

Convertitori analogico/digitali

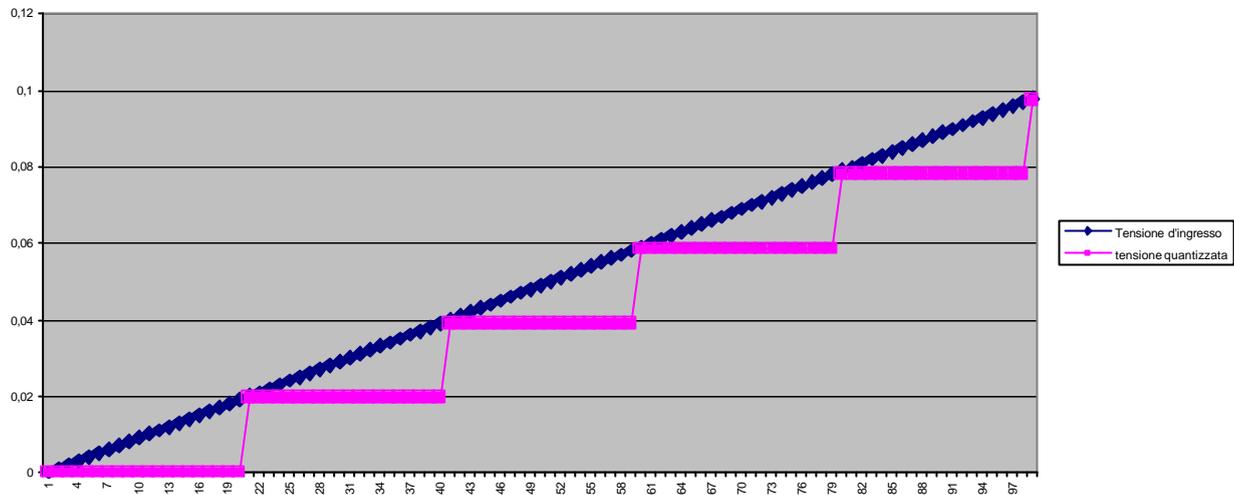
Un convertitore analogico digitale ha la funzione inversa a quella di un convertitore DAC, poiché il suo scopo è quello di permettere ad un sistema a microprocessore di acquisire informazioni su grandezze analogiche, trasformandole in stringhe di bit corrispondenti. Quindi un ADC accetta in ingresso una grandezza analogica, per esempio una tensione, e restituisce in uscita un numero espresso in forma binaria secondo un opportuno codice, che rappresenta la grandezza analogica in ingresso.

Nella conversione analogico/digitale è ineliminabile la perdita d'informazione. Infatti una grandezza analogica varia con continuità all'interno di un range quindi assume infiniti valori mentre la grandezza numerica in uscita può assumere un numero finito di valori in dipendenza del numero di bit che compongono il dato in uscita. Se, ad esempio, abbiamo un ADC ad otto bit che accetta in ingresso tensioni comprese fra 0 e 5 volt, poiché con 8 bit sono rappresentabili soltanto $2^8=256$ combinazioni diverse, si possono rappresentare soltanto 256 valori diversi di tensioni d'ingresso, valori che differiscono fra loro di un quanto Q pari a $5 \text{ volt}/256= 19,3$ millivolt circa. Ciò significa che una variazione della tensione d'ingresso inferiore al quanto non verrebbe rilevata in uscita. Il quanto viene anche detto intervallo di quantizzazione. Tale intervallo di quantizzazione o errore di quantizzazione può essere minimizzato soltanto aumentando il numero di bit che costituiscono l'uscita dell'ADC.

Supponiamo ora, sempre continuando con il nostro esempio, di fare in modo che il numero prodotto in uscita dall'ADC vari da N ad $N+1$ quando la tensione di ingresso varia fra $N*Q$ ed $(N+1)Q$.

[Vedi esempio nella cartella Excel associata.](#)

[Vedi esempio nel documento associato.](#)

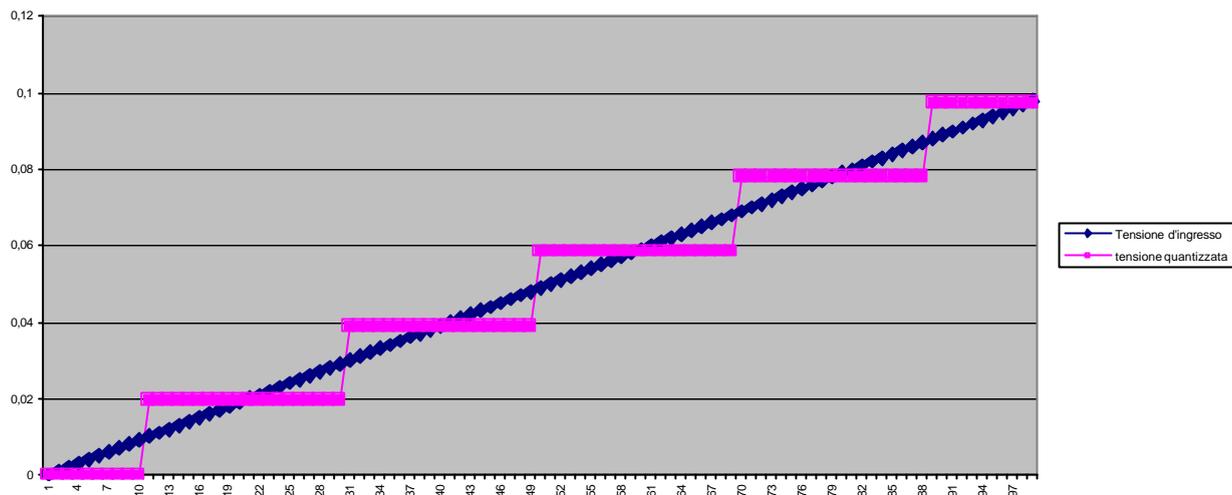


Come si può notare, l'errore massimo che si commette, confrontando la tensione quantizzata con la tensione d'ingresso è proprio pari al quanto.

Vediamo ora un secondo esempio in cui facciamo in modo che l'uscita vari da N ad $N+1$ quando la tensione d'ingresso sia compresa a metà dell'intervallo compreso fra NQ ed $(N+1)Q$.

[Vedi esempio nella cartella Excel associata.](#)

[Vedi esempio nel documento associato.](#)

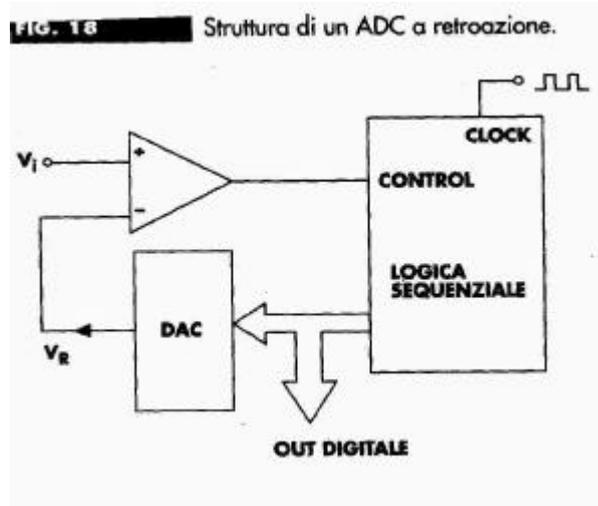


come si può verificare dal grafico e dai calcoli effettuati dal foglio elettronico si vede che l'errore massimo che si commette diventa la metà del quanto. Questa è la scelta ottimale, nel senso che permette di minimizzare l'errore di quantizzazione. . poiché con n bit si possono comporre al massimo 2^n combinazioni diverse, il numero in uscita può essere al massimo $2^n - 1$ corrispondente ad una tensione quantizzata $V = Q \cdot (2^n - 1)$. La tensione $V_{FS} = Q \cdot 2^n$ viene detta tensione di fondo scala. Nella migliore delle ipotesi l'errore di quantizzazione è pari a $Q/2 = (V_{FS}/2^n)/2 = V_{FS}/2^{n+1}$.

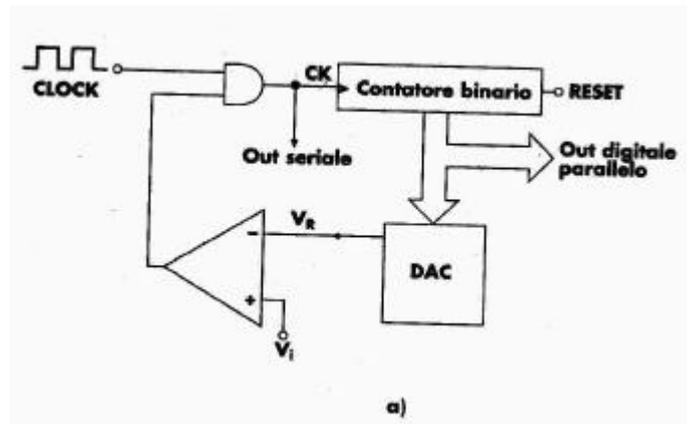
L'errore di quantizzazione si può interpretare come un rumore indesiderato che si sovrappone al nostro segnale d'ingresso. Con una serie di passaggi matematici che non riportiamo per non rischiare di offendere le vostre meravigliose menti matematiche , degne di un Renato Caccioppoli, si trova che il rapporto fra il segnale e questo rumore diventa tanto migliore quanto maggiore è il rapporto fra il valore massimo del segnale ed il valore di fondo scala dell'ADC. Questo deriva dal fatto che un segnale di piccola ampiezza rispetto al valore di fondo scala dell'ADC non ne sfrutta tutta la risoluzione. Se dobbiamo convertire un segnale, quindi, conviene

prima amplificarlo in modo da portare il suo valore massimo al massimo valore ammissibile in ingresso all'ADC. Dalla formula matematica che abbiamo tralasciato nonostante sapessimo di poter contare sulle vostre incredibili capacità matematiche universalmente riconosciute, si vede anche che il rapporto segnale-rumore migliora al crescere del numero n dei bit dell'ADC.

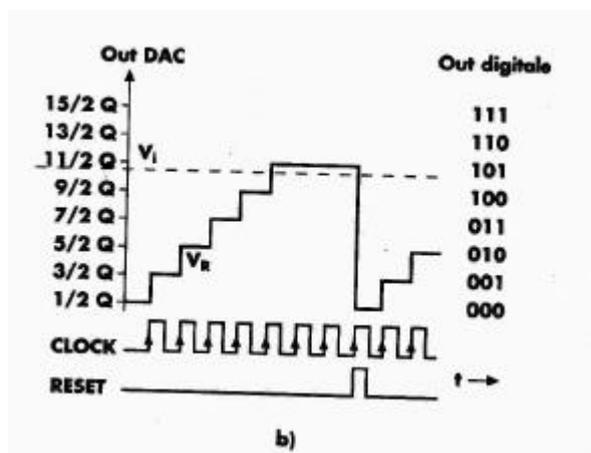
Il principio di funzionamento di un ADC è rappresentato in figura. Il blocco logica sequenziale è un blocco che è in grado di generare tutte le combinazioni possibili degli n bit dell'ADC. Queste sequenze di bit vanno in ingresso ad un DAC che genera la corrispondente tensione in uscita. Appena tale tensione diventa maggiore o uguale a quella di ingresso, l'uscita del comparatore va bassa bloccando il circuito sequenziale e congelando la stringa di bit che si voleva ottenere come risultato della conversione.



questo è lo schema di principio dei *convertitori A/D a retroazione*. L'esempio più semplice è il *counting ADC o convertitore a conteggio o a gradinata* i cui schema di principio è il seguente



In questo ADC il circuito di logica sequenziale è un contatore binario, pilotato da un clock. Ad ogni impulso di clock il contenuto del contatore binario si aggiorna . il numero contenuto nel contatore passa al DAC. Quando la gradinata prodotta dal DAC raggiunge e supera il valore di tensione V_i , l'uscita del comparatore va a zero per cui blocca a zero l'uscita della porta AND, impedendo al clock di raggiungere il contatore binario. In tal modo l'uscita del contatore binario rimane congelata. Per far partire una nuova conversione occorre resettare il contatore binario mediante il segnale di RESET.

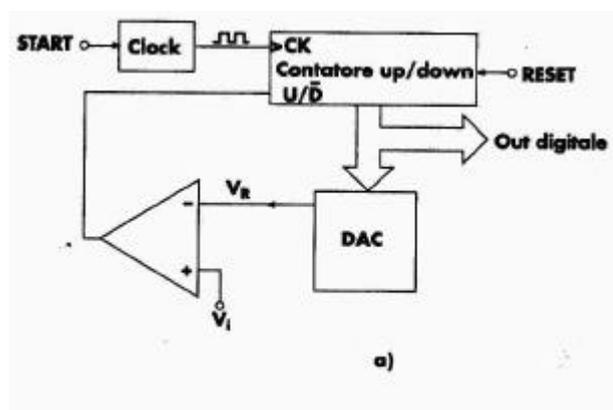


Questo tipo di convertitore presenta come problema fondamentale quello del tempo impiegato per la conversione che è elevato ed inoltre è molto variabile a seconda della tensione da convertire. Infatti tanto più è elevato il valore di tale tensione tanto

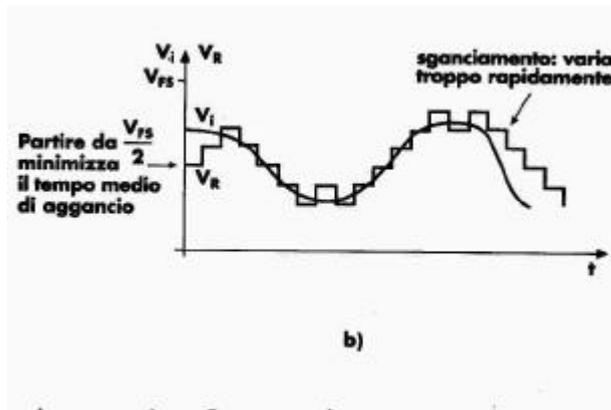
maggiore sarà il numero di periodi di clock necessari perché l'uscita del DAC giunga a superare la tensione d'ingresso. Per sicurezza chi progetta il circuito in cui è inserito l'ADC, per evitare problemi di sincronizzazione dovrà considerare prudenzialmente il tempo massimo possibile che coincide con il tempo necessario affinché il contatore giunga a contare il numero massimo possibile che è $2^n - 1$.

Servo-convertitore o tracking ADC o ADC ad inseguimento.

Se il segnale di ingresso diminuisce la sua ampiezza dopo che l'ADC ha portato a termine la sua conversione, il convertitore a gradinata non può modificare la sua uscita per seguire la diminuzione del segnale in ingresso. Per ottenere questa possibilità possiamo modificare lo schema precedente sostituendo al contatore up un contatore up/down facendo in modo che l'uscita del comparatore serva per selezionare la direzione del conteggio (vedi figura)



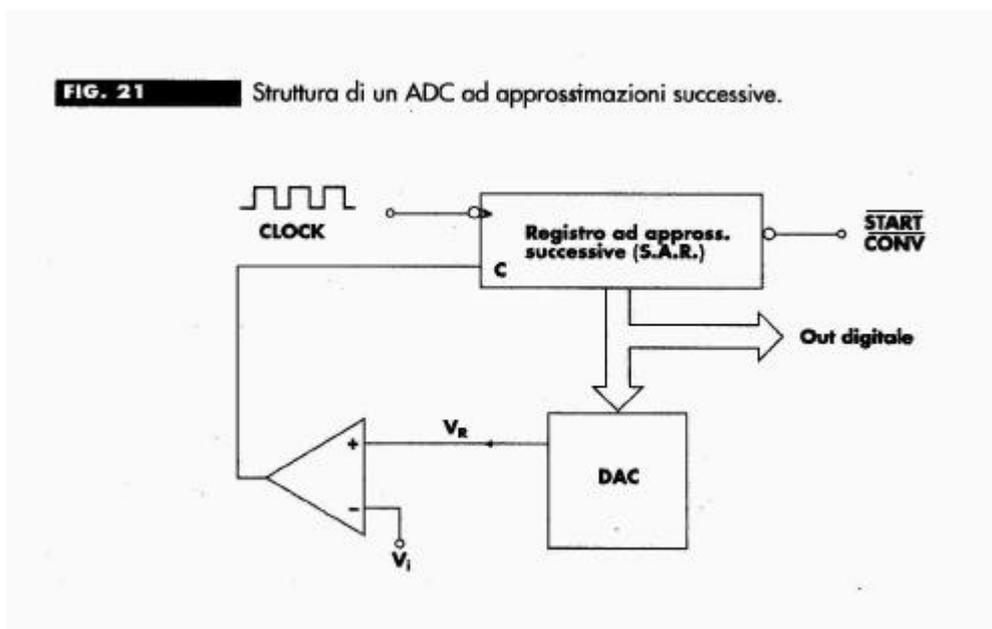
se la tensione d'ingresso è minore di quella fornita dal DAC, l'uscita del comparatore è a zero per cui il contatore funziona in modalità down e la tensione di uscita del DAC tende a diminuire. Se la tensione da convertire diventa superiore a quella del DAC, l'uscita del comparatore va ad 1 ed il contatore viene impostato in modalità up. Nella figura seguente è mostrato il processo di aggancio



se il convertitore è lasciato libero di inseguire le variazioni della tensione d'ingresso si dice che è in free running. Per questi tipi di convertitori non ha molto senso cercare di stabilire la velocità di conversione. diventa più interessante definire lo slew rate ammissibile in ingresso, cioè la velocità massima di variazione dell'ingresso affinché il convertitore sia ancora in grado di agganciarsi ad essa. Si può dimostrare che lo slew rate massimo è pari al rapporto fra quanto Q e tempo di clock.

Convertitori ad approssimazioni successive

Questa architettura consente di avere convertitori più veloci e quindi più adatti a sistemi di acquisizione dati basati su microprocessore.



il circuito sequenziale in questo caso è un registro S.A.R. o registro ad approssimazioni successive. Questo registro funziona nella logica della ricerca dicotomica. Applicata al nostro problema questa strategia si può rappresentare nel modo seguente. Noi vogliamo individuare la stringa di bit corrispondente al valore di tensione V_i presente in ingresso. Consideriamo la seguente tabella che mostra le uscite del DAC per un ADC a 8 bit.

DECIM	BINARIO								TENSIONE	Quanto
	A7	A6	A5	A4	A3	A2	A1	A0	QUANTIZZATA	
										0,019531
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	0,01953125	
2	0	0	0	0	0	0	1	0	0,0390625	
3	0	0	0	0	0	0	1	1	0,05859375	
4	0	0	0	0	0	1	0	0	0,078125	
5	0	0	0	0	0	1	0	1	0,09765625	
6	0	0	0	0	0	1	1	0	0,1171875	
7	0	0	0	0	0	1	1	1	0,13671875	
8	0	0	0	0	1	0	0	0	0,15625	
9	0	0	0	0	1	0	0	1	0,17578125	
10	0	0	0	0	1	0	1	0	0,1953125	
11	0	0	0	0	1	0	1	1	0,21484375	
12	0	0	0	0	1	1	0	0	0,234375	
13	0	0	0	0	1	1	0	1	0,25390625	
14	0	0	0	0	1	1	1	0	0,2734375	
15	0	0	0	0	1	1	1	1	0,29296875	
16	0	0	0	1	0	0	0	0	0,3125	
17	0	0	0	1	0	0	0	1	0,33203125	

18	0 0 0 1 0 0 1 0	0,3515625
19	0 0 0 1 0 0 1 1	0,37109375
20	0 0 0 1 0 1 0 0	0,390625
21	0 0 0 1 0 1 0 1	0,41015625
22	0 0 0 1 0 1 1 0	0,4296875
23	0 0 0 1 0 1 1 1	0,44921875
24	0 0 0 1 1 0 0 0	0,46875
25	0 0 0 1 1 0 0 1	0,48828125
26	0 0 0 1 1 0 1 0	0,5078125
27	0 0 0 1 1 0 1 1	0,52734375
28	0 0 0 1 1 1 0 0	0,546875
29	0 0 0 1 1 1 0 1	0,56640625
30	0 0 0 1 1 1 1 0	0,5859375
31	0 0 0 1 1 1 1 1	0,60546875
32	0 0 1 0 0 0 0 0	0,625
33	0 0 1 0 0 0 0 1	0,64453125
34	0 0 1 0 0 0 1 0	0,6640625
35	0 0 1 0 0 0 1 1	0,68359375
36	0 0 1 0 0 1 0 0	0,703125
37	0 0 1 0 0 1 0 1	0,72265625
38	0 0 1 0 0 1 1 0	0,7421875
39	0 0 1 0 0 1 1 1	0,76171875
40	0 0 1 0 1 0 0 0	0,78125
41	0 0 1 0 1 0 0 1	0,80078125
42	0 0 1 0 1 0 1 0	0,8203125
43	0 0 1 0 1 0 1 1	0,83984375
44	0 0 1 0 1 1 0 0	0,859375
45	0 0 1 0 1 1 0 1	0,87890625
46	0 0 1 0 1 1 1 0	0,8984375

47	0 0 1 0 1 1 1 1	0,91796875
48	0 0 1 1 0 0 0 0	0,9375
49	0 0 1 1 0 0 0 1	0,95703125
50	0 0 1 1 0 0 1 0	0,9765625
51	0 0 1 1 0 0 1 1	0,99609375
52	0 0 1 1 0 1 0 0	1,015625
53	0 0 1 1 0 1 0 1	1,03515625
54	0 0 1 1 0 1 1 0	1,0546875
55	0 0 1 1 0 1 1 1	1,07421875
56	0 0 1 1 1 0 0 0	1,09375
57	0 0 1 1 1 0 0 1	1,11328125
58	0 0 1 1 1 0 1 0	1,1328125
59	0 0 1 1 1 0 1 1	1,15234375
60	0 0 1 1 1 1 0 0	1,171875
61	0 0 1 1 1 1 0 1	1,19140625
62	0 0 1 1 1 1 1 0	1,2109375
63	0 0 1 1 1 1 1 1	1,23046875
64	0 1 0 0 0 0 0 0	1,25
65	0 1 0 0 0 0 0 1	1,26953125
66	0 1 0 0 0 0 1 0	1,2890625
67	0 1 0 0 0 0 1 1	1,30859375
68	0 1 0 0 0 1 0 0	1,328125
69	0 1 0 0 0 1 0 1	1,34765625
70	0 1 0 0 0 1 1 0	1,3671875
71	0 1 0 0 0 1 1 1	1,38671875
72	0 1 0 0 1 0 0 0	1,40625
73	0 1 0 0 1 0 0 1	1,42578125
74	0 1 0 0 1 0 1 0	1,4453125
75	0 1 0 0 1 0 1 1	1,46484375

76	0	1	0	0	1	1	0	0	1,484375
77	0	1	0	0	1	1	0	1	1,50390625
78	0	1	0	0	1	1	1	0	1,5234375
79	0	1	0	0	1	1	1	1	1,54296875
80	0	1	0	1	0	0	0	0	1,5625
81	0	1	0	1	0	0	0	1	1,58203125
82	0	1	0	1	0	0	1	0	1,6015625
83	0	1	0	1	0	0	1	1	1,62109375
84	0	1	0	1	0	1	0	0	1,640625
85	0	1	0	1	0	1	0	1	1,66015625
86	0	1	0	1	0	1	1	0	1,6796875
87	0	1	0	1	0	1	1	1	1,69921875
88	0	1	0	1	1	0	0	0	1,71875
89	0	1	0	1	1	0	0	1	1,73828125
90	0	1	0	1	1	0	1	0	1,7578125
91	0	1	0	1	1	0	1	1	1,77734375
92	0	1	0	1	1	1	0	0	1,796875
93	0	1	0	1	1	1	0	1	1,81640625
94	0	1	0	1	1	1	1	0	1,8359375
95	0	1	0	1	1	1	1	1	1,85546875
96	0	1	1	0	0	0	0	0	1,875
97	0	1	1	0	0	0	0	1	1,89453125
98	0	1	1	0	0	0	1	0	1,9140625
99	0	1	1	0	0	0	1	1	1,93359375
100	0	1	1	0	0	1	0	0	1,953125
101	0	1	1	0	0	1	0	1	1,97265625
102	0	1	1	0	0	1	1	0	1,9921875
103	0	1	1	0	0	1	1	1	2,01171875
104	0	1	1	0	1	0	0	0	2,03125

105	0 1 1 0 1 0 0 1	2,05078125
106	0 1 1 0 1 0 1 0	2,0703125
107	0 1 1 0 1 0 1 1	2,08984375
108	0 1 1 0 1 1 0 0	2,109375
109	0 1 1 0 1 1 0 1	2,12890625
110	0 1 1 0 1 1 1 0	2,1484375
111	0 1 1 0 1 1 1 1	2,16796875
112	0 1 1 1 0 0 0 0	2,1875
113	0 1 1 1 0 0 0 1	2,20703125
114	0 1 1 1 0 0 1 0	2,2265625
115	0 1 1 1 0 0 1 1	2,24609375
116	0 1 1 1 0 1 0 0	2,265625
117	0 1 1 1 0 1 0 1	2,28515625
118	0 1 1 1 0 1 1 0	2,3046875
119	0 1 1 1 0 1 1 1	2,32421875
120	0 1 1 1 1 0 0 0	2,34375
121	0 1 1 1 1 0 0 1	2,36328125
122	0 1 1 1 1 0 1 0	2,3828125
123	0 1 1 1 1 0 1 1	2,40234375
124	0 1 1 1 1 1 0 0	2,421875
125	0 1 1 1 1 1 0 1	2,44140625
126	0 1 1 1 1 1 1 0	2,4609375
127	0 1 1 1 1 1 1 1	2,48046875
128	1 0 0 0 0 0 0 0	2,5
129	1 0 0 0 0 0 0 1	2,51953125
130	1 0 0 0 0 0 1 0	2,5390625
131	1 0 0 0 0 0 1 1	2,55859375
132	1 0 0 0 0 1 0 0	2,578125
133	1 0 0 0 0 1 0 1	2,59765625

134	1 0 0 0 0 1 1 0	2,6171875
135	1 0 0 0 0 1 1 1	2,63671875
136	1 0 0 0 1 0 0 0	2,65625
137	1 0 0 0 1 0 0 1	2,67578125
138	1 0 0 0 1 0 1 0	2,6953125
139	1 0 0 0 1 0 1 1	2,71484375
140	1 0 0 0 1 1 0 0	2,734375
141	1 0 0 0 1 1 0 1	2,75390625
142	1 0 0 0 1 1 1 0	2,7734375
143	1 0 0 0 1 1 1 1	2,79296875
144	1 0 0 1 0 0 0 0	2,8125
145	1 0 0 1 0 0 0 1	2,83203125
146	1 0 0 1 0 0 1 0	2,8515625
147	1 0 0 1 0 0 1 1	2,87109375
148	1 0 0 1 0 1 0 0	2,890625
149	1 0 0 1 0 1 0 1	2,91015625
150	1 0 0 1 0 1 1 0	2,9296875
151	1 0 0 1 0 1 1 1	2,94921875
152	1 0 0 1 1 0 0 0	2,96875
153	1 0 0 1 1 0 0 1	2,98828125
154	1 0 0 1 1 0 1 0	3,0078125
155	1 0 0 1 1 0 1 1	3,02734375
156	1 0 0 1 1 1 0 0	3,046875
157	1 0 0 1 1 1 0 1	3,06640625
158	1 0 0 1 1 1 1 0	3,0859375
159	1 0 0 1 1 1 1 1	3,10546875
160	1 0 1 0 0 0 0 0	3,125
161	1 0 1 0 0 0 0 1	3,14453125
162	1 0 1 0 0 0 1 0	3,1640625

163	1 0 1 0 0 0 1 1	3,18359375
164	1 0 1 0 0 1 0 0	3,203125
165	1 0 1 0 0 1 0 1	3,22265625
166	1 0 1 0 0 1 1 0	3,2421875
167	1 0 1 0 0 1 1 1	3,26171875
168	1 0 1 0 1 0 0 0	3,28125
169	1 0 1 0 1 0 0 1	3,30078125
170	1 0 1 0 1 0 1 0	3,3203125
171	1 0 1 0 1 0 1 1	3,33984375
172	1 0 1 0 1 1 0 0	3,359375
173	1 0 1 0 1 1 0 1	3,37890625
174	1 0 1 0 1 1 1 0	3,3984375
175	1 0 1 0 1 1 1 1	3,41796875
176	1 0 1 1 0 0 0 0	3,4375
177	1 0 1 1 0 0 0 1	3,45703125
178	1 0 1 1 0 0 1 0	3,4765625
179	1 0 1 1 0 0 1 1	3,49609375
180	1 0 1 1 0 1 0 0	3,515625
181	1 0 1 1 0 1 0 1	3,53515625
182	1 0 1 1 0 1 1 0	3,5546875
183	1 0 1 1 0 1 1 1	3,57421875
184	1 0 1 1 1 0 0 0	3,59375
185	1 0 1 1 1 0 0 1	3,61328125
186	1 0 1 1 1 0 1 0	3,6328125
187	1 0 1 1 1 0 1 1	3,65234375
188	1 0 1 1 1 1 0 0	3,671875
189	1 0 1 1 1 1 0 1	3,69140625
190	1 0 1 1 1 1 1 0	3,7109375
191	1 0 1 1 1 1 1 1	3,73046875

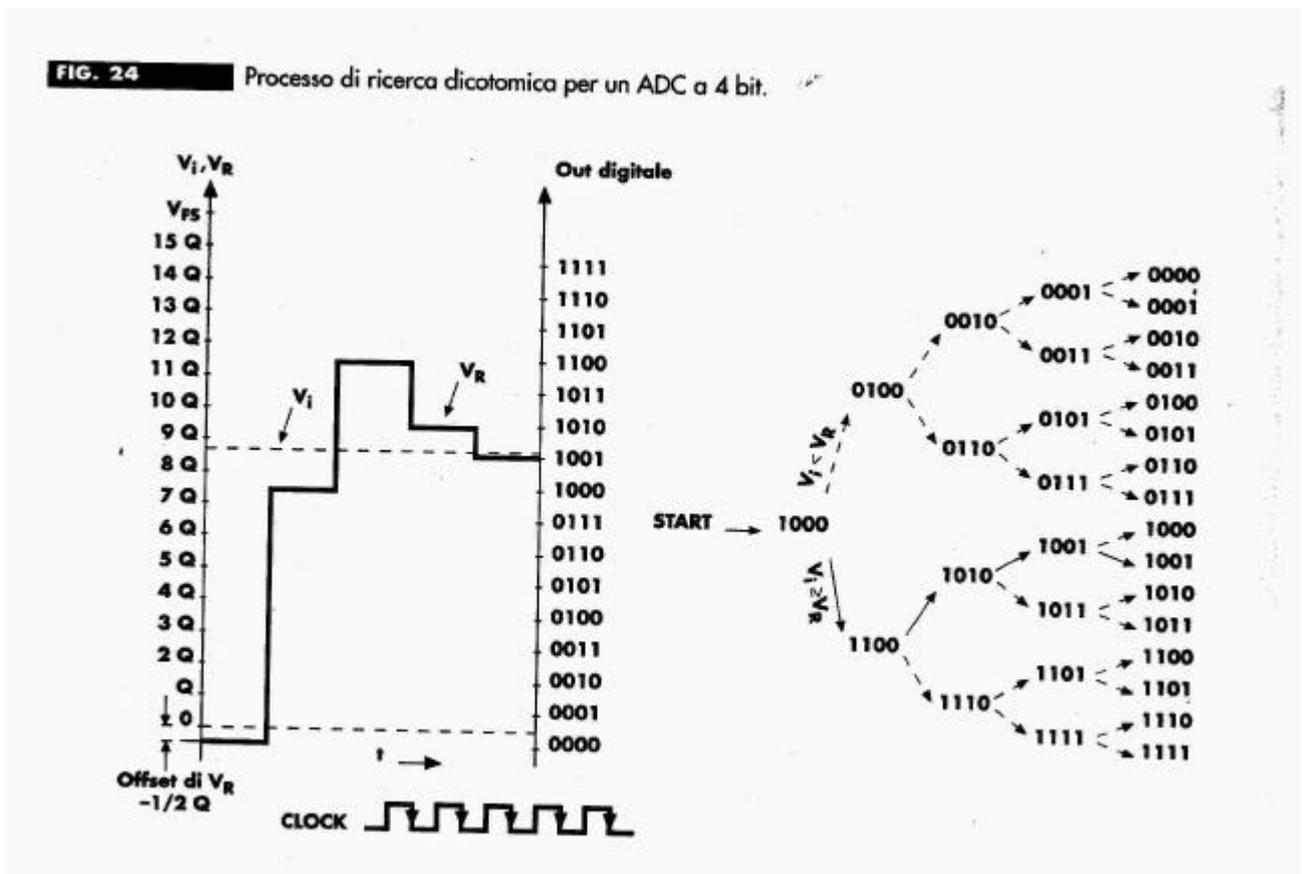
192	1	1	0	0	0	0	0	0	3,75
193	1	1	0	0	0	0	0	1	3,76953125
194	1	1	0	0	0	0	1	0	3,7890625
195	1	1	0	0	0	0	1	1	3,80859375
196	1	1	0	0	0	1	0	0	3,828125
197	1	1	0	0	0	1	0	1	3,84765625
198	1	1	0	0	0	1	1	0	3,8671875
199	1	1	0	0	0	1	1	1	3,88671875
200	1	1	0	0	1	0	0	0	3,90625
201	1	1	0	0	1	0	0	1	3,92578125
202	1	1	0	0	1	0	1	0	3,9453125
203	1	1	0	0	1	0	1	1	3,96484375
204	1	1	0	0	1	1	0	0	3,984375
205	1	1	0	0	1	1	0	1	4,00390625
206	1	1	0	0	1	1	1	0	4,0234375
207	1	1	0	0	1	1	1	1	4,04296875
208	1	1	0	1	0	0	0	0	4,0625
209	1	1	0	1	0	0	0	1	4,08203125
210	1	1	0	1	0	0	1	0	4,1015625
211	1	1	0	1	0	0	1	1	4,12109375
212	1	1	0	1	0	1	0	0	4,140625
213	1	1	0	1	0	1	0	1	4,16015625
214	1	1	0	1	0	1	1	0	4,1796875
215	1	1	0	1	0	1	1	1	4,19921875
216	1	1	0	1	1	0	0	0	4,21875
217	1	1	0	1	1	0	0	1	4,23828125
218	1	1	0	1	1	0	1	0	4,2578125
219	1	1	0	1	1	0	1	1	4,27734375
220	1	1	0	1	1	1	0	0	4,296875

221	1	1	0	1	1	1	0	1	4,31640625
222	1	1	0	1	1	1	1	0	4,3359375
223	1	1	0	1	1	1	1	1	4,35546875
224	1	1	1	0	0	0	0	0	4,375
225	1	1	1	0	0	0	0	1	4,39453125
226	1	1	1	0	0	0	1	0	4,4140625
227	1	1	1	0	0	0	1	1	4,43359375
228	1	1	1	0	0	1	0	0	4,453125
229	1	1	1	0	0	1	0	1	4,47265625
230	1	1	1	0	0	1	1	0	4,4921875
231	1	1	1	0	0	1	1	1	4,51171875
232	1	1	1	0	1	0	0	0	4,53125
233	1	1	1	0	1	0	0	1	4,55078125
234	1	1	1	0	1	0	1	0	4,5703125
235	1	1	1	0	1	0	1	1	4,58984375
236	1	1	1	0	1	1	0	0	4,609375
237	1	1	1	0	1	1	0	1	4,62890625
238	1	1	1	0	1	1	1	0	4,6484375
239	1	1	1	0	1	1	1	1	4,66796875
240	1	1	1	1	0	0	0	0	4,6875
241	1	1	1	1	0	0	0	1	4,70703125
242	1	1	1	1	0	0	1	0	4,7265625
243	1	1	1	1	0	0	1	1	4,74609375
244	1	1	1	1	0	1	0	0	4,765625
245	1	1	1	1	0	1	0	1	4,78515625
246	1	1	1	1	0	1	1	0	4,8046875
247	1	1	1	1	0	1	1	1	4,82421875
248	1	1	1	1	1	0	0	0	4,84375
249	1	1	1	1	1	0	0	1	4,86328125

250	1	1	1	1	1	0	1	0	4,8828125
251	1	1	1	1	1	0	1	1	4,90234375
252	1	1	1	1	1	1	0	0	4,921875
253	1	1	1	1	1	1	0	1	4,94140625
254	1	1	1	1	1	1	1	0	4,9609375
255	1	1	1	1	1	1	1	1	4,98046875

Data la tensione V_i d'ingresso si cerca di stabilire innanzitutto se il bit più significativo deve stare a 1 o a zero. Si pone il bit più significativo ad 1 e tutti gli altri bit a zero. Il DAC darà la tensione corrispondente e il comparatore confronterà le due tensioni. Se la tensione di uscita del DAC supera la tensione di ingresso da convertire il comparatore porta la sua uscita a 1 costringendo il SAR a porre l'MSB a 0. se la tensione di uscita del DAC è inferiore a quella di ingresso da convertire, il comparatore pone la sua uscita a 0 lasciando il MSB del SAR ad 1. a questo punto il SAR pone il bit successivo (il bit 6) ad 1 e si esegue lo stesso controllo e così via fino a giungere a determinare il bit meno significativo. Supponiamo ad esempio di voler convertire una tensione d'ingresso pari a 4,2 volt. Il SAR pone l'MSB ad 1 (il bit 7) e tutti gli altri bit a zero. Dalla tabella notiamo che ciò comporta una tensione in uscita dal DAC pari a 2,5 volt, per cui l'uscita del comparatore sarà 1. il SAR pone allora il bit 6 ad 1. dalla tabella vediamo che ciò corrisponde ad una tensione in uscita dal DAC pari a 3,75 volt. L'uscita del comparatore è ancora ad 1 per cui la scelta del bit 6 rimane inalterata. Ora il SAR prova a porre il bit 5 ad 1. ciò corrisponde ad una tensione in uscita dal DAC pari a 4,375 volt. Stavolta la tensione di uscita dal DAC è inferiore alla tensione che si vuole convertire per cui l'uscita del comparatore va a

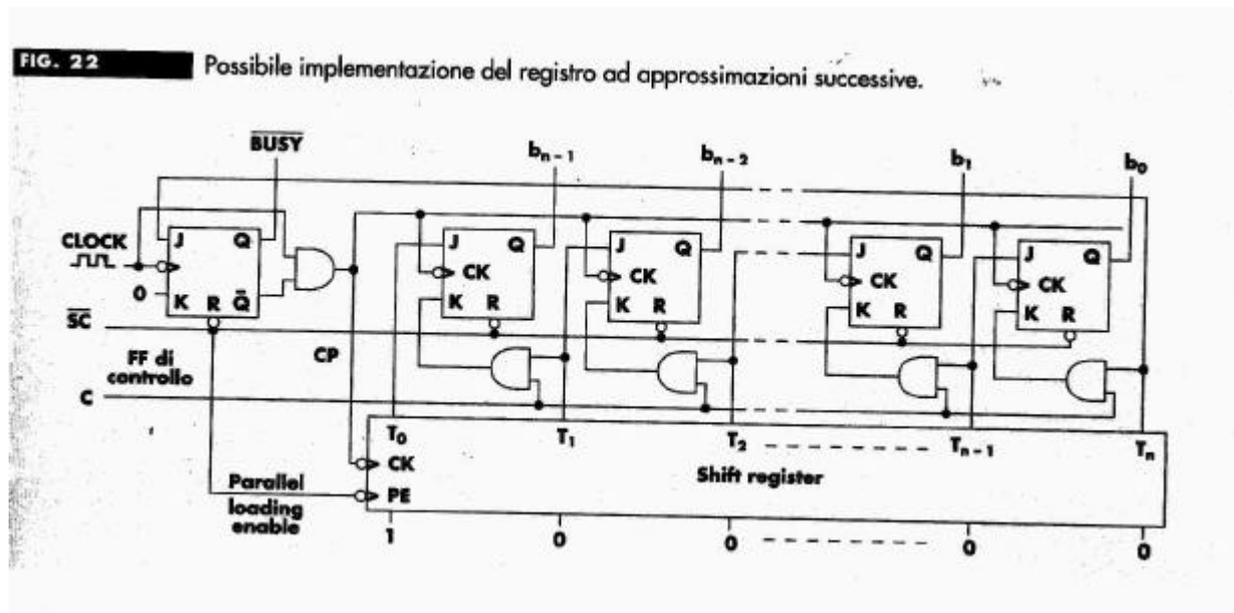
zero. Ne deriva che il SAR riporta il bit 5 a zero. Ora il SAR porta il bit 4 ad 1, ciò corrisponde alla stringa 11010000 e ad una tensione in uscita dal Dac pari a 4,0625 volt. Questa tensione è inferiore a quella d'ingresso per cui l'uscita del comparatore rimane a livello alto e il bit 4 rimane a livello logico 1. successivamente il SAR prova a porre il bit 3 a 1, la stringa 11011000 corrisponde ad una tensione in uscita dal Dac pari a 4,21875 volt. L'uscita del comparatore va bassa ed il bit verrà riportato a zero. Ciò avverrà anche per i bit 2,1 e 0. in definitiva il risultato della conversione sarà la stringa 11010000.



e' evidente il vantaggio in termini di tempi di conversione di un ADC ad approssimazioni successive rispetto ad un ADC a gradinata. Nel primo caso esso era

pari a $(2^n - 1)T_{CK}$, ora è pari soltanto a $(n - 1)T_{CK}$. vedremo più avanti che minore è il tempo di conversione e maggiore è la frequenza dei segnali che possiamo convertire.

Un esempio di realizzazione di un SAR è il seguente.

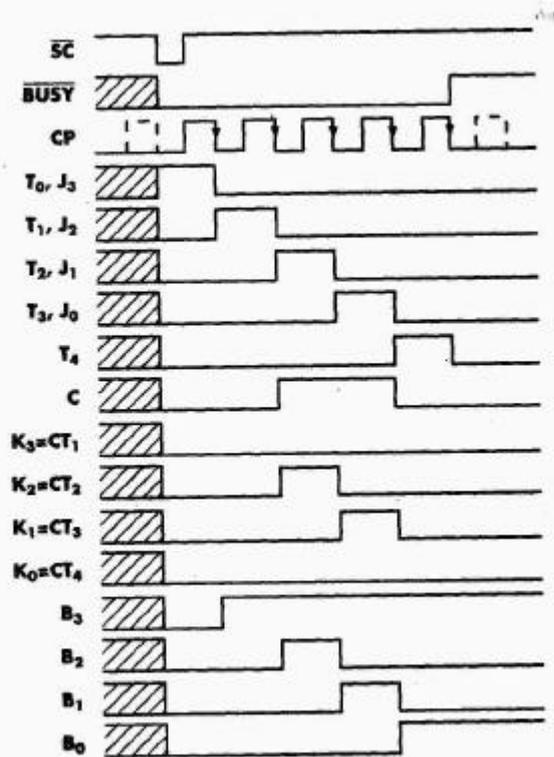


possiamo notare n flip flop di tipo JK ed un flip flop di controllo alla sinistra dello schema che decide se il clock di sistema può raggiungere o meno gli altri flip flop. Supponiamo di dare avvio alla conversione mediante il segnale SC (start of Conversion). Come si può vedere dallo schema, SC è collegato al segnale di Reset di ogni flip flop per cui resetta tutti i contatori e quindi il risultato dell'ultima conversione effettuata. Si noti che SC funge anche da segnale PE (Parallel Loading Enable) per lo shift register cioè consente che il registro venga caricato dai suoi ingressi. Ma i suoi ingressi sono bloccati in modo che il bit meno significativo venga posto ad 1 e gli altri a 0. in sostanza i flip flop vengono azzerati e lo shift register viene caricato con la stringa 00000001. poiché il flip flop di controllo è stato resettato la sua uscita Q si porta a zero e viene usata come si può vedere dallo schema per

generare un segnale BUSY attivo basso che indica, finché sta a zero, che l'ADC è impegnato nella conversione. L'uscita Q negata sta invece ad 1 e, costituendo uno degli ingressi di una porta AND consente al clock di sistema di raggiungere gli altri flip flop e lo shift register. Ora consideriamo il flip flop più significativo: J si trova ad 1 perché corrisponde al bit meno significativo in uscita dello shift register, mentre come si può notare dallo schema K è pari a 0. allora $b_{n-1}=1$ e gli altri bit si portano a zero. Ad un nuovo ciclo di clock viene attivato lo shift register per cui il bit 1 passa da T_0 a T_1 . per cui per il flip flop corrispondente al bit b_{n-2} $J=1$ e $K=0$, allora b_{n-2} viene portato ad 1. intanto il DAC ha provveduto a generare la tensione corrispondente alla stringa 10000000. ora se tale tensione risulta superiore a V_i , l'uscita del comparatore si trova ad 1, per cui $C=1$, allora per il flip flop corrispondente al bit b_{n-1} si ha che $J=0$ perché corrisponde a T_0 che per effetto dello shift è diventato 0, mentre K è il risultato della AND fra T_1 che adesso è pari ad 1 per effetto dello shift e di $C=1$,. l'uscita della And è allora pari a 1 per cui $J=0$ e $K=1$ e il flip flop viene resettato per cui b_{n-1} torna a zero. se invece la tensione V_R , corrispondente alla stringa 10000000, è inferiore alla tensione V_i , si ha $C=0$, l'uscita della AND =0, per cui $J=0$, $K=0$, e il flip flop resta nello stato precedente. Abbiamo dunque fatto in modo di variare automaticamente il valore del bit più significativo. Via via che i vari impulsi di clock fanno shiftare l'uno nello shift register lo stesso processo avverrà per ogni flip flop, per cui i vari bit verranno settati o resettati automaticamente. Quando il bit di controllo raggiunge il bit T_n , poiché questo è collegato all'ingresso J del flip flop di controllo tale flip flop viene settato, l'uscita BUSY torna ad 1 ad indicare che la

conversione è terminata, la sua uscita Q negata va a zero ed il clock non si può più propagare al SAR bloccando la conversione.

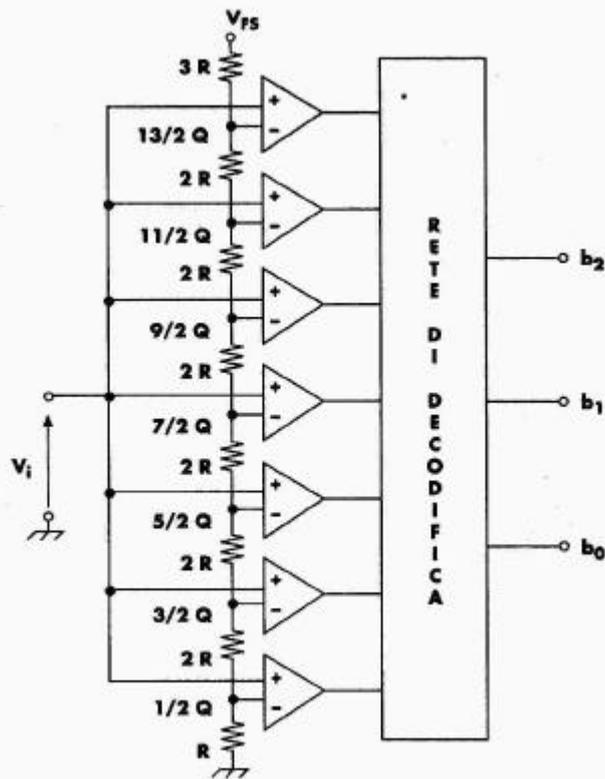
FIG. 23 Diagramma di temporizzazione di un S.A.R. a 4 bit.



IL convertitore parallelo o flash

La struttura di un convertitore parallelo è la seguente

FIG. 16 Struttura di un ADC flash a 3 bit di uscita.



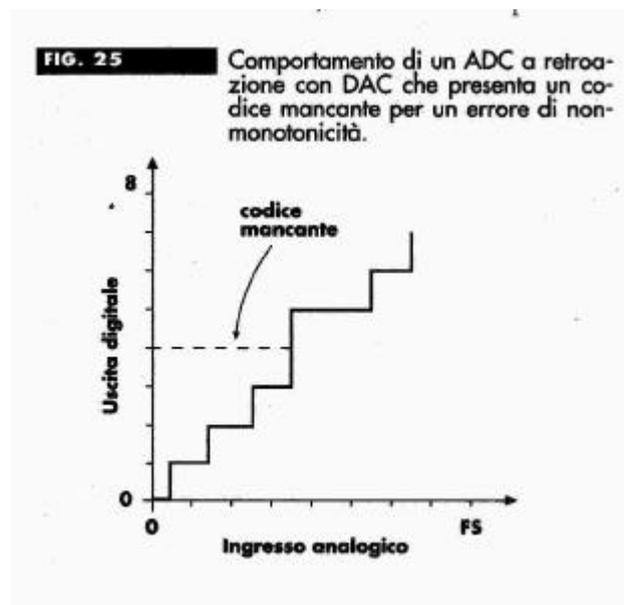
come si può vedere dalla figura abbiamo una batteria di comparatori di tensione. La tensione da convertire viene mandata in parallelo a tutti i comparatori. Ogni comparatore la confronterà con una tensione di riferimento ottenuta da un partitore resistivo. Come si può notare dalla figura si vuole fare in modo che le transizioni dei bit avvengano a metà dell'intervallo di quantizzazione.

La rete di decodifica trasforma le uscite dei comparatori in codice binario. E' evidente che un sistema di questo tipo è il più veloce immaginabile. Il difetto è dato dal fatto che all'aumentare dei bit del ADC la complessità del circuito aumenta in maniera proibitiva. Se, ad esempio, vogliamo realizzare un DAC ad 8 bit, le combinazioni possibili, cioè il numero dei livelli di discretizzazione è pari a $2^8=256$, per cui occorrerebbero 256 comparatori diversi.

Parametri e cause di errore in un ADC.

Codici mancanti

Questa possibile causa di errore si può verificare nel caso di ADC a retroazione che contengono un DAC al proprio interno. Questo errore deriva da un errore di non linearità del DAC.



a causa della non linearità del DAC (vedi figura), si ha un codice mancante, infatti se, in corrispondenza di quel codice il DAC fornisce in uscita il livello corrispondente ad un altro codice, il blocco di logica sequenziale non potrà riprodurlo in uscita.

Transizione di zero

Abbiamo visto che, allo scopo di minimizzare l'errore introdotto dalla quantizzazione si cerca di fare in modo che la transizione da una combinazione di bit ad un'altra avvenga a metà di un intervallo di quantizzazione. L'offset del comparatore o errori di zero del DAC possono fare in modo che ci sia una traslazione nel punto di transizione.

Errore di guadagno, di non linearità e di non linearità differenziale

Sono errori causati dagli errori omologhi del DAC.