



# INFORMATICA MUSICALE

*UNIVERSITA' DEGLI STUDI DI CATANIA  
DIPARTIMENTO DI MATEMATICA E INFORMATICA  
LAUREA TRIENNALE IN INFORMATICA  
A.A. 2018/19  
Prof. Filippo L.M. Milotta*

**ID: 12**

**TITOLO: “DAC⚡ADC”.**

**AUTORE: Santonocito Orazio.**

## Indice

1. Obiettivi del progetto .....	2
2. Riferimenti Bibliografici .....	3
3. Argomenti Teorici Trattati .....	4
4. Analogico & Digitale: ingredienti fondamentali.....	5
5. Conversione analogico-digitale .....	7
6. Conversione digitale-analogica .....	14

## 1. Obiettivi del progetto

Il presente progetto vuol rappresentare una possibilità di far luce sulla natura circuitale dei dispositivi di conversione di un segnale analogico in un segnale digitale e viceversa. Al giorno d'oggi siamo ormai tutti abituati al confronto diretto con dispositivi di tipo elettronico digitale, cioè con dispositivi in grado di espletare un gran numero di funzioni, tra le quali, di nostro diretto interesse, la memorizzazione e la riproduzione di un file audio. Un qualsiasi suono, in quanto nato da fenomeni fisici, rappresenta la variazione di una grandezza fisica (la pressione) nel tempo, cioè un segnale che possiamo analiticamente definire come una funzione continua del tempo ed immaginare come un vettore di transito per l'informazione da una sorgente all'ambiente circostante. Utilizzando dispositivi di trasduzione è possibile creare segnali di tensione con forme d'onda che risultano essere repliche del segnale fisico. Quindi, vista l'epoca di digitalizzazione in cui oggi viviamo, è interessante scoprire come sia possibile inserire un segnale di tensione analogico all'interno di un dispositivo digitale, che, su nostra richiesta ci restituisce il segnale fisico di partenza permettendoci di apprezzare per esempio la maestosità della "Toccata e fuga in re minore" di J.S. Bach che avevamo precedentemente registrato e memorizzato dopo una scappata a teatro!

Primo obiettivo che ci prefissiamo di raggiungere è quello di capire, anche attraverso schematizzazioni che semplificheranno la comprensione del circuito, la topologia circuitale alla base dei dispositivi DAC e ADC. Durante la trattazione verranno presentati diversi approcci di progettazione che serviranno a mettere in risalto peculiarità dei fenomeni di tipo elettrico che stanno alla base della analisi svolta. Il progetto non vuol essere una trattazione complessa dei fenomeni fisici che regolano il funzionamento di tali circuiti, trattazione per cui vengono spesi interi corsi di elettronica, ma un modo per capire quali siano i principi di funzionamento alla base della conversione.

Il secondo obiettivo che ci prefissiamo di raggiungere è quello di riuscire a stimare, mediante l'aiuto di un'analisi grafica, l'errore commesso durante questi processi di conversione. Noteremo che oltre ad un errore, sempre presente, dovuto alla natura stessa del processo di campionamento e quantizzazione di un segnale analogico, esistono altri errori dovuti alla natura fisica dei componenti circuitali utilizzati, che per quanto progettati in modo ottimale non presenteranno mai un comportamento ideale come quello pronosticato dal nostro modello.

Ultimo obiettivo, è quello di ricreare in ambiente di simulazione LTSpice un esempio di dispositivo DAC per osservare, utilizzando il tool, alcune peculiarità del processo di conversione, simulando il comportamento di alcune componenti elettroniche che ci permetteranno, anche se non perfettamente, di ricreare un modello reale per questi circuiti. La simulazione sarà utilizzata per proiettare quanto analizzato teoricamente nel mondo pratico!

## 2. Riferimenti Bibliografici

### 1) «Microelettronica» Jaeger-Blalock.

Questo testo rappresenta la fonte di riferimento per l'analisi dei circuiti di conversione analogico-digitale e digitale-analogico e per l'analisi degli errori dovuti alla conversione.

