campi finiti, essendo gli spazi di riferimento dei corpi finiti (si pensi per esempio al campo GF(2) usato per l'aritmetica binaria).

Gli LSC, con la teoria matematica di supporto ad essi associata, sono degli strumenti efficienti per risolvere questo tipo di problematiche. Progettati correttamente, possono eseguire operazioni tra polinomi, ricevendo in ingresso e producendo in uscita sequenze di simboli appartenenti ad un certo alfabeto finito.

Considerata la vastità dell'argomento, di seguito saranno analizzati solo quattro circuiti, che verranno poi utilizzati nella progettazione di encoders e decoders nel capitolo dedicato ai sistemi di comunicazione. Il primo di questi LSC realizza la moltiplicazione polinomiale, il secondo la divisione, il terzo implementa entrambe le operazioni in un unico circuito, mentre l'ultimo esegue la riduzione di polinomi.

1.2. Moltiplicazione polinomiale

Si supponga di voler calcolare il prodotto tra due polinomi a(x) e q(x) in maniera automatizzata. E' possibile risolvere il problema sfruttando le proprietà degli LSC? La risposta a questa domanda è ovviamente sì, come sarà spiegato a breve.



Sia dato il QLSC di shift-register $\mathcal{A}$, riportato in figura. Esso ha funzione di trasferimento:

$W(d) = \beta_0 + \beta_1 d + \ldots + \beta_n d^n$

Si ipotizzi di fornire, in ingresso al circuito, la sequenza di input :

$u(t) = n_0\, n_1\, n_2 \ldots\, n_k 0000\ldots$

cui corrisponde il polinomio alle trasformate D:

$U(d) = n_0 + n_1 d + \ldots + n_k d^k$

Ricordando i legami ingresso-uscita che regolano il funzionamento dei sistemi lineari, si ottiene l'espressione dell'output alle trasformate D, dato dal prodotto di W(d) per la trasformata D dell'ingresso, cioè U(d):

$Y(d) = U(d)\, W(d) = (n_0 + n_1 d + \ldots n_k d^k)(\beta_0 + \beta_1 d + \ldots + \beta_n d^n) = y_0 + y_1 d + \ldots + y_{n+k} d^{n+k}$

con:

$$y_i = \sum_{u+v=i} n_u \beta_v$$

Si consideri ora il prodotto di polinomi:

$f(x) = a(x)\, q(x) = (n_k + n_{k-1}x + \ldots + n_0 x^k)(\beta_n + \beta_{n-1}x + \ldots + \beta_0 x^n) = y'_{n+k} + y'_{n+k-1}x + \ldots + y'_0 x^{n+k}$

con:

$$y'_{n+k-j} = \sum_{u+v=n+k-j} n_u \beta_v$$

Si osserva che, ponendo n+k-j = i si ottiene proprio:

$$y'_i = \sum_{u+v=i} n_u \beta_v = y_i$$

che è per l'appunto l'espressione usata per descrivere i coefficienti del polinomio di uscita del QLSC, considerato al tempo t = i (a partire da x^{n+k}). Pertanto il polinomio f(x) si può scrivere come:

$f(x) = y_0 x^{n+k} + y_1 x^{n+k-1} + \ldots + y_{n+k-1} x + y_{n+k}$

E' quindi fondata l'idea di utilizzare un siffatto QLSC per svolgere operazioni di moltiplicazione polinomiale. Nello specifico con questo tipo di circuiti si può calcolare il prodotto tra un generico polinomio di ingresso a(x) e un polinomio fisso q(x):

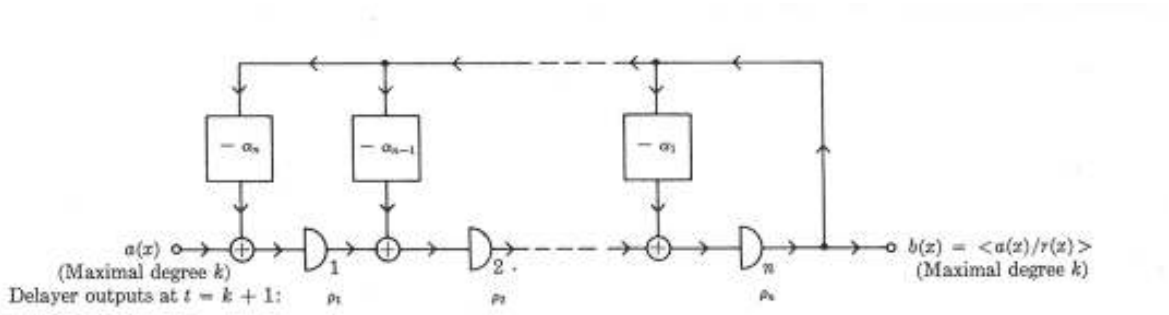
$q(x) = \beta_n + \beta_{n-1}x + \ldots + \beta_0 x^n$

q(x) viene considerato polinomio fisso perché corrisponde alla funzione di trasferimento del QLSC, pertanto non è possibile modificarlo senza alterare l'intero circuito. Ne deriva che, qualunque sia il polinomio fornito in ingresso rappresentato dalla sequenza dei suoi coefficienti, questi sarà comunque moltiplicato per q(x). E' altresì chiaro come lo schema realizzato sia esattamente lo stesso per ogni q(x). In effetti la menzionata modifica del circuito, riguarda semplicemente il contenuto dei blocchi moltiplicatori $\beta_0, \ldots, \beta_n$.

1.3. Divisione polinomiale

Il circuito, che verrà analizzato in questo paragrafo e che è riportato in figura, attua la divisione tra due polinomi a(x) e b(x), avvalendosi di un QLSC di shift-register con funzione di trasferimento:

$$W(d) = \frac{d^n}{1 + \alpha_1 d + ... + \alpha_n d^n}$$



Sia u(t) l'input passato al QLSC:

$u(t) = n_0 n_1 n_2 \ldots n_k 0000\ldots$

Applicando la trasformata D all'ingresso e ricordando le relazioni ingresso e uscita, si ottiene la D-trasformata dell'uscita:

$$Y(d) = U(d)\, W(d) = \frac{n_0 d^n + n_1 d^{n+1} + ... + n_k d^{n+k}}{1 + \alpha_1 d + ... + \alpha_n d^n} \qquad (2.0)$$

Tale espressione può essere riscritta svolgendo esplicitamente la divisione, ottenendo il polinomio:

$$Y(d) = n_0 d^n + n_1^{(1)} d^{n+1} + n_2^{(2)} d^{n+2} + ... + n_{k-n}^{(k-n)} d^k \qquad (2.1)$$

Il resto della divisione può essere espresso nella forma:

$$r(d) = p_n d^{1+k} + ... + p_1 d^{n+k}$$

A partire dal polinomio Y(d) si ricava la sequenza di uscita dell'LSC, data dai coefficienti:

$$00...0 n_0 n_1^{(1)} ... n_{k-n}^{(k-n)}$$

I primi n campioni sono nulli perché i blocchi ritardatori sono inizialmente azzerati e sono necessari n passi affinché il primo termine dell'ingresso raggiunga l'uscita, come è evidente dalla figura.

