

---

# Il Delay Lock Loop

# Misura della distanza

---

- Misura continua della distanza usando sequenza di impulsi



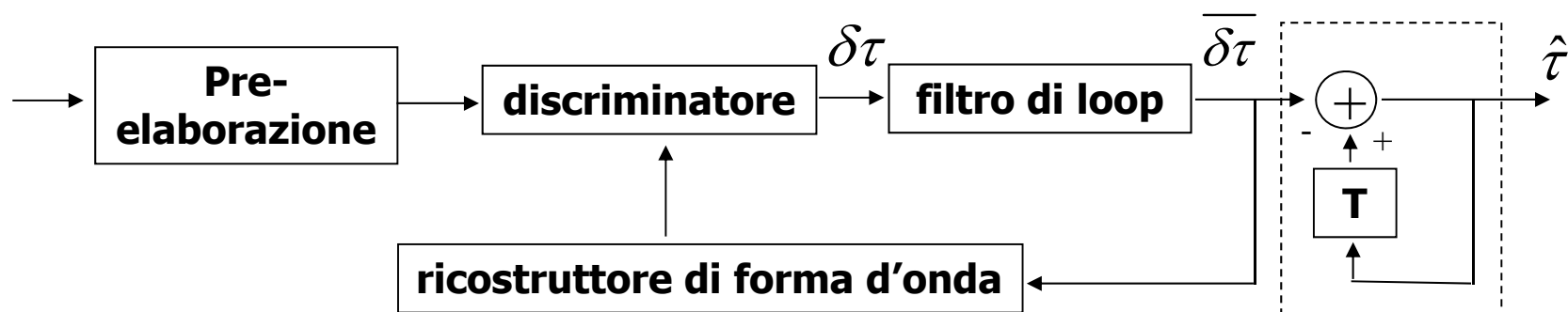
- *Si vuole misurare continuamente una distanza che varia nel tempo.*
- *Si assume di ricevere una sequenza di impulsi dal riferimento, trasmessi con intervallo di ripetizione costante.*
- *Ovviamente si vuole ottenere la migliore accuratezza di stima possibile, mediando le misure su più impulsi.*



- mediare a blocchi, ha due svantaggi:
  - misure disponibili solo alla fine del blocco
  - errore di misura dovuto al moto relativo durante il blocco
- effettuare una media mobile, risolve il primo problema ma non il secondo

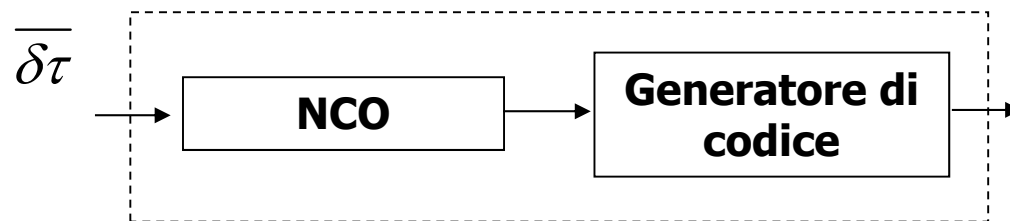
# Il principio del Delay Lock Loop

- se assumo di avere una stima ragionevole, posso pensare di “raffinarla” ad ogni nuovo impulso e allo stesso tempo di tenere in conto il fatto che è possibile che la distanza stia cambiando
- **l’approccio del DLL è composto dei seguenti passi:**
  1. generare una replica del segnale che ci si aspetta di ricevere, assumendo che la stima di distanza disponibile sia corretta;
  2. calcolare l’errore di disallineamento temporale fra il segnale ricevuto e quello generato (**discriminatore**);
  3. mediare le correzioni da introdurre (**filtro di loop**);
  4. correggere la stima di distanza = istante in cui mi aspetto di ricevere l’impulso successivo
- E’ una stima di ritardo ad anello chiuso: **Delay Lock Loop**



# Ricostruttore di forma d'onda

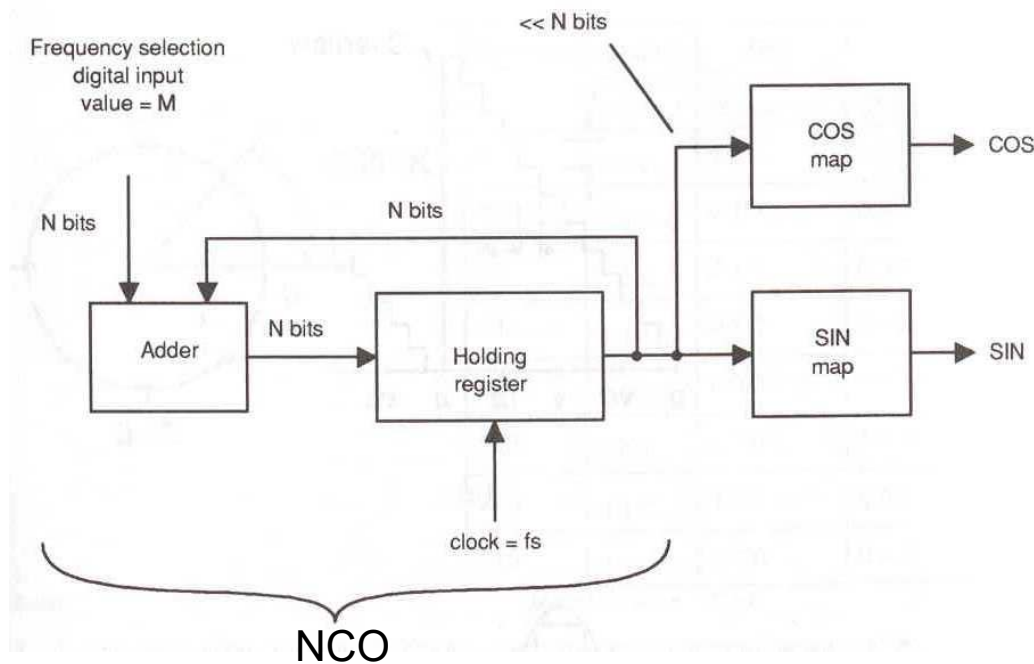
---



- Ha lo scopo generare il codice PRN del satellite considerato, ritardato (o anticipato, a seconda del segno) della quantità  $\overline{\delta\tau}$
- può ottenersi usando un generatore di codice (coppia di registri a scorrimento, opportunamente utilizzati) identico a quello utilizzato per la generazione in trasmissione nel payload del SV
- Come sappiamo la larghezza dei chip del generatore di codice è ottenuta pilotando i registri a scorrimento con un clock a 1.023 MHz. Ovviamente, accelerando tale clock si ottengono chip più corti ed i bit del codice tendono ad anticipare. Analogamente, decelerando il clock si ottengono chip più lunghi ed i bit del codice tendono a ritardare
- Quindi si può usare come clock un oscillatore, la cui frequenza vari (a partire da 1.023 MHz) in funzione dell'errore  $\overline{\delta\tau}$

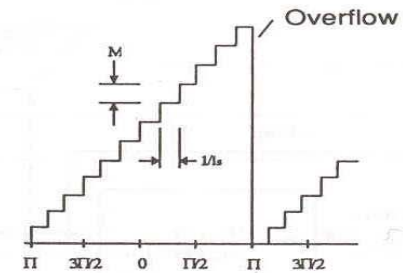
# Schema di un NCO

- NCO: Numerically Controlled Oscillator  
*usato sia per il loop del codice che per il loop di fase*

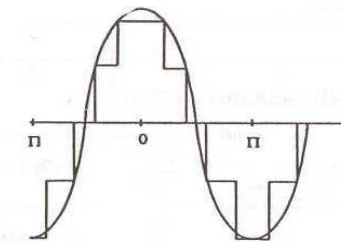


- $N$  = lunghezza del registro di tenuta
- $2^N$  = massimo valore del contatore
- $\frac{f_s M}{2^N}$  = frequenza di uscita
- $\frac{f_s}{2^N}$  = risoluzione di frequenza

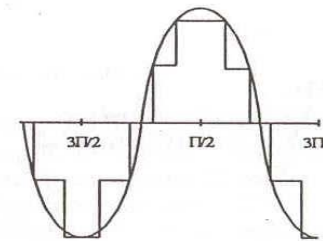
Radiotecnica e Radiolocalizzazione



(a)



(b)



(c)

# Utilizzazione dell'NCO nel loop

---

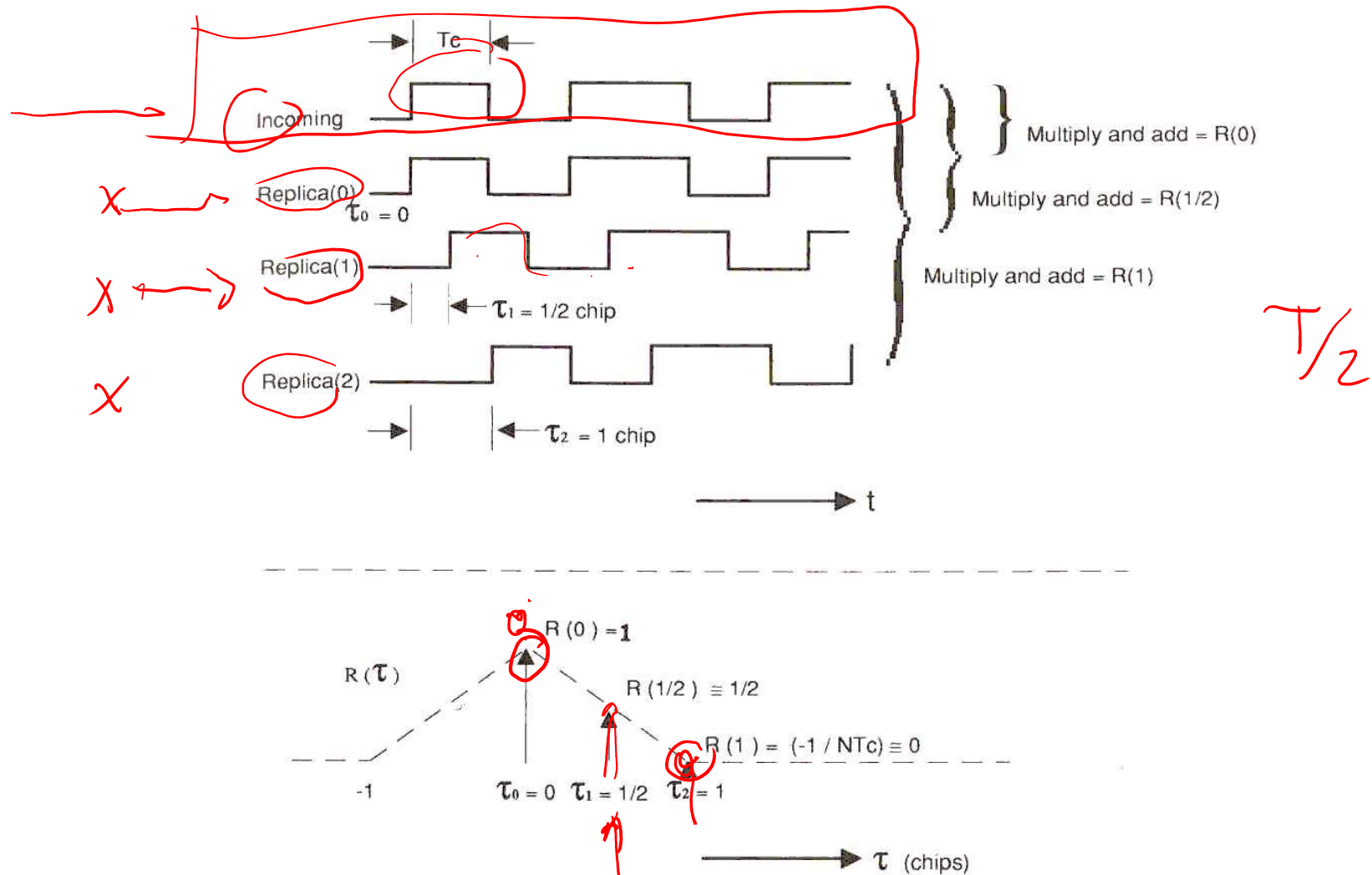
- *avendo la stima di ritardo di un dato impulso, si vuole predire il successivo*
- *gli impulsi codificati arrivano a distanza  $T$ , quindi serve un dispositivo che funzioni da “contatore” di intervalli elementari ( $1/f_s$ ), fino ad arrivare a  $T$*
- *arrivato a  $T$ , il dispositivo deve ricominciare a contare da zero, per l'impulso successivo: è dunque un oscillatore digitale*
- *se il ritardo stimato è in ritardo rispetto al tempo di arrivo della forma d'onda (come da uscita del discriminatore) è necessario ridurre  $T$  per compensare*
- *la riduzione di  $T$  si può effettuare facilmente aumentando la velocità di conteggio del contatore*
- *se il ritardo stimato è in anticipo rispetto al tempo di arrivo della forma d'onda (come da uscita del discriminatore) è necessario aumentare  $T$  per compensare*
- *l'aumento di  $T$  si può effettuare facilmente diminuendo la velocità di conteggio del contatore*
- *l'aumento e la velocità di conteggio di un NCO, si controllano in modo numerico, con la parola di ingresso al contatore*

# Il discriminatore

---

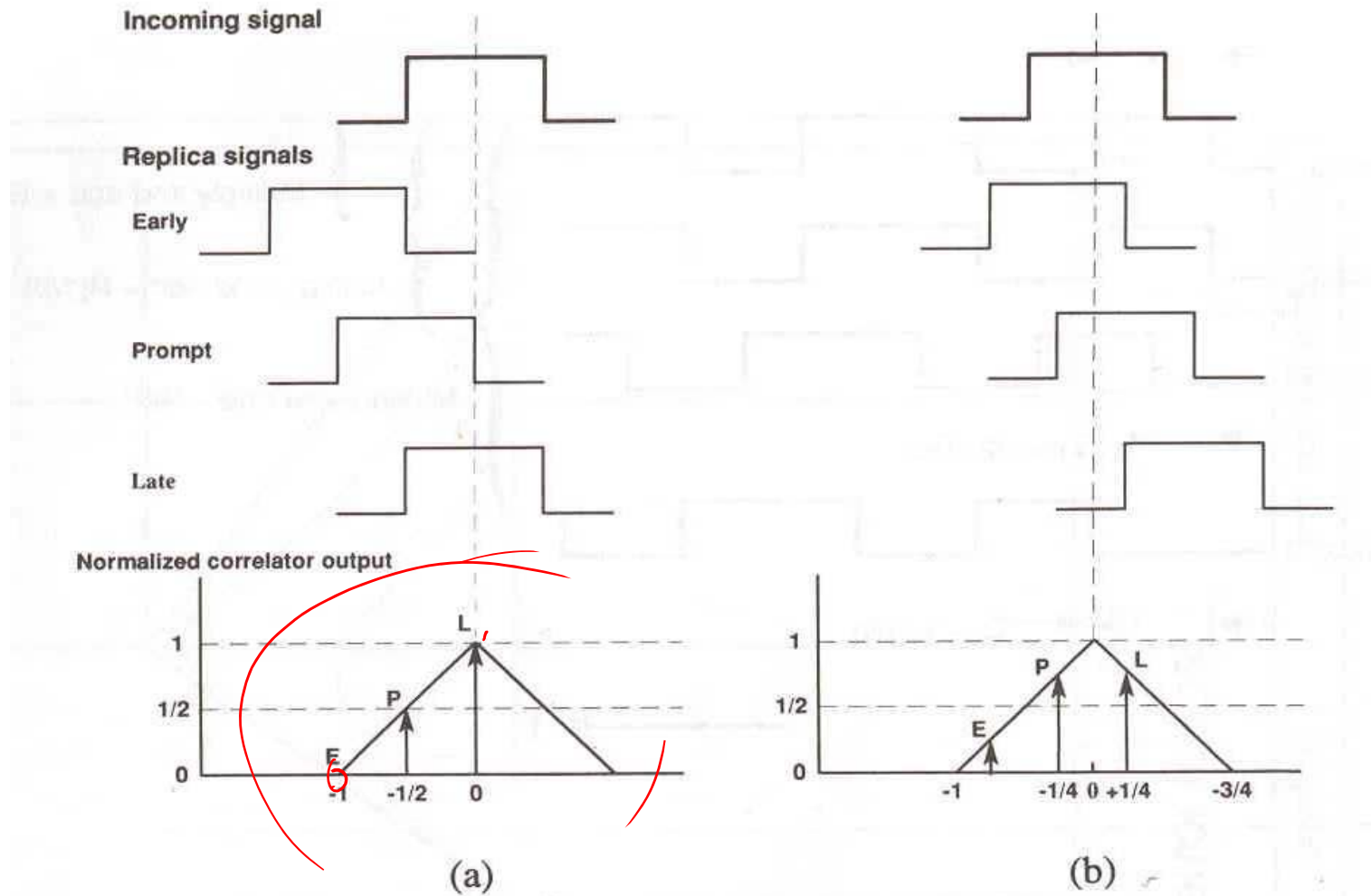
- per “raffinare” la stima mi serve un segnale di errore:
  - *segnale di valore zero se ho stima corretta*
  - *segnale con andamento monotono con l’errore*
- *si ricorda che la tecnica di stima basata sul calcolo del massimo dell’uscita del filtro adattato, porta ad annullare l’uscita di un filtro pari alla combinazione di filtro adattato e derivatore*
- *l’uscita di un tale filtro ha proprio le caratteristiche richieste ad un discriminatore*
- *stesso vale per approssimazioni, come in particolare per lo Early-Late Gates e per il Separated-Gates (approx del derivatore)*
- *possiamo utilizzare dunque questi schemi come discriminatori*

# Filtro adattato e discriminatore (I)

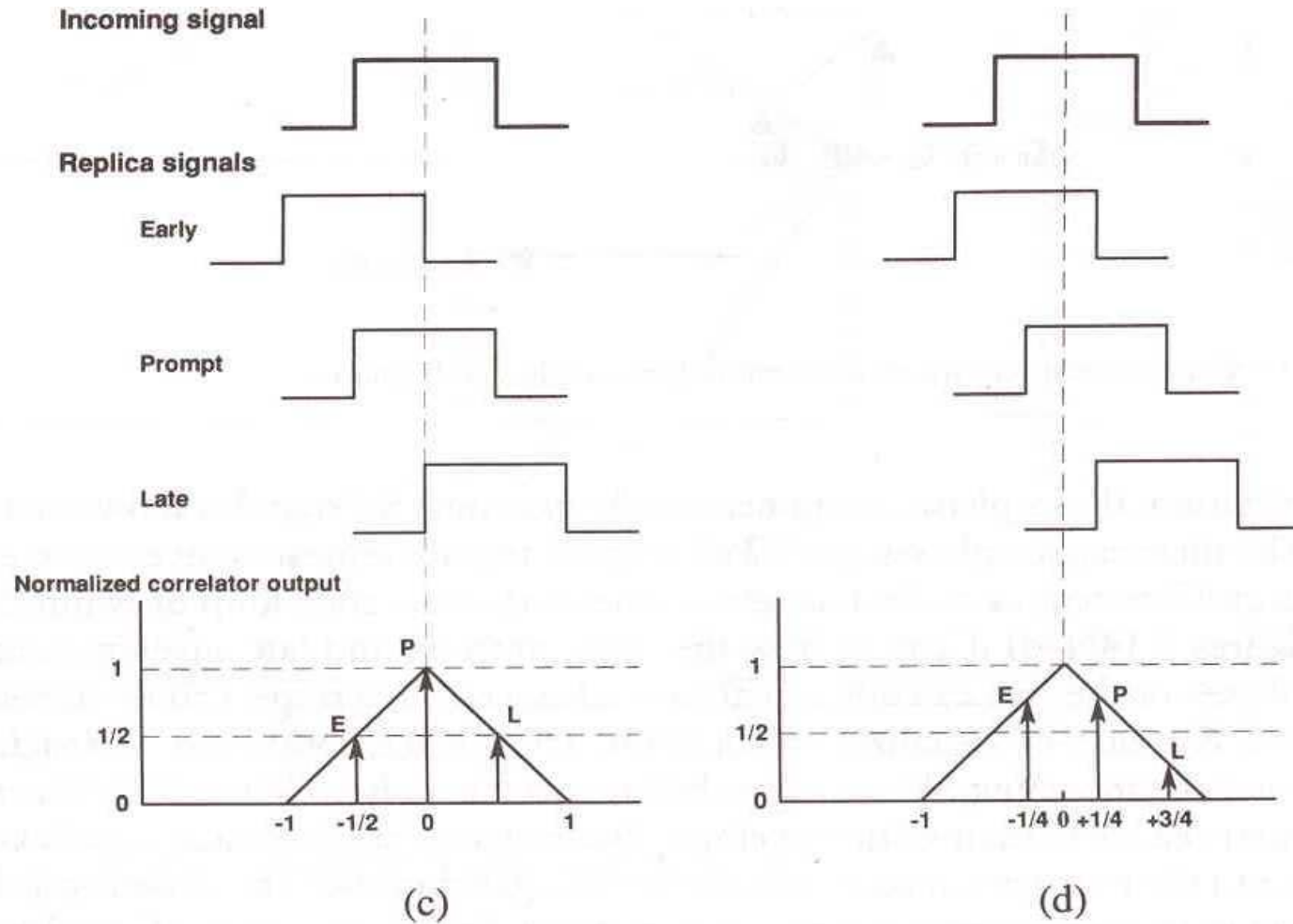




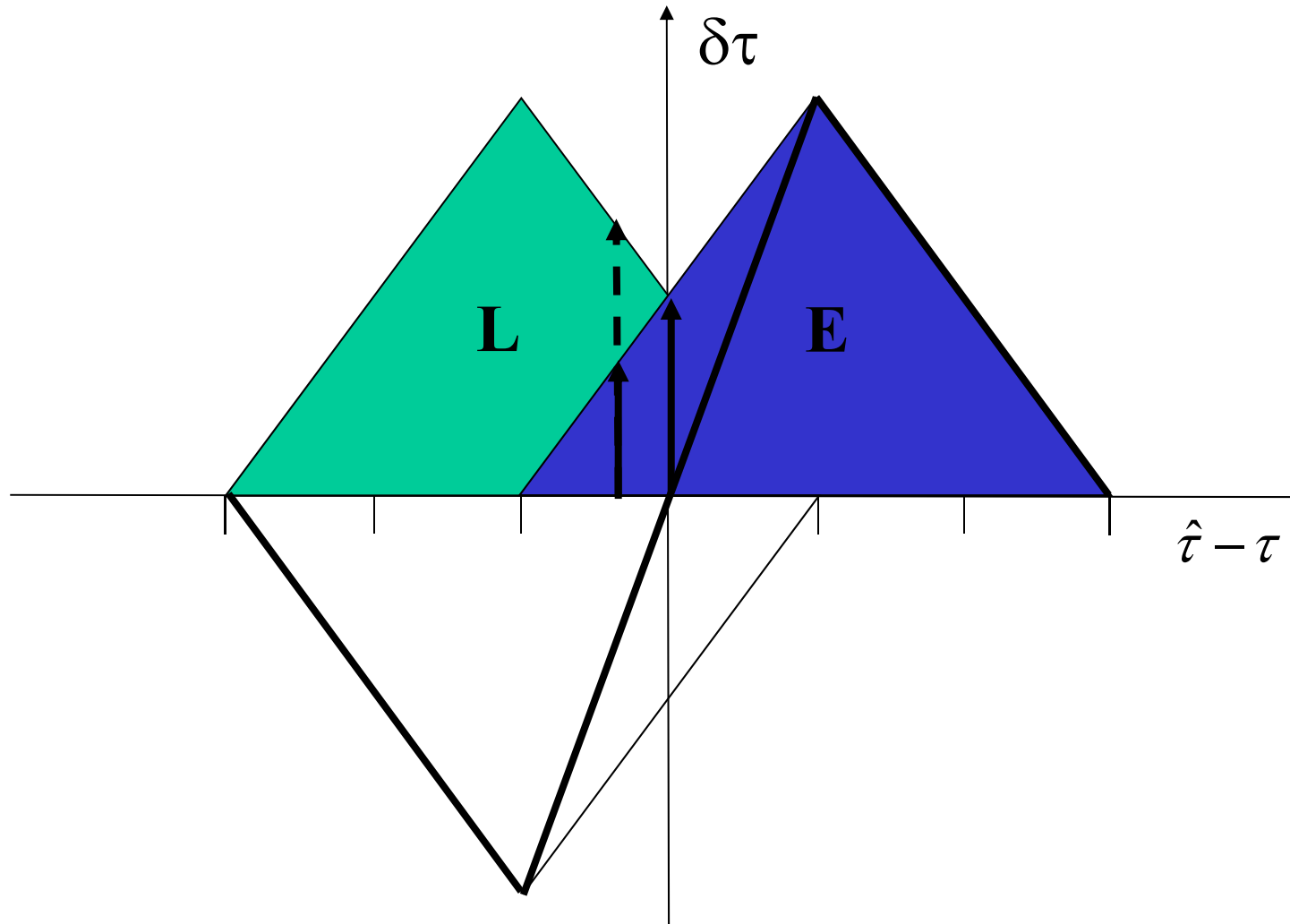
# Filtro adattato e discriminatore (II)



# Filtro adattato e discriminatore (III)



# Filtro adattato e discriminatore (IV)

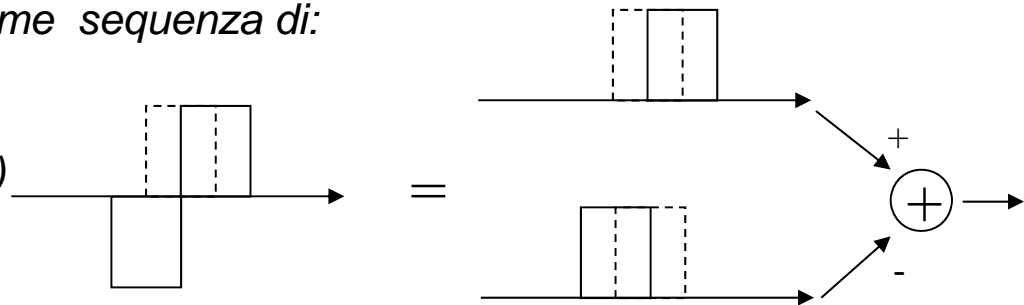


# Discriminatore per segnali codificati (I)

- **Per impulso singolo:**

implementazione dello E-L Gates come sequenza di:

- prodotto con impulso stesso anticipato (Early) e ritardato (Late) di 1/2 chip

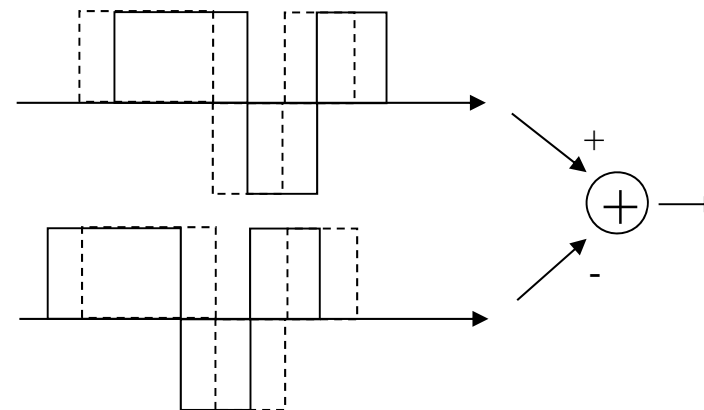


- Integrate & Dump

- Error detector

- **Per segnale codificato:**

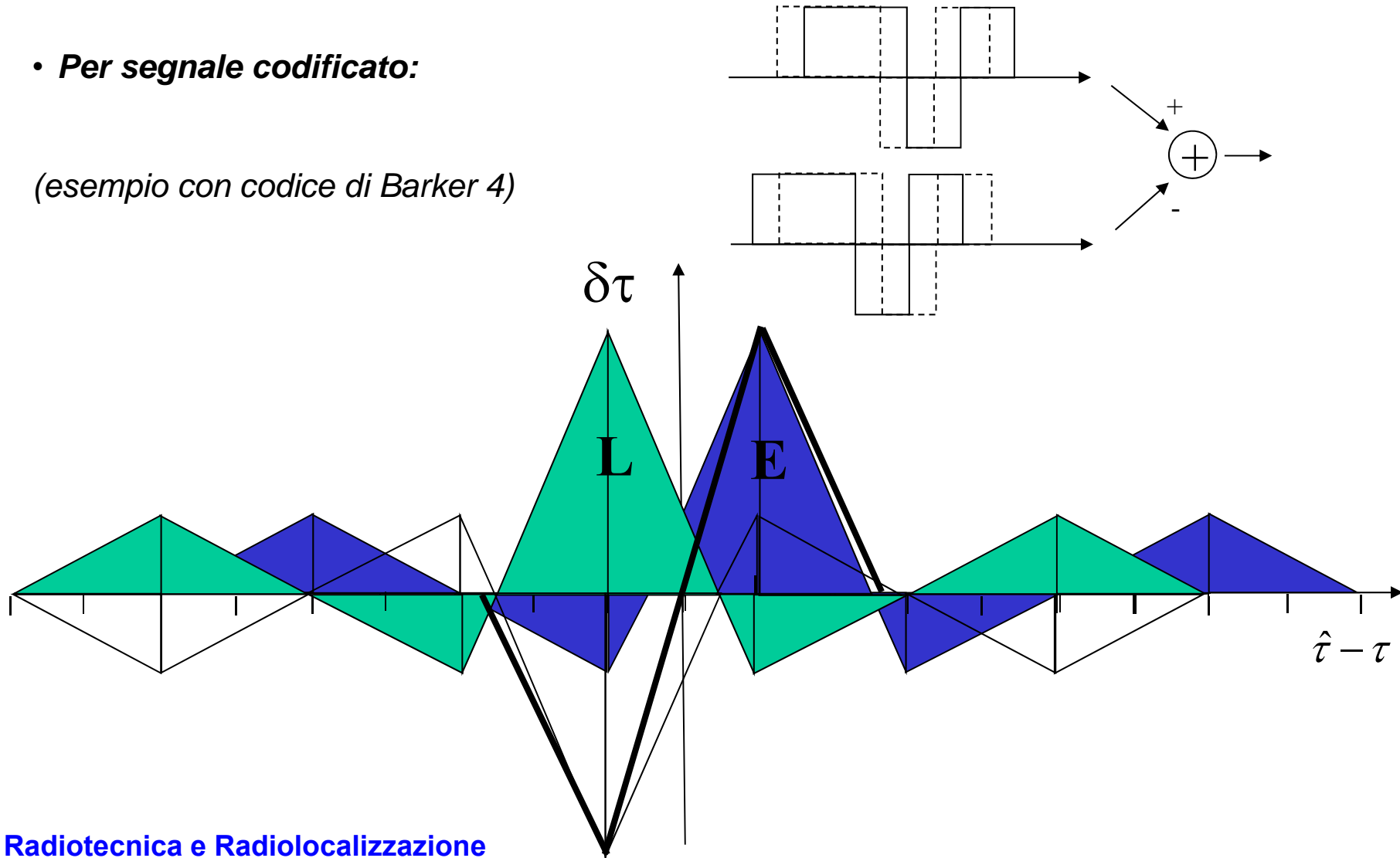
(esempio con codice di Barker 4)



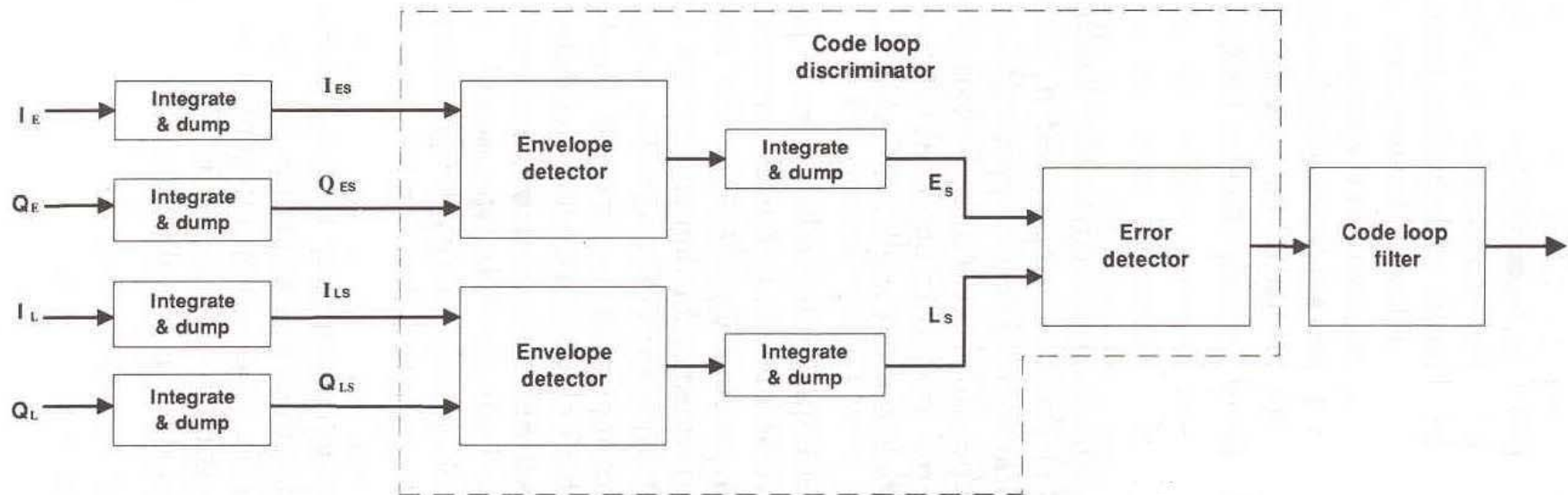
# Discriminatore per segnali codificati (II)

- *Per segnale codificato:*

(esempio con codice di Barker 4)



# Implementazione del discriminatore per il DLL



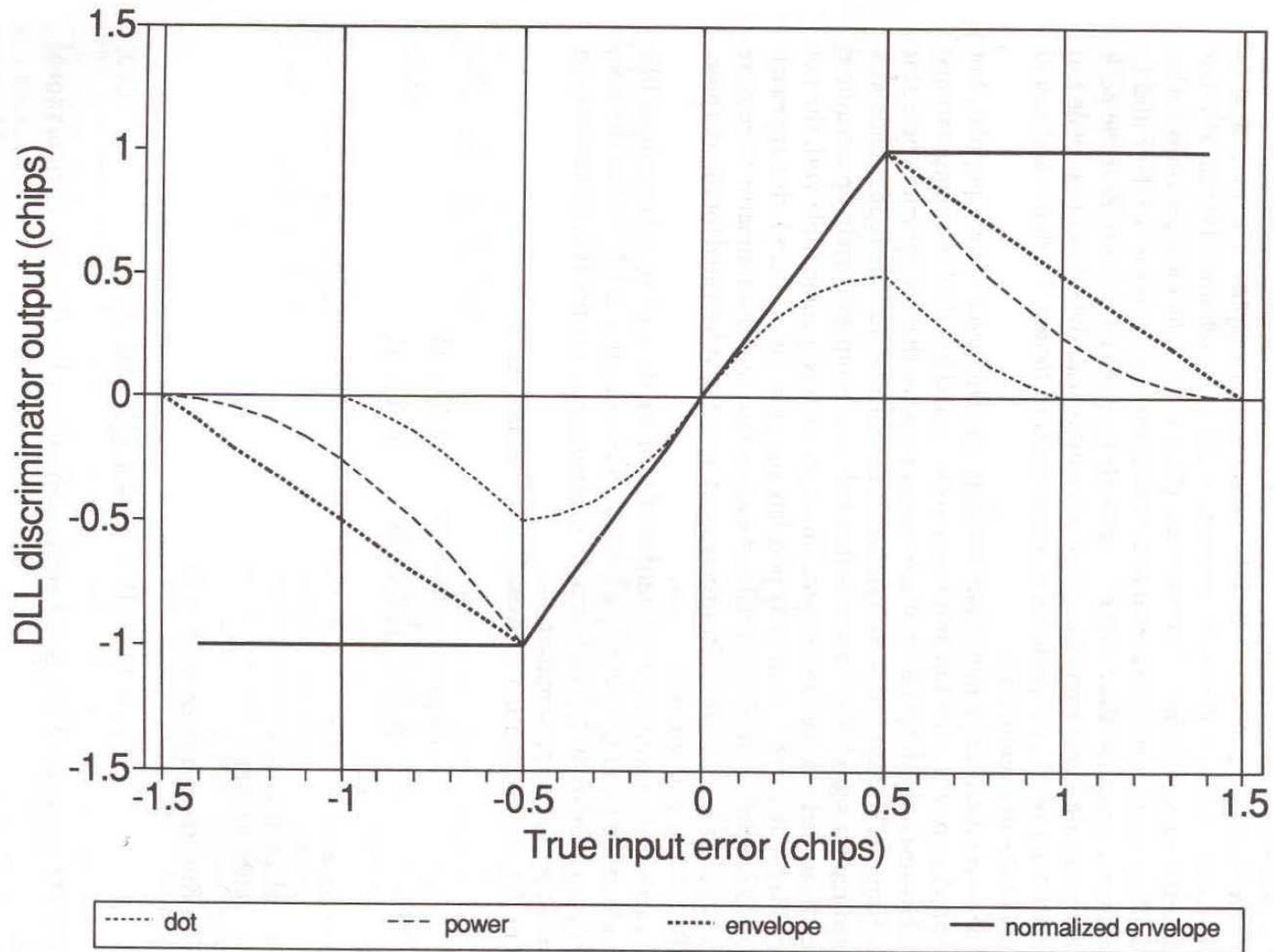
- *implementazione dello E-L Gates come sequenza di*
  - *Integrate & Dump*
  - *Error detector*

*su due segnali: uno anticipato ed uno ritardato*

# Discriminatori per il DLL

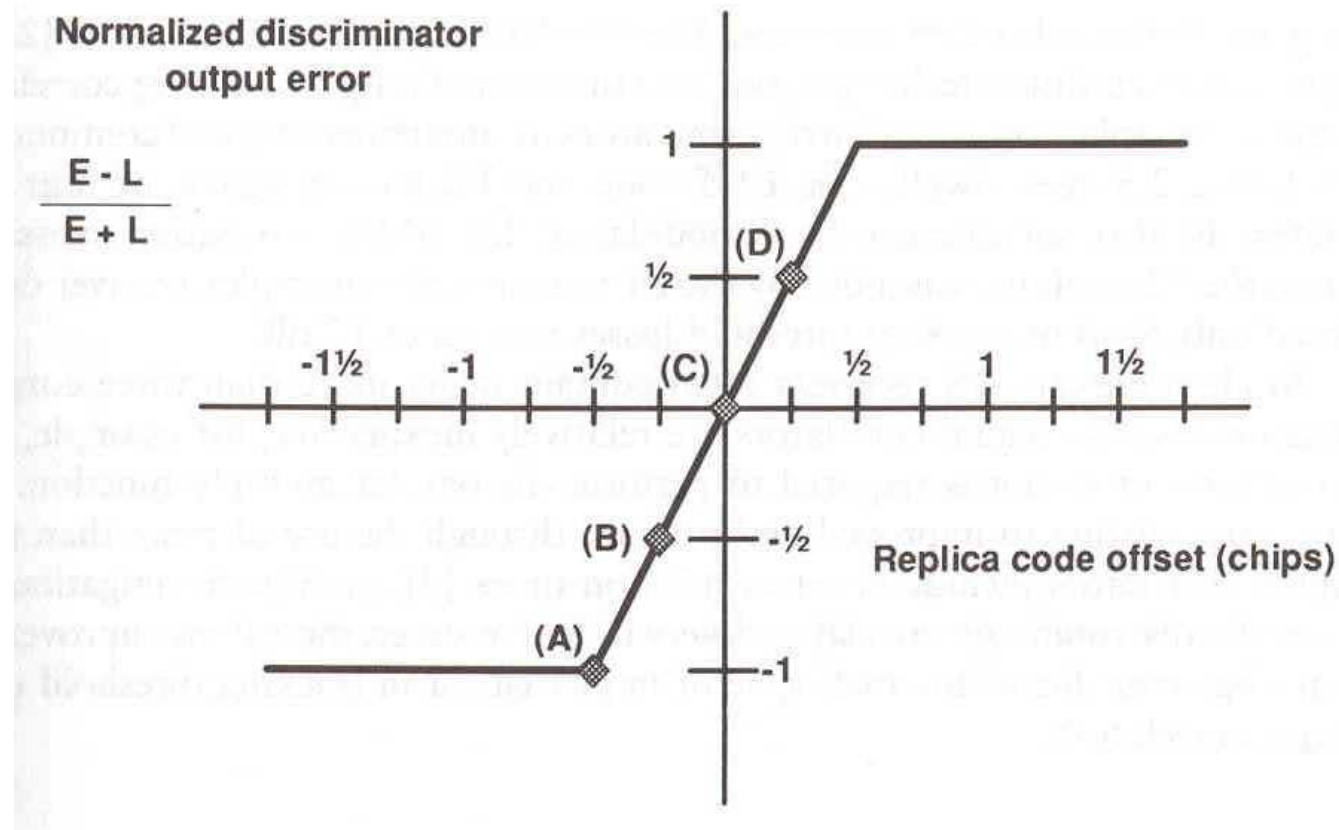
<i>Discriminator Algorithm</i>	<i>Characteristics</i>
$\sum(I_{ES} - I_{LS})I_{PS} + \sum(Q_{ES} - Q_{LS})Q_{PS}$	Dot product power. This is the only DLL discriminator that uses all three correlators and this results in the lowest baseband computational load. For 1/2 chip correlator spacing, it produces nearly true error output within $\pm 1/2$ chip of input error.
$\sum(I_{ES}^2 + Q_{ES}^2) - \sum(I_{LS}^2 + Q_{LS}^2)$	Early minus late power. Moderate computational load. Essentially the same DLL discriminator error performance as early minus late envelope within $\pm 1/2$ chip of input error.
$\sum\sqrt{(I_{ES}^2 + Q_{ES}^2)} - \sum\sqrt{(I_{LS}^2 + Q_{LS}^2)}$	Early minus late envelope. Higher computational load. For 1/2 chip correlator spacing, produces good tracking error within $\pm 1/2$ chip of input error.
$\frac{\sum\sqrt{(I_{ES}^2 + Q_{ES}^2)} - \sum\sqrt{(I_{LS}^2 + Q_{LS}^2)}}{\sum\sqrt{(I_{ES}^2 + Q_{ES}^2)} + \sum\sqrt{(I_{LS}^2 + Q_{LS}^2)}}$	Early minus late envelope normalized by the early plus late envelope to remove amplitude sensitivity. Highest computational load. For 1/2 chip correlator spacing, produces good tracking error within less than $\pm 1.5$ chip of input error. Becomes unstable (divide by zero) at $\pm 1.5$ chip input error.

# Uscita dei discriminatori (DLL)

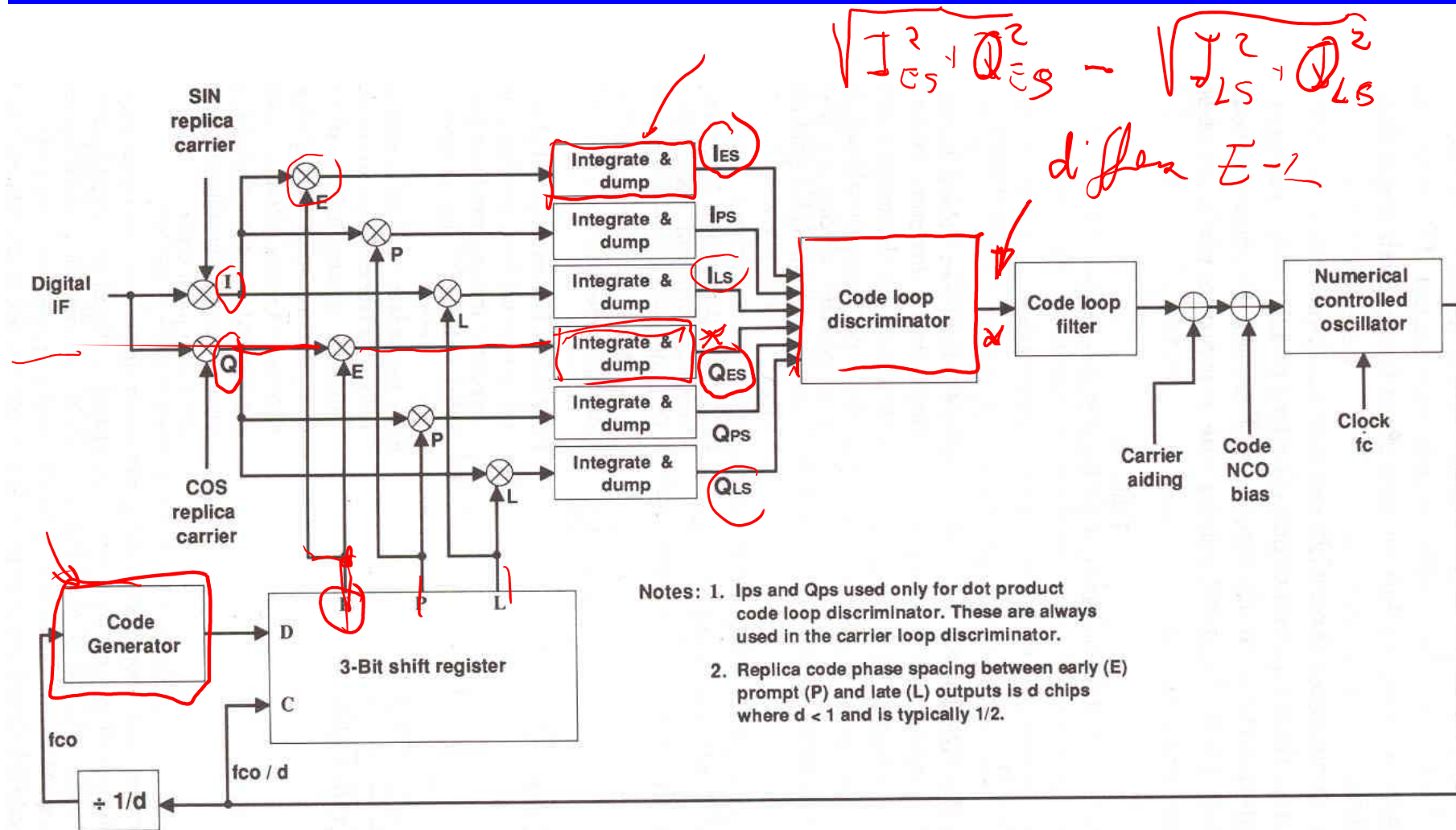


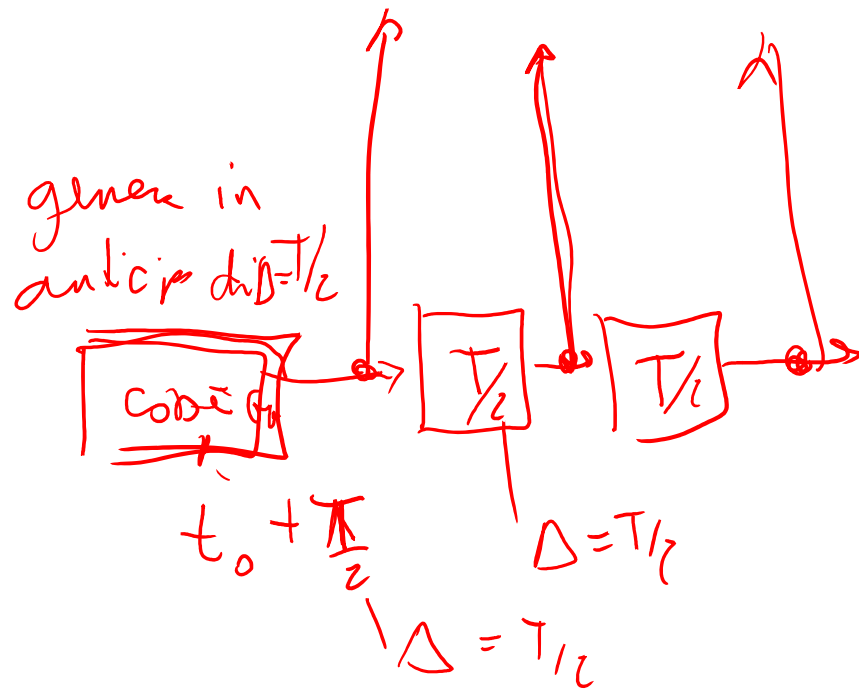


# Esempio di uscita dal discriminatore del DLL



# Schema del DLL





## Radiotecnica e Radiolocalizzazione