### 2) «Fondamenti di telecomunicazioni» Leon Couch.

Questo testo rappresenta la fonte di riferimento per i concetti di segnale, campionamento e quantizzazione, fondamentali per la comprensione dei dispositivi di conversione analogico-digitale e per la comprensione delle scelte effettuate in termini di frequenza e periodo di campionamento.

### 3) «Circuiti» Miano-deMagistris.

Questo testo rappresenta la fonte di riferimento per i concetti elettrotecnici di base necessari per l'analisi che abbiamo prefissato di effettuare riguardo i dispositivi di conversione.

### 3. Argomenti Teorici Trattati

- 1) Campionamento.
- 2) Quantizzazione uniforme.
- 3) Segnale analogici.
- 4) Segnali digitali.
- 5) Codifica di un segnale analogico in binario naturale.
- 6) Aspetti fondamentali sui convertitori A/D.
- 7) Errori nei convertitori A/D dovuti alla quantizzazione.
- 8) Errori nei convertitori A/D dovuti alle componenti circuitali.
- 9) Tecniche di base per la conversione A/D.
- 10) Conversione D/A.
- 11) Aspetti fondamentali sui convertitori D/A.
- 12) Errori nei convertitori D/A.
- 13) Progettazione di un dispositivo D/A.

In rosso gli argomenti trattati a lezione durante il corso di Informatica Musicale ed approfonditi nel progetto.

In nero gli argomenti di approfondimento a completamento di quanto fatto durante il corso.