Si consideri ora la seguente espressione:

$$f(x) = \frac{n_0 x^k + n_1 x^{k-1} + ... + n_k}{x^n + \alpha_1 x^{n-1} + ... + \alpha_n} = \langle f(x) \rangle + \frac{f_r(x)}{x^n + \alpha_1 x^{n-1} + ... + \alpha_n} \qquad (2.2)$$

dove <f(x)> rappresenta la parte principale (ovvero il quoziente) di f(x), mentre $f_r(x)$ è il resto della divisione sempre tra numeratore e denominatore di f(x). Svolgendo i calcoli si ottiene:

$$<f(x)> = n_0 x^{n-k} + n_1^{(1)} x^{k-1-n} + .n_2^{(2)} x^{k-n-2} + ... + n_{k-n}^{(k-n)} \quad (2.3)$$

Dal confronto di <f(x)> con Y(d), riportate rispettivamente nell'equazione (2.1) e (2.3), segue immediatamente che il polinomio di uscita dell'LSC è proprio il quoziente <f(x)>.
Inoltre se si considerano le uscite dei blocchi ritardatori al tempo t = k + 1, si vede subito come queste coincidano con i coefficienti del resto della divisione.

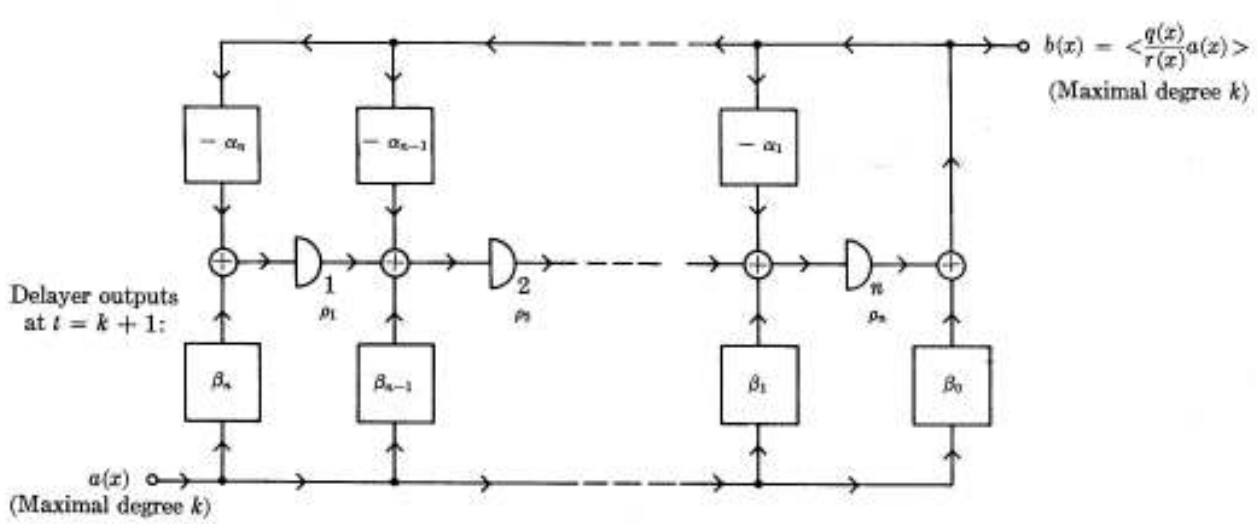
Questo circuito produce in uscita la parte principale della divisione tra due qualsiasi polinomi a(x) e b(x), con b(x) fisso che dipende dalla struttura interna dell'LSC, e presenta al tempo t = k + 1 nel polinomio interno, che è proprio il polinomio formato dalle uscite dei blocchi ritardatori ad un certo tempo τ, il resto della divisione.

Il ricorso agli LSC per svolgere divisioni tra polinomi è stato quindi giustificato.

1.4. Moltiplicazione e divisione polinomiale.

Si consideri il circuito riportato in figura, il cui scopo è quello di eseguire la divisione di due polinomi, moltiplicando il quoziente per un terzo polinomio:

$$a(x)\frac{q(x)}{r(x)}$$



Per descrivere il funzionamento di questo LSC si tenga conto di quanto detto precedentemente sui circuiti moltiplicatori e divisori.

Si consideri il dividendo della divisione (2.2) del precedente paragrafo, ovvero il polinomio:

$$n(x) = n_0 x^k + n_1 x^{k-1} + ... + n_k$$

Lo si moltiplichi per βx^h, con β appartenente a GF(p) e con $0 \leq h \leq n$. Come si è visto, utilizzando un circuito QLSC per svolgere la divisione polinomiale, i coefficienti dell'uscita dell'LSC, corrispondono a quelli della parte principale della divisione tra i polinomi. Avendo introdotto un termine moltiplicativo, nel caso sotto analisi si avrà che i coefficienti della parte principale coincideranno con quelli dell'uscita del QLSC una volta che il numeratore nell'equazione (2.0) sia stato correttamente moltiplicato per βd^{-h}.

Pertanto, il QLSC di shift-register che implementa le operazioni appena descritte avrà funzione di trasferimento:

$$W(d) = \frac{\beta d^{n-h}}{1 + \alpha_1 d + ... + \alpha_n d^n}$$

Detto a(x) il polinomio di ingresso, si ottiene, ricordando le relazioni ingresso-uscita, l'espressione dell'output:

$$y(x) = < \frac{\beta x^h a(x)}{x^n + \alpha_1 x^{n-1} + ... + \alpha_n} >$$

Più in generale, se il QLSC che implementa il circuito ha funzione di trasferimento:

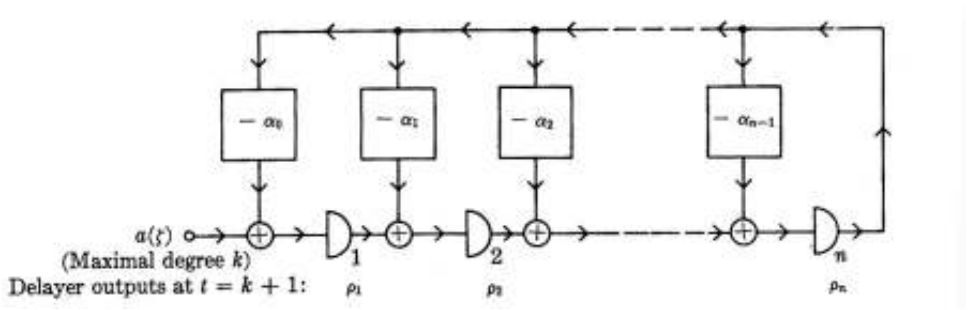
$$W(d) = \frac{\beta_0 + \beta_1 d + ... + \beta_n d^n}{1 + \alpha_1 d + ... + \alpha_n d^n}$$

allora il risultato dell'operazione è:

$$y(x) = < a(x) \frac{\beta_0 x^n + \beta_1 x^{n-1} + ... + \beta_n}{x^n + \alpha_1 x^{n-1} + ... + \alpha_n} > = < a(x) \frac{q(x)}{r(x)} >$$

dove a(x) è un generico polinomio di ingresso di grado massimo k, mentre q(x) e r(x) sono polinomi fissati dalla struttura del circuito, legati alla funzione di trasferimento dell'LSC utilizzato.
Come nel caso dell'esempio precedente, il resto della divisione è dato dal polinomio interno al tempo t=k+1.

1.5. Riduzione polinomiale



L'LSC in figura ha per scopo il calcolo della forma ridotta â(z) di un polinomio a(z), fornito in ingresso, tramite la successione dei suoi coefficienti; con forma ridotta si intende l'unico polinomio in z di grado al massimo n-1, che equivale ad a(z) in $GF(p^n, \Psi)$.

Sia quindi z diverso da zero un elemento di $GF(p^n, \Psi)$, con funzione minima:

$$m_z(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{n-1}\, x^{n-1} + x^n \qquad (4.0)$$

Si supponga che i blocchi ritardatori siano inizialmente azzerati e si fornisca come input al circuito il polinomio:

$$a(z) = b_k + b_{k-1}\, z + \ldots + b_0\, z^k \qquad (4.1)$$

si ricava:

$$\hat{a}\,(z) = p_1 + p_2\, z + \ldots + p_n\, z^{n-1} \qquad (4.2)$$

In un LSC come quello di figura, tale polinomio compare come polinomio interno al tempo t = k+1. Questa affermazione può essere dimostrata nel seguente modo. Si definisca

$$s_1(z) = b_0 \qquad (4.3)$$

Ciò significa che al tempo t = 1 il polinomio interno del circuito corrisponde esattamente a b_0.

Osservando il circuito si nota subito che per v = 2, 3, …, k+1 si ha:

$$s_v(z) = b_{v-1} + z s_{v-1}(z) \qquad (4.4)$$

Ricordando la (4.1), si ottiene la seguente uguaglianza:

$$s_{k+1}(z) = a(z) \qquad (4.5)$$

Si supponga che la forma ridotta di $s_v(z)$ sia:

$$\hat{s}_v(z) = c_0 + c_1 z + \ldots + c_{n-1}\, z^{n-1} \qquad (4.6)$$

Allora tenendo presente la (4.4) si può scrivere:

$$s_{v+1} = b_v + z(c_0 + c_1 z + \ldots + c_{n-1}\, z^{n-1}) = b_v + c_0 z + c_1 z^2 + \ldots + c_{n-1}\, z^n \qquad (4.7)$$

Dall'espressione della funzione minima (4.0) e dalle proprietà della stessa, si ricava la forma ridotta di $s_{v+1}(z)$, che è:

$$\hat{s}_{v+1}(z) = b_v + c_0 z + c_1 z^2 + \ldots + c_{n-2}\, z^{n-1} - c_{n-1}\,(\alpha_0 + \alpha_1 z + \ldots + \alpha_{n-1}\, z^{n-1}) \qquad (4.8)$$

Se, come visto precedentemente, b_0 è il polinomio interno al tempo t = 1 e se $\hat{s}_v(z)$ come nella (4.6) è il polinomio interno al tempo $t = v \leq k$, allora l'espressione della (4.8) rappresenta proprio il polinomio interno al tempo t = k+1. Per induzione si ricava l'uguaglianza tra $\hat{s}_{v+1}(z)$ e $\hat{a}(z)$, che risulta quindi essere, come supposto, il polinomio interno del circuito analizzato.

3. Sistemi di comunicazione: rilevazione e correzione di errori in trasmissione.

3.1. Introduzione

L'obiettivo principale dei sistemi di comunicazione, può essere riassunto in maniera esauriente dalle parole di Shannon:

"*Il problema fondamentale della comunicazione è quello di riprodurre in un punto, in maniera esatta o approssimata, un messaggio selezionato in un altro punto*."

Detto in altri termini, se un messaggio viene inviato attraverso un sistema di comunicazione si desidera riceverlo corretto, senza alterazioni. Sfortunatamente questo non è in genere possibile per la presenza di errori, che vanno a modificare il segnale inviato.

La causa primaria di errore è data dal rumore introdotto dal canale di trasmissione. Questo segnale indesiderato fa sì che il segnale ricevuto non coincida con quello di partenza e talvolta rende impossibile la ricostruzione del messaggio originario, obbligando ad una seconda trasmissione. Si possono infatti ricevere sequenze che non corrispondono a nessuna parola prevista, oppure si può ricevere una sequenza prevista dal codice al posto di un'altra anch'essa ammessa e quindi plausibile. Di conseguenza, risulta indispensabile realizzare sistemi che riconoscano la presenza di errori in modo che si possano prendere le contromisure necessarie, come il reinvio del messaggio originale o di parte di esso. Più utile ancora sarebbe avere a disposizione dei dispositivi che correggano gli errori una volta rilevati. A tale scopo è stata sviluppata la "teoria dei codici correttori", che vede tra i suoi fondatori R.W. Hamming.

L'idea di base è quella di codificare il segnale da inviare in modo tale che si possano individuare ed eventualmente correggere in ricezione eventuali errori in trasmissione. Per fare ciò si introduce in genere della ridondanza nel segnale originario. Si consideri il seguente esempio. Si supponga di voler inviare un messaggio rappresentante una tensione, attraverso un canale che vada in errore al più una volta ogni tre simboli trasmessi. Il sistema di codifica più semplice consisterebbe nello scegliere 1 come livello alto di tensione e 0 come livello basso. Tuttavia, una codifica siffatta non permetterebbe né di individuare né tantomeno di correggere eventuali errori. Infatti, ricevendo 0 al posto di 1, è impossibile riconoscere l'errore, poiché 0 è un valore accettato di codifica. È quindi necessario procedere diversamente. Supponiamo di scegliere come livello alto 11 e come livello basso 00: tale codifica introduce una ridondanza di un simbolo per ciascun livello. In questo caso, può essere individuato al più un errore nei messaggi ricevuti: se all'uscita del canale si rileva il segnale 01 o il segnale 10 si può dedurre che qualcosa non abbia funzionato e richiedere una seconda trasmissione. Rimane però impossibile risalire al messaggio originario trasmesso, in quanto non si sa quale sia la cifra errata tra le trasmesse (0 o 1). Per risolvere questo problema, è possibile introdurre un'ulteriore cifra ridondante nella codifica e porre rispettivamente come livelli

alto e basso 111 e 000. In questo caso, se il segnale ricevuto è 010, sicuramente si può dire che si è verificato un errore e, tenendo conto del fatto che il canale va in errore al massimo una volta ogni 3 simboli, ricostruire il segnale di partenza 000, i.e. il livello basso di tensione. Pertanto quest'ultima codifica, con ridondanza di due simboli, garantisce non solo la rilevazione, ma anche la correzione degli errori.

Naturalmente esistono sistemi di codifica molto più complessi di quello banale riportato in questo esempio, ma il principio base e l'obiettivo è lo stesso: aggiungere informazione per riconoscere gli errori e se possibile porvi rimedio.

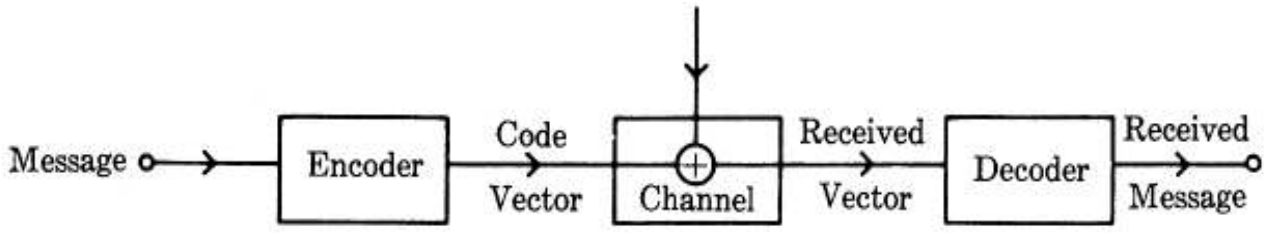
Rimane da specificare come implementare dispositivi di codifica e decodifica, ovvero gli encoders e i decoders, in modo da mettere in pratica i principi della teoria dei codici correttori.

Una possibilità è quella di usare i Circuiti Sequenziali Lineari (LSC), che, opportunamente progettati, permettono la realizzazione di sistemi capaci di rilevare e correggere errori in trasmissione per sistemi di comunicazione.

Nel proseguo saranno riportati esempi di encoders e decoders realizzati con LSC e ne verranno analizzati pregi e difetti, con riferimento alla problematica appena introdotta.

3.2. Encoders e decoders realizzati con LSC.

Si consideri un generico sistema di comunicazione, come quello riportato in figura, costituito da un encoder, un canale rumoroso e un decoder. Il segnale che si vuole trasmettere viene passato in ingresso all'encoder, che fornisce in uscita un segnale atto alla trasmissione. Tale segnale viene inviato attraverso il canale, ottenendo il segnale ricevuto, dato dal segnale inviato cui è aggiunto del rumore. Il segnale in uscita dal canale viene processato dal decoder, onde ricostruire il segnale originario.



Communication system

In questo schema, sono presenti due elementi che possono essere realizzati a partire dalle tecniche viste per gli LSC, ovvero l'encoder e il decoder. Il motivo alla base di questa affermazione richiede il richiamo di alcuni concetti di teoria introdotti nei precedenti capitoli.

Si è visto che una sequenza su GF(p) di lunghezza fissata n è una successione di simboli:

$f(t) = a_0a_1a_2\ldots a_n$

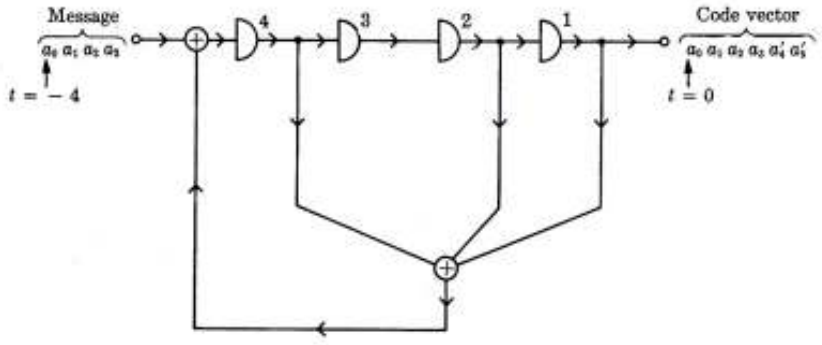
definita per tempi non negativi. Ciò significa che esistono p^n possibili sequenze tra loro distinte su GF(p). Sia quindi una di queste sequenze il messaggio da inviare all'encoder. Questo dispositivo trasforma il messaggio di partenza in una nuova sequenza sempre su GF(p) ma di lunghezza T, detta vettore-codice. L'insieme di tutti questi vettori è detto semplicemente codice.

Se i vettori che costituiscono il codice originano un sottospazio n-dimensionale dello spazio vettoriale T-dimensionale su GF(p), allora il codice, che in questo caso viene definito codice lineare, può essere specificato univocamente da una matrice nxT, le cui righe formano una base per tale spazio vettoriale. Ricordando che lo spazio ciclico è l'insieme di tutte le sequenze di uscita producibili da un ALSC semplice e non singolare, si prenda come spazio proprio uno spazio ciclico. Allora la matrice nxT definita dai vettore-codice può essere vista come una matrice di base di un ALSC e si può descrivere il blocco dell'encoder con i metodi visti nei precedenti capitoli. Un discorso analogo si può fare per i decoders. Risulta pertanto lecito utilizzare degli LSC per implementare circuiti di codifica e decodifica e tutta la teoria matematica di supporto può venire sfruttata per rendere più efficienti tali dispositivi.

Nei prossimi paragrafi verranno presentati alcuni esempi di encoders e decoders, con caratteristiche e prestazioni diverse e complessità crescente per quanto concerne la realizzazione.

3.3. Esempio 1

Come primo esempio di encoder si consideri l'ALSC di dimensione 4 su GF(2) riportato in figura.



Il polinomio di feedback e il periodo sono rispettivamente:

$\varphi(x) = 1 + x + x^3 + x^4$

$T = 6$

A partire dal circuito è possibile ricavare le matrici che descrivono il sistema:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \qquad D = 0$$

Lo stato iniziale è:

$s(0) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T$

avendo applicato in ingresso al sistema il segnale $a_0a_1a_2a_3$ al tempo t = -4.

Si considerino ora le equazioni che descrivono l'evoluzione libera dell'uscita (l'azione degli ingressi applicati a t = -4 termina in t = 0) a partire dal tempo t = 0. Si ha:

$$y(0) = C\,A^0\,s(0) = C\,I\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_0$$

$$y(1) = C\,A\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_1$$

$$y(2) = C\,A^2\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_2$$

$$y(3) = C\,A^3\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_3$$

$$y(4) = C\,A^4\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 3 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_0 + a_1 + a_3 = a'_4$$

$$y(5) = C\,A^5\,s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 2 & 3 & 2 & 4 \\ 4 & 6 & 3 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_0 + 2\,a_1 + a_2 + a_3 = a'_5$$

Se la sequenza di ingresso $a_0a_1a_2a_3$ corrisponde al segnale 1101 applicato al tempo t = -4, a partire dal tempo t = 0 si osserverà in uscita la seguente sequenza: 110110.

Si nota subito che le prime 4 cifre della sequenza di uscita corrispondono al segnale di ingresso. L'encoder restituisce, cioè, il messaggio originario inalterato nei primi 4 termini dell'uscita, aggiungendo in coda altri due termini dovuti alla dinamica interna del circuito.

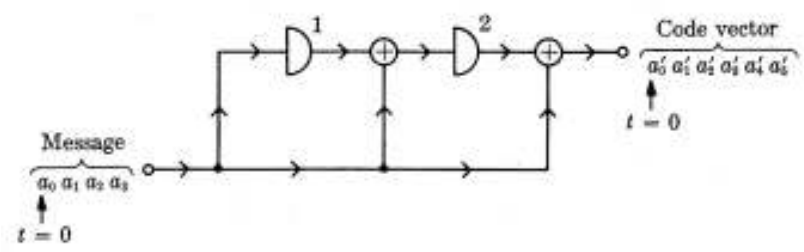
La possibilità di mantenere nell'uscita l'intero segnale originale è particolarmente utile; una volta ricevuto il segnale codificato in uscita dal canale, sarà sufficiente prelevare i primi 4 simboli del segnale ricevuto per ricostruire quello di partenza (naturalmente a meno di errori in trasmissione).

Il problema fondamentale di questo schema è la dimensione, basti considerare che già per una codifica (4,6) sono stati necessari 4 blocchi ritardatori.

Nel prossimo esempio verrà proposto un circuito che utilizza un numero inferiore di blocchi ritardatori per la stessa codifica.

3.4. Esempio 2

Si consideri l'encoder realizzato con il QLSC su GF(2) riportato in figura.



Come nell'esempio precedente il polinomio di feedback e il periodo sono:

$\varphi(x) = 1 + x + x^3 + x^4$

$T = 6$

mentre le matrici del sistema sono:

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad D = 1$$

Prima di proseguire è bene richiamare alcuni concetti.

Dalla teoria e da quanto visto precedentemente sui vettore-codice, si può dire che una sequenza $b_0b_1b_2\ldots b_{T-1}$ è un vettore-codice se e solo se si trova nello spazio ciclico, ovvero se e solo se è una

sequenza di output dell'LSC, il che significa che è rappresentata nell'anello $[GF(p)[x] \bmod 1 - x^T]$ da un elemento b(x) sempre in $[GF(p)[x] \bmod 1 - x^T]$.
Ricordando che $(g(x))_T$ è l'ideale con:

$$g(x) = x^{T-n} + \gamma_1 x^{T-n-1} + \ldots + \gamma_{T-n}$$

Quanto sopra si può esprimere in altri termini dicendo che la sequenza dei b_i è un vettore-codice se e solo se b(x) è un multiplo di g(x) in $[GF(p)[x] \bmod 1 - x^T]$. Inoltre si è visto che gli LSC possono essere usati per realizzare moltiplicazioni e divisioni polinomiali. Quindi si può cercare un modo per sfruttare quanto appena detto nell'implementazione fisica dell'encoder dell'esempio, che è proprio un circuito per la moltiplicazione di polinomi.
A partire dal polinomio di feedback e tenendo conto del periodo T si calcola il polinomio generatore del sistema:

$$g(x) = \frac{1 + x^6}{1 + x + x^3 + x^4} = (1 + x + x^2)$$

cui corrisponde in $[GF(p)[x] \bmod (1 - x^T)]$ proprio:

$$g(x) = (1 + x + x^2)_6$$

Sia il messaggio di ingresso all'encoder 1101; ad esso corrisponde il polinomio di ingresso:

$$a(x) = x^3 + x^2 + 1$$

ovvero il primo fattore della moltiplicazione, nonché il polinomio non fisso del circuito moltiplicatore (il fattore fissato è g(x)).
Eseguendo la moltiplicazione tra g(x) e a(x) si ottiene il polinomio di uscita:

$$b(x) = g(x)\, a(x) = (1 + x + x^2)(x^3 + x^2 + 1) = x^5 + x + 1$$

cui corrisponde la sequenza di uscita 100011.
Alla stessa conclusione si può pervenire ricavando le equazioni che descrivono l'evoluzione dell'uscita forzata del sistema (essendo il sistema sotto analisi un QLSC il suo stato iniziale è per definizione nullo). Considerando la stringa di ingresso $u(t) = a_0a_1a_2a_3$, si ottiene infatti:

$y(0) = D\,u(0) = a_0$

$y(1) = CA^0\,B\,u(0) + D\,u(1) = C\,B\,u(0) + Du(1) = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} a_0 + a_1 = a_0 + a_1$

$y(2) = C\,A\,B\,u(0) + C\,B\,u(1) + D\,u(2) = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} a_0 + \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} a_1 + a_2 = a_0 + a_1 + a_2$

$y(3) = C\,A^2\,B\,u(0) + C\,A\,B\,u(1) + C\,B\,u(2) + D\,u(3) = a_1 + a_2 + a_3 \qquad A^2 = 0$

$y(4) = C\,A^3\,B\,u(0) + C\,A^2\,B\,u(1) + C\,A\,B\,u(2) + C\,B\,u(3) + D\,u(4) = a_2 + a_3$

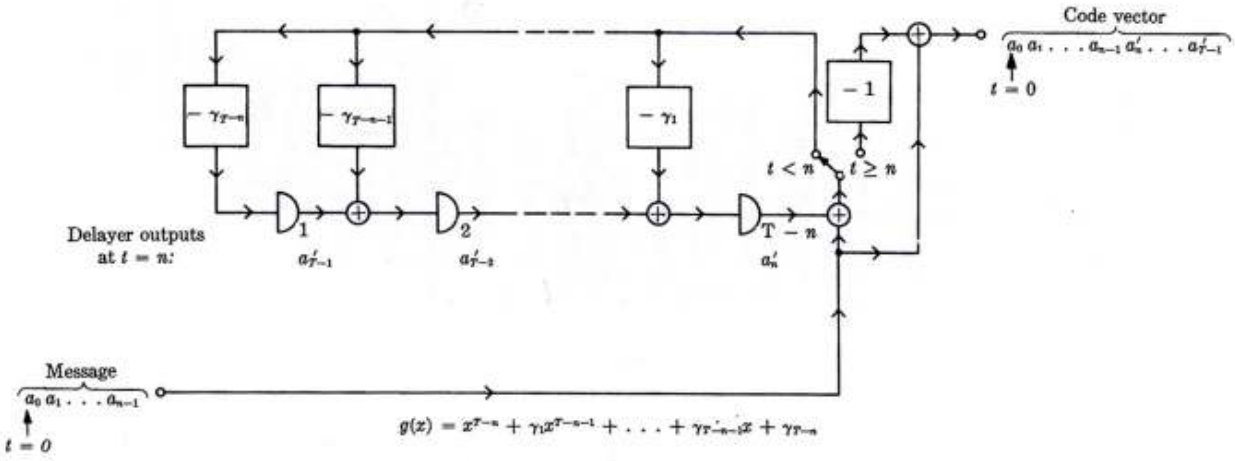
$y(5) = C\,A^4\,B\,u(0) + C\,A^3\,B\,u(1) + C\,A^2\,B\,u(2) + C\,A\,B\,u(3) + C\,B\,u(4) + D\,u(5) = a_3$

Sostituendo ad $a_0a_1a_2a_3$ la sequenza 1101 si ricava proprio lo stesso risultato visto prima.

Si rileva subito una differenza con l'encoder presentato nel primo esempio; quello appena proposto, infatti, genera un segnale in uscita completamente diverso da quello inviato, mentre il circuito dell'esempio 1 aveva nei primi 4 termini dell'uscita il segnale originale. Quest'encoder avrà quindi bisogno di un decoder più complesso a parità di prestazioni di quello necessario per l'Esempio 1. Tuttavia, considerando solo lo schema di codifica, la sua complessità è inferiore, visto che presenta un numero minore di blocchi ritardatori, rispetto al primo, pur realizzando la stessa codifica e il messaggio generato, essendo più criptato, ha un livello di sicurezza superiore, rispetto al primo.

3.5. Esempio 3

Oltre ai due encoders appena descritti ne esiste una terza tipologia e lo schema tipico di questo circuito è riportato in figura.



L'LSC, di periodo T, utilizzato nell'implementazione dell'encoder realizza un circuito per moltiplicazione e divisione polinomiale. In questo tipo di circuiti il polinomio corrispondente all'ingresso a(x), di grado massimo n-1, applicato al tempo t = 0, viene moltiplicato per q(x) che è il fattore fissato e poi diviso per r(x), polinomio divisore. In uscita viene calcolato il quoziente di:

$$\frac{\mathrm{q(x)\,a(x)}}{\mathrm{r(x)}} + \frac{w(x)}{r(x)}$$

ovvero:

$$\frac{q(x)a(x)}{r(x)}$$

Nel polinomio interno del circuito, al tempo t = n, compare il resto w(x) della divisione (di periodo massimo T – n - 1).

Si definiscano nel caso dell'encoder considerato i seguenti polinomi:

$q(x) = x^{T-n}$

$r(x) = g(x) = x^{T-n} + \gamma_1\, x^{T-n-1} + \ldots + \gamma_{T-n-1}\, x + \gamma_{T-n}$ con g(x) il polinomio generatore

$b(x) = q(x)\, a(x) / r(x)$

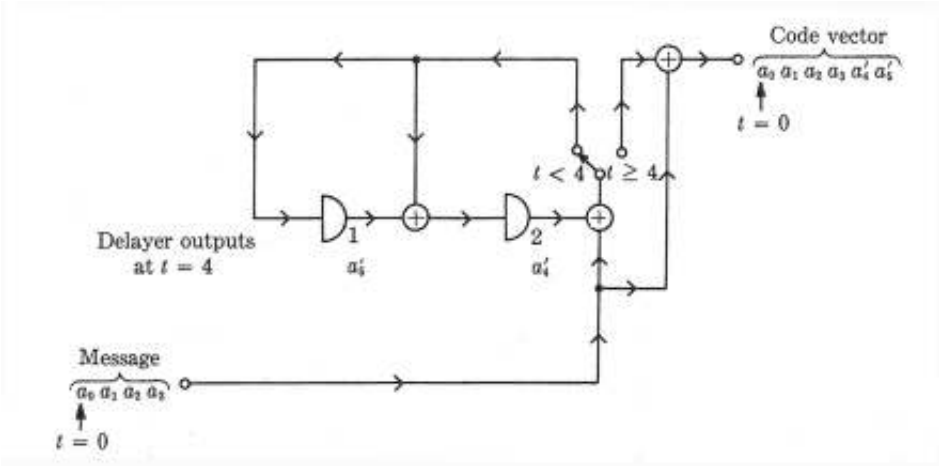
Allora si avrà:

$q(x)\, a(x) = x^{T-n}\, a(x) = b(x)\, g(x) + w(x)$

$x^{T-n}\, a(x) – w(x) = g(x)\, b(x) = b'(x)$

Dato che i T coefficienti di b'(x), che ha periodo massimo T-1, sono gli n coefficienti di a(x) seguiti dai coefficienti di w(x) cambiati si segno, il vettore-codice corrispondente a b'(x) può essere ricavato inviando prima i simboli del messaggio originale, ovvero a(x) al tempo 0, 1,…, n-1, seguiti dai T-1 coefficienti cambiati di segno dei blocchi ritardatori a partire dal tempo n, n+1, …., T-1. Questo passaggio è reso possibile dallo switch, che scatta a t = n.

Si consideri un esempio numerico per verificare la validità di quanto detto.



A partire da t = 0 e per t < 4, l'uscita del circuito è data direttamente dagli ingressi, visto che lo switch in posizione 1 rende inattivo il nodo sommatore nel ramo di uscita. Il valore di y(t) nei primi 4 istanti è quindi:

$y(0) = a_0$

$y(1) = a_1$

$y(2) = a_2$

$y(3) = a_3$

cioè il segnale di ingresso.

Si osservi ora la parte superiore del circuito, ovvero quella costituita dalle maglie con i blocchi ritardatori. Le matrici che ne descrivono il funzionamento sono:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

L'evoluzione dello stato, partendo da x(0) = 0, è:

$x(1) = B\ u(0) = Ba_0$

$x(2) = A\ B\ u(0) + B\ u(1) = A\ B\ a_0 + B\ a_1$

$x(3) = A^2\ B\ u(0) + A\ B\ u(1) + B\ u(2) = A^2\ B\ a_0 + A\ B\ a_1 + B\ a_2$

$x(4) = A^3\ B\ u(0) + A^2\ B\ u(1) + A\ B\ u(2) + B\ u(3) = a'_4$

$x(5) = A^4\ B\ u(0) + A^3\ B\ u(1) + A^2\ B\ u(2) + A\ B\ u(3) + B\ u(4) = a'_5$

con ovviamente a'_4 e a'_5 dipendenti dagli ingressi $a_0a_1a_2a_3$.

Al tempo t = 4 lo switch scatta in posizione 2. L'uscita ora non dipenderà più dagli ingressi correnti, visto che da u(4) in poi sono nulli, bensì dal valore dello stato nei blocchi ritardatori. Si avrà quindi:

$y(4) = a'_4$

$y(5) = a'_5$

Come per il circuito dell'Esempio 1, questo encoder produce in uscita un segnale che presenta nei primi n termini il messaggio di partenza. Il numero ritardatori complessivi utilizzati è T-n, un numero inferiore rispetto al primo circuito analizzato che, per mantenere il segnale inalterato, necessita di n ritardatori. L'LSC appena visto quindi presenta i pregi di entrambi quelli precedentemente analizzati: non modifica il segnale originale e usa un numero contenuto di blocchi ritardatori.

3.6. Trasmissione del vettore-codice e errori in trasmissione

Si sono visti alcuni esempi di encoders, che codificano il messaggio originario in un segnale adatto alla trasmissione, ovvero il vettore-codice. Una volta che il vettore-codice è stato ricavato, viene inviato attraverso il canale rumoroso, ottenendo in uscita il vettore ricevuto.

L'eventuale differenza tra i due vettori viene definita vettore-errore.

Definito il peso di un vettore come il numero di componenti diverse da zero e definita distanza tra due vettori come il numero di coordinate per cui differiscono, l'effetto dell'errore sul segnale ricevuto viene misurato in base al peso massimo del vettore di errore stesso.

Se il peso dell'errore è *w*, allora l'errore è in grado di modificare *w* componenti del vettore-codice.

Da notare è la seguente affermazione: un errore può essere rilevato se e solo se il vettore di errore non è esso stesso un vettore-codice nel codice lineare.

Infatti siano **e** il vettore-errore, **v** il vettore-codice e **v'** il vettore-ricevuto; si ha: **v'** = **v** + **e**.

Se **e** è un vettore-codice, allora anche la somma **v'** è nel codice (ovvero l'insieme di tutti i possibili vettore-codice) e quindi non è possibile rilevare la presenza di **e**. Se invece **e** non è un vettore-codice, allora **v'**- **v** non è nel codice lineare e quindi **e** può essere individuato.

Se la distanza minima in un codice è δ, allora può essere rilevato ogni errore che abbia peso minore o uguale a δ-1.

3.7. Decodifica e rilevamento di errori

Per capire se il segnale ricevuto è affetto da rumore, è necessario verificare se il vettore ricevuto sia o meno un vettore-codice. Un primo modo per affrontare questo problema è il seguente.
Si è visto che ad una sequenza $b_0b_1b_2\ldots b_{T-1}$ è associato un polinomio b(x) nell'ideale $(g(x))_T$ e che la sequenza è un vettore-codice solo se b(x) è multiplo di g(x).

Pertanto, per capire se il vettore ricevuto è affetto da rumore, è sufficiente dividere i due polinomi b(x) e g(x) e vedere se il resto della divisione sia nullo o meno: nel primo caso il segnale sarà effettivamente un vettore-codice, nel secondo ovviamente no.
Dato che gli LSC possono essere usati per implementare circuiti che attuano la divisione polinomiale, basta applicare b(x) come input al tempo t = 0 ad un circuito che abbia g(x) come divisore e valutare il polinomio interno, che è proprio il resto della divisione, al tempo t = T : se questo è diverso da zero, allora la sequenza in ingresso non è un vettore-codice e sicuramente il vettore-ricevuto contiene un errore.

Un siffatto LSC può essere quindi usato come rilevatore di errori accoppiandolo a qualunque encoder e, se il codificatore è sul genere di quello proposto nell'Esempio 2, in caso di assenza di

errori, permette di ricostruire il messaggio originario. Infatti il vettore-codice generato dall'encoder dell'Esempio 2 è dato dal prodotto di g(x) con il segnale di ingresso a(x). In mancanza di errori l'output del circuito divisore che funge da decoder è proprio a(x), ovvero il segnale che era stato inviato all'encoder. Se il segnale ricevuto è affetto da errori, sarà possibile individuarli, in quanto il resto della divisione tra i polinomi sarà non nullo, ma per ottenere il segnale corretto bisognerà richiedere una nuova trasmissione.

Per rendere più semplice la comprensione di quanto appena detto, verranno proposti alcuni esempi. Nel primo, si farà utilizzo dell'encoder proposto nell'Esempio 1 e del decoder per l'individuazione di errori in trasmissione, nel secondo lo stesso dispositivo di decodifica verificherà la correttezza di un codice, generato dall'encoder dell'Esempio 2.

3.8. Esempio 4

Nell'Esempio 1, fornendo in ingresso all'encoder la sequenza 1101, si genera il vettore-codice 110110. Si supponga di inviare il segnale attraverso un canale rumoroso e di voler verificare se il messaggio ricevuto sia o meno affetto da errori.

Sia 110010 il vettore ricevuto: si è verificato ovviamente un errore nella quarta cifra. Da quanto detto precedentemente sul funzionamento del decoder, in caso di errore il resto della divisione tra il polinomio corrispondente al codice ricevuto e il polinomio generatore del sistema deve essere non nullo.

Svolgendo tutti i calcoli della divisione, considerando gli stessi dati dell'Esempio 1, si ottiene:

$x^2 + x + 1$ $\quad$ $\mathbf{x^3 + x}$

$x^5 + x^4 + 0x^3 + 0x^2 + x + 0$

$x^5 + x^4 + x^3$

$0 + 0 + x^3 + 0x^2 + x$

$x^3 + x^2 + x$

$0 + x^2$

$x^2 + x + 1$

$x + 1$

Il resto della divisione è effettivamente non nullo, come supposto data la presenza di una cifra errata.

Sorge spontanea la seguente domanda: l'individuazione di errori vale solo per le cifre del codice originario, oppure anche per quelle inserite secondo il principio della ridondanza?

Si svolga nuovamente la divisione, questa volta considerando come sequenza di ingresso: 110100 (errore nella penultima cifra).

$x^2 + x + 1$ $\quad$ $\mathbf{x^3 + x}$

$x^5 + x^4 + 0x^3 + x^2 + 0x + 0$

$x^5 + x^4 + x^3$

$0 + 0 + x^3 + x^2 + 0x$

$x^3 + x^2 + x$

$0 + 0 + x$

Anche in questo caso il resto della divisione è diverso da zero, come è giusto che sia.

L'ultimo caso da verificare è quello in cui il vettore codice viene ricevuto correttamente senza errori: il resto della divisione deve essere nullo.

$x^2 + x + 1$ $\quad$ $\mathbf{x^3 + x}$

$x^5 + x^4 + 0x^3 + x^2 + x + 0$

$x^5 + x^4 + x^3$

$0 + 0 + x^3 + x^2 + x$

$x^3 + x^2 + x$

$0 + 0 + 0$

Come annunciato, il resto della divisione è zero ed il codice è stato ricevuto correttamente.

3.9. Esempio 5

Si consideri ora l'encoder proposto nell'Esempio 2 e si proceda come nel caso precedente. Sia 100001 la sequenza ricevuta: presenta un errore nella penultima cifra. Svolgendo la divisione tra il polinomio corrispondente all'ingresso e il polinomio generatore g(x), si ha:

$x^2 + x + 1$ $\quad$ $\mathbf{x^3 + x^2 + 1}$

$x^5 + 0x^4 + 0x^3 + 0x^2 + 0x + 1$

$x^5 + x^4 + x^3$

$0 + x^4 + x^3 + 0x^2 + 0x + 1$

$x^4 + x^3 + x^2$

$0 + 0 + x^2 + 0x + 1$

$x^2 + x + 1$

$0 + x$

Il resto della divisione è diverso da zero, come presupposto.

Se invece della sequenza errata si riceve il codice: 100011, il resto della divisione è nullo. Infatti:

$x^2 + x + 1$ $\quad \mathbf{x^3 + x^2 + 1}$

$x^5 + 0x^4 + 0x^3 + 0x^2 + x + 1$

$x^5 + x^4 + x^3$

$0 + x^4 + x^3 + 0x^2 + x + 1$

$x^4 + x^3 + x^2$

$0 + 0 + x^2 + x + 1$

$x^2 + x + 1$

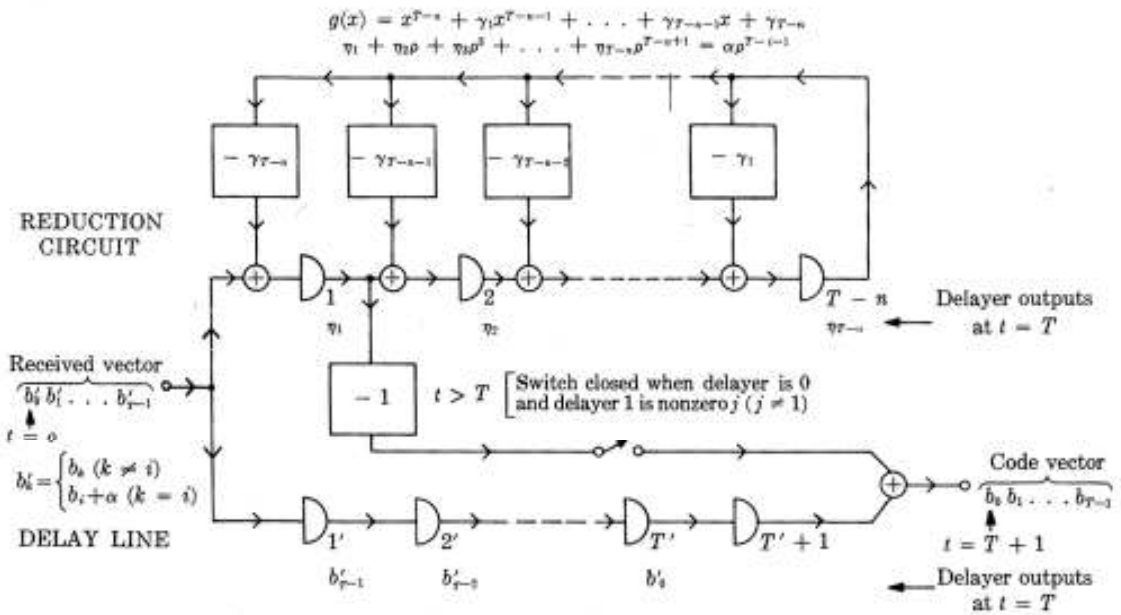
$0 + 0 + 0$

Si può notare una cosa interessante. Il risultato della divisione corrisponde proprio al polinomio che codifica il segnale iniziale. Ovvero è verificato quanto detto prima: in mancanza di errori a partire dal quoziente si è in grado di ricostruire il segnale originale.

3.10. Circuito per la correzione degli errori

Sarebbe opportuno, ai fini del problema di comunicazione esposto nell'introduzione, poter contare su un circuito che oltre a rilevare gli errori, possa anche correggerli, evitando così una seconda trasmissione dei dati, che comporterebbe gli stessi rischi di errore della prima.

Tale LSC esiste ed è riportato in figura.



Decoder for generalized Hamming code

Questo circuito completa gli schemi di codifica-decodifica visti per gli Esempi 1 e 3, e si presenta come il decoder più adatto alla ricostruzione del segnale di partenza, prevedendo la correzione del segnale ricevuto in caso di errori.

L'idea alla base di questo decoder è la tecnica di individuazione e correzione di errori del codice di Hamming binario.

Questo codice aggiunge ai simboli, appartenenti a GF(2), del messaggio originario degli altri simboli, in genere chiamati bit di parità, che vanno a "coprire/proteggere" quelli del messaggio originario. Nella fase di codifica, il vettore che costituisce il messaggio da trasmettere viene premoltiplicato per la matrice G, ovvero la generatrice del codice. Trasmesso il messaggio, si vuole verificare che non si siano verificati errori. Si premoltiplica pertanto il vettore ricevuto per la matrice di controllo H e se il risultato è zero, allora il messaggio è corretto, altrimenti il vettore risultante indica proprio la cifra errata. Nel qual caso è sufficiente negarla per ottenere il codice corretto.

Il circuito in esame estende questo ragionamento al caso generico. Si supponga che il vettore-codice sia la solita sequenza $b_0b_1b_2\ldots b_{T-1}$, che il corrispondente vettore-ricevuto sia $b'_0b'_1b'_2\ldots b'_{T-1}$ e che valga:

$$b'_j = \begin{cases} b_j & j \neq i \\ b_i + a & j = i \end{cases}$$

con $a = 0$ se non si sono verificati errori in trasmissione.

Come si vede dalla figura, il vettore-ricevuto viene inviato sia attraverso il "circuito di riduzione", sia attraverso la serie di blocchi ritardatori (inizialmente azzerati).
Dalle proprietà dei campi di Galois, si sa che b(x) è nell'ideale $(g(x))_T$ solo se:
$(b_0b_1b_2\ldots b_{T-1})\,\Gamma^T = 0$

dove:

$$\Gamma = \begin{bmatrix} k_1^{T-1} & k_1^{T-2} & \ldots & k_1 & 1 \\ & & & & . \\ k_2^{T-1} & k_2^{T-2} & \ldots & k_2 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ & & \ldots & & \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ k_n^{T-1} & k_n^{T-2} & \ldots & k_n & 1 \end{bmatrix}$$

e k appartenente a $GF(p^n, \Psi)$

Si può quindi dire che la sequenza $b_0b_1b_2\ldots b_{T-1}$ è un vettore-codice, e quindi non si sono verificati errori, se e solo se:

$b_0\, k^{T-1} + b_1\, k^{T-2} + \ldots + b_{T-2}\, k + b_{T-1} = 0$

il che significa:

$b'_0\, k^{T-1} + b'_1\, k^{T-2} + \ldots + b'_{T-2}\, k + b'_{T-1} = a\, k^{T-i-1}$

tenendo conto della: $b'_j = b_i + a$

Dalla dinamica interna del circuito di riduzione si ricava che, al tempo $t = T + i + 1$, l'uscita del primo blocco ritardatore è proprio pari ad a e l'uscita della serie di ritardatori è pari a b'_i.

Per cui se $-a$ è aggiunto all'output della serie di blocchi ritardatori al tempo $t = T + i + 1$, il risultato sarà proprio $b_i = b'_i - a$, ovvero il vettore-codice.

L'operazione di correzione è realizzata attraverso lo switch, mediante il quale all'output della serie di ritardatori è sommato $-a$, ovvero l'output cambiato di segno del primo ritardatore. Lo switch si chiude se e solo se l'output del blocco ritardatore 1 ha valore $a \neq 0$, mentre l'output di tutti gli altri è zero, per un intervallo di tempo che va da $T + 1$ a $2T$.

Naturalmente se $a = 0$, lo switch non si chiude mai e il vettore in uscita è il vettore-codice desiderato.

Sia nel caso sia eseguita la correzione degli errori, sia in caso di trasmissione corretta, il messaggio originario può essere ricavato dalla sequenza di output semplicemente considerando i primi n simboli a partire dal tempo $t = T + 1$.

Da quanto visto fin'ora risulta evidente che gli LSC possono essere dei validi strumenti per affrontare il problema della comunicazione, così come Shannon l'aveva espresso. Permettono infatti la progettazione di encoders efficienti e di decoders in grado di far fronte agli errori, individuandoli ed eventualmente correggendoli. Consentono inoltre di realizzare dispositivi di complessità variabile a seconda delle esigenze di progetto e delle specifiche da perseguire.

4. Bibliografia

- ARTHUR GILL, *Linear Sequential Circuits Analysis, Synthesis, and Applications*, , McGraw-Hill Book Company,1966
- MAURO BISIACCO, SIMONETTA BRAGHETTO, *Teoria dei sistemi dinamici*, Progetto Leonardo Esculapio-Bologna, 2010
- E. FORNASINI, G. MARCHESINI, *Appunti di teoria dei sistemi*, Edizioni Libreria Progetto Padova, 2003
- FRANCESCO MAZZOCCA, *Lucidi del corso di codici lineari*, Seconda Università degli Studi di Napoli, Corso di Laurea in Matematica e Informatica, A.A. 2009/10 (http://francesco.mazzocca.name/CODICI/Codici09-10.pdf)
- R. BERNARDINI, *funwithmodn*, Università di Udine, Lucidi del corso di Sistemi di Telecomunicazione, A.A. 2007 (http://www.diegm.uniud.it/~bernardi/Didattica/Sis/funwithmodn.pdf)

Testi di consultazione proposti

- ROBERTO MORESCO, *Lezioni di algebra lineare e geometria 3^a edizione*, Edizioni Libreria Progetto Padova, 2006
- CORRADO ZANELLA, *Fondamenti di algebra lineare e geometria*, Progetto Leonardo Bologna, 2010
- GENE F. FRANKLIN, J.DAVID POWELL, ABBAS EMAMI- NAEINI, *Controllo a retroazione di sistemi dinamici volume II*, EdiSES, 2005
- DORIANO CISCATO, *Appunti di controllo digitale A.A. 2009/10*

5. Indice