## 4. Analogico & Digitale: ingredienti fondamentali.

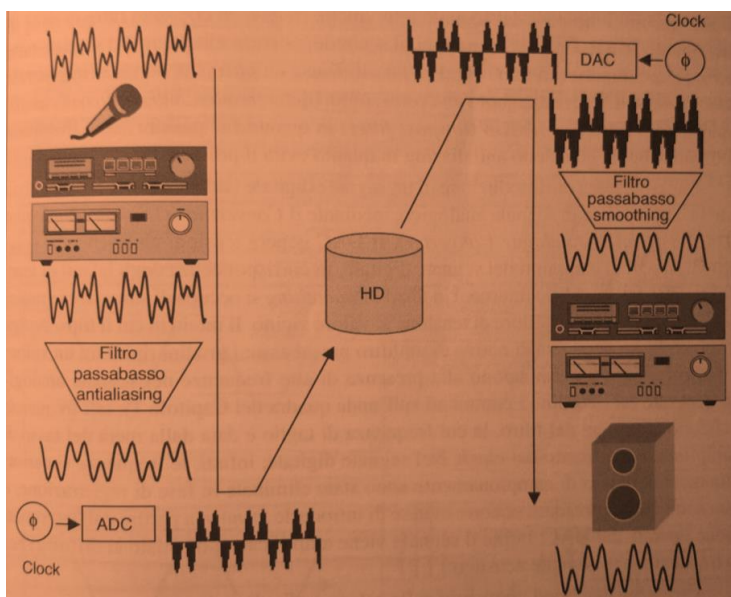
Quando parliamo di elettronica digitale, ci riferiamo al trattamento di segnali digitali binari, cioè che assumono solo due valori. Gli stati associati ad un sistema binario sono rappresentati da due simboli: 1 per il livello alto (high) e 0 per il livello basso (low). I due stati logici generalmente corrispondono a due valori di tensione distinti, che rappresentano i livelli di ampiezza alto e basso. Negli anni, i valori di tensione  $V_h$  (tensione livello high) e  $V_l$  (tensione di livello low) hanno subito diverse variazioni. Per le prime applicazioni di elettronica digitale le tensioni erano di  $\pm 5V$ , ma al giorno d'oggi, vista la migrazione verso elaboratori con processori ad altissime frequenze, per limitare la dissipazione di potenza dinamica delle porte logiche, si sono notevolmente ridotte. Per tutta la trattazione dunque un segnale digitale sarà considerato come un segnale che può assumere solo due valori di tensione: 0V sarà il livello di tensione basso e 3V il livello di tensione alto. La maggior parte dei fenomeni fisici può esser considerata di tipo analogico. Grandezze fisiche come temperatura, umidità, pressione oppure intensità luminosa possono assumere valori continui nel tempo in un intervallo limitato. Da quanto appreso nel corso, un suono è dovuto all'oscillazione delle particelle che compongono l'aria, che rappresenta un mezzo di propagazione elastico. Un suono è, dunque, riconducibile ad una variazione della pressione su una superficie, quella formata dalle molecole che compongono l'aria, ed è possibile percepirlo come tale se si dispone di una superficie di esposizione all'onda ed uno strumento in grado di apprezzarne la vibrazione. Quindi, il suono è anch'esso un segnale analogico, cioè una funzione continua in uno specifico intervallo di tempo. Se immaginiamo un tono puro, è possibile pensare al suono come una funzione sinusoidale del tempo avente un'ampiezza (o intensità) ed una frequenza. Supponendo di avere un dispositivo di trasduzione, cioè un dispositivo in grado di percepire il suono come una variazione di pressione su una specifica superficie nel tempo e trasdurlo in una funzione che rappresenta la variazione di una tensione nel tempo, troveremo un segnale di tensione che assume con continuità dei valori in uno specifico lasso di tempo. Il segnale così ottenuto rappresenta un segnale analogico in tensione. Quando parliamo di elettronica analogica, ci rifaremo al trattamento di segnali di questo tipo. Occorre comunque fare una doverosa precisazione. Un segnale digitale per poter assumere solo due valori dovrebbe esser caratterizzato da transizioni  $V_h \Rightarrow V_l$  e  $V_l \Rightarrow V_h$  infinitamente veloci, cioè a derivata infinita (analiticamente una transizione equivarrebbe ad una discontinuità di salto). Analiticamente tutto ciò è possibile ma in realtà "natura non facit saltus" dunque la parte di elettronica digitale tratta segnali approssimabili ad un segnale digitale ideale ma che dobbiamo esser consapevoli non essere segnali digitali ideali (del resto ogni componente elettronico, anche se sollecitato con segnali digitali ideali, risponderebbe sempre alle transizioni di livello con continuità).

Per passare da un segnale analogico ad un segnale digitale, che supporremo ideale, occorre un processo detto di campionamento e quantizzazione. Il campionamento è il processo secondo cui, dato un segnale analogico, preleviamo, ad intervalli di tempo regolari, il valore del segnale negli istanti di tempo considerati. Il lasso di tempo che intercorre tra un campione ed il successivo è detto periodo di campionamento. Per ogni periodo di campionamento è possibile, sapendo che la frequenza è il reciproco del periodo, associare una frequenza detta frequenza di campionamento. Non possiamo scegliere una frequenza qualsiasi di campionamento ma occorre che sia soddisfatta la condizione di Nyquist. Prima di parlare di tale condizione, che nelle analisi successive considereremo sempre soddisfatta, bisogna parlare dello spettro associato ad un segnale. Ad ogni segnale fisico è possibile associare uno spettro, che ci informa sul contenuto armonico del segnale e sull'importanza di ogni singola frequenza in termini di modulo (energia) e fase. Il range di frequenze caratteristiche di un segnale viene chiamato banda del segnale. Per segnali a banda limitata (verrà indicata con  $B$ ) la condizione di Nyquist ci assicura che è possibile ricostruire il segnale iniziale a partire dai propri campioni se è verificata la condizione secondo cui la frequenza di campionamento risulta essere maggiore di due volte la banda associata al segnale. Dunque, in prospettiva, vista la volontà di applicare la conversione in modo bidirezionale, occorre tener presente questa condizione quando andiamo

ad effettuare il campionamento di un segnale. Occorre prestare attenzione al significato di banda limitata. Dire che la banda debba essere limitata vuol dire poter individuare per il segnale una frequenza di Nyquist finita, e dunque poter associare a questo il giusto rate di campionamento.

La quantizzazione è il processo secondo cui, data una sequenza di campioni e specifici livelli di tensione (quanti) a cui viene associata una codifica, ogni singolo campione, che rappresenterà uno specifico livello di tensione, viene approssimato ad un determinato quanto, commettendo un errore (perdita di informazione) che sarà impossibile recuperare. Dunque, ogni singolo campione verrà codificato attraverso l'identificativo del quanto che ne rappresenta l'approssimazione alla fine del processo. Come visto a lezione abbiamo a disposizione due approcci alla quantizzazione: uniforme e non uniforme. Nella trattazione che seguirà nei prossimi paragrafi si farà riferimento alla quantizzazione uniforme per semplicità.

Prima di passare ai dispositivi di conversione è utile avere una panoramica complessiva del processo di registrazione e riproduzione di una traccia audio digitale, per sottolineare l'ubicazione dei DAC e ADC all'interno di questo processo.



Il processo consiste in:

- 1) Trasduzione del segnale analogico attraverso un dispositivo traduttore.
- 2) Fase di preamplificazione ed amplificazione necessarie per far sì che il segnale prodotto dal trasduttore (poco intenso) sia in grado di pilotare le componenti hardware successive necessarie per la lavorazione del segnale (una variazione molto piccola di tensione non è in grado di produrre una potenza tale da permettere il funzionamento di altre componenti hardware).
- 3) Il filtraggio antialiasing è di fondamentale

importanza per limitare la banda e dunque identificare una frequenza di Nyquist finita, che ci permetta di stabilire un rate di campionamento che non porti alla sovrapposizione delle repliche dello spettro bilatero associato al segnale, quando questo verrà campionato (secondo quanto enunciato dalle formule di Poisson).

4) Trasformazione del segnale analogico in un segnale digitale mediante un ADC (Convertitore analogico-digitale) attraverso tre fasi fondamentali: campionamento, quantizzazione e codifica.

5) Memorizzazione su un supporto fisico.

6) Trasformazione del segnale digitale nel corrispondente segnale analogico mediante un DAC (Convertitore digitale-analogico) attraverso un processo di interpolazione.

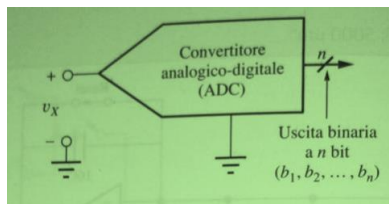
7) Il filtraggio smoothing (smussante) è di fondamentale importanza per tagliar fuori dallo spettro del segnale ricostruito contenuti armonici ad altissima frequenza che nascono dal processo di interpolazione per impulsi (per sua natura un impulso ha variazioni molto veloci che ne contraddistinguono le transizioni da alto a basso e viceversa e dunque, immaginandone l'analisi in frequenza, introduce componenti armoniche ad alta frequenza che renderebbero il suono pieno di stridii metallici).

8) Infine si passa ad una fase di amplificazione e riproduzione grazie rispettivamente a dei circuiti di amplificazione e ad altoparlanti.

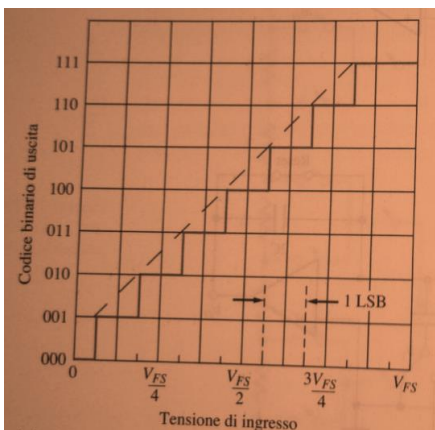
## 5. Conversione analogico-digitale.

### Errori in un convertitore A/D:

Il convertitore analogico digitale anche noto come convertitore A/D o ADC, è utilizzato per trasformare una informazione analogica in un dato binario. L'ADC preleva un segnale continuo incognito, molto spesso una tensione  $V_x$ , e la converte in un numero binario a  $n$ -bit che può essere facilmente trattato con un computer. Il numero binario rappresenta il rapporto tra la tensione incognita di ingresso e la tensione di fondo scala del convertitore  $V_{fs} = k * V_{ref}$  (di seguito il simbolo che rappresenta questo elemento).

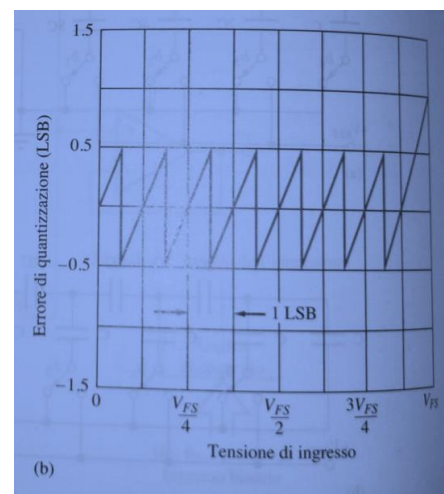


Per illustrare il comportamento di questo elemento sfruttiamone la transcaratteristica (relazione grafica che lega ingresso ed uscita associata al dispositivo) nel caso di un ADC ideale a 3 bit.



Al variare dell'ingresso da 0 a fondo scala, l'uscita digitale varia da 000 a 111 con un andamento a gradini. Tranne che per 000 e 111, l'uscita è costante per variazioni della tensione di ingresso pari a 1 LSB (Least Significant Bit: la minima variazione di tensione che può verificarsi in uscita, si ha quando il bit meno significativo LSB passa da 0 a 1) in questo caso  $V_{fs} / 8$  (in generale  $V_{fs}$  diviso il numero di quanti). Al crescere della tensione di ingresso, il codice di uscita prima sottostima l'ingresso, poi lo sovrastima. Questo errore, chiamato errore di quantizzazione è compreso tra 0, nel caso in cui l'uscita rappresenti esattamente l'ingresso e  $V_{fs}/(2*8)$  nel caso in cui l'uscita sia l'approssimazione peggiore possibile dell'ingresso nel nostro sistema.

Il grafico riportato di fianco illustra come, aumentando la tensione di ingresso a partire da zero, l'errore che commettiamo in fase di quantizzazione (di sovrastima) è sempre maggiore, questo perché ci allontaniamo dal quanto codificato con 000 e ci avviciniamo al quanto di codifica 001, ma fin tanto che non superiamo  $0V + 0.5 * LSB$ , tutti i valori di tensione di ingresso verranno "schiacciati" sul quanto 0 V di codifica 000, accrescendo l'errore. L'errore di sovrastima risulta massimo (pari a mezzo LSB) quando raggiungiamo  $V_{fs}/16$  rispetto a 000. Ci accorgiamo però che, vista la transizione a 001, in corrispondenza di  $V_{fs} / 16$  abbiamo anche l'errore massimo di sottostima rispetto a 001. Ci chiediamo:



“Come considerare il valore  $V_{fs}/16$ ? Esso rappresenta una sovrastima o una sottostima?” In realtà il problema che stiamo riscontrando non si pone poiché la transizione da 000 a 001 non avviene mai perfettamente a gradino come abbiamo messo su grafico, però non è errato nel nostro modello considerare il punto di massima sovrastima coincidente con il punto di massima sottostima. Continuando



con il nostro esempio, vediamo che l'errore di sottostima diminuisce fino a diventare zero per il valore di tensione di ingresso proprio pari  $V_{fs}/8$ , che è il valore esatto da codificare con la word 001. Ovviamente questo ragionamento può essere applicato fino a  $V_{fs}$ . Occorre notare un ultimo aspetto. Si evince dal grafico come, avvicinandoci a  $V_{fs}$ , l'errore ecceda l'errore massimo precedentemente stimato come  $V_{fs}/16$ . Questo avviene poiché, come detto all'inizio, consideriamo la prima transizione (000 => 001) dopo  $V_{fs}/16$  e non dopo  $V_{fs}/8$ , dunque la nostra transcaratteristica è traslata di  $V_{fs}/16$  verso l'asse delle ordinate. **Dal punto di vista matematico un ADC ideale dovrebbe essere progettato in modo da minimizzare l'ampiezza dell'errore di quantizzazione  $V_e$  tra la tensione di ingresso incognita e il più vicino valore quantizzato di tensione:  $V_e = |V_x - (b_1 * 2^{-1} + b_2 * 2^{-2} + \dots + b_n * 2^{-n}) * V_{fs}|$  dove  $V_x$  è la tensione di ingresso al dispositivo.**

Come tratteggiato nella prima transcaratteristica, i punti di transizione si trovano tutti su una linea retta. Tuttavia, un convertitore A/D reale presenta errori di linearità differenziale e integrale. La qualità del convertitore è valutabile attraverso la reiezione a queste "non idealità".

**ERRORE DI LINEARITA' INTEGRALE:** è una misura della deviazione dell'uscita del convertitore dalla linea retta che meglio interpola i valori della tensione di uscita. Questo errore viene di solito indicato in termini di frazioni di LSB o come percentuale della tensione di fondo scala.

**ERRORE DI LINEARITA' DIFFERENZIALE:** è la massima differenza tra l'incremento effettivo di una tensione e l'incremento ideale di 1 LSB. L'errore di linearità integrale, per un assegnato ingresso, è la somma degli errori di linearità differenziale di tutti gli ingressi precedenti fino a quello considerato.

**ERRORE DI GUADAGNO:** rappresenta la differenza tra il valore effettivo di tensione in uscita ed il valore che idealmente avremmo dovuto ottenere se tutte le transizioni fossero state interpolate da un'unica retta: quella del convertitore ideale. Come guadagno si intende la pendenza della retta passante per due punti di transizione (ideale). Un discostamento da tale retta rappresenta graficamente proprio l'errore.

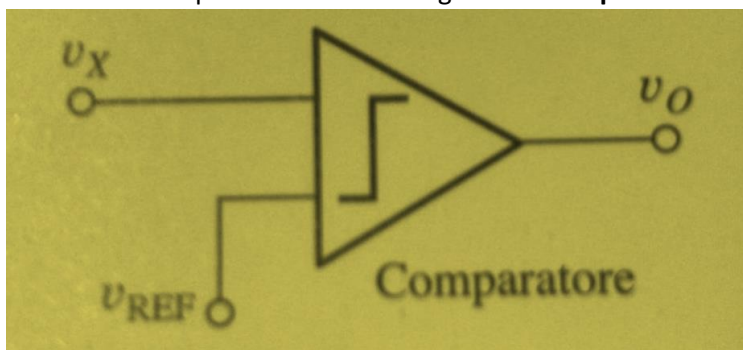
**ERRORE DI OFFSET:** è la misura del surplus di tensione che occorre applicare a partire da 0 V per ottenere la prima transizione (000 => 001).

**CODICE MANCANTE:** è una forma di errore che consiste nell'assenza di una word e dunque nella presenza di un salto di ampiezza maggiore nella transcaratteristica.

Tutti questi errori sono legati alla temperatura di lavoro del componente. Il costruttore ha l'obbligo di specificare opportuni coefficienti di temperatura per l'offset, il guadagno e la linearità.

## Tecniche di base per la conversione analogico-digitale:

Esistono molte tecniche adoperate per la conversione A/D. Tratterò due approcci molto noti per questo tipo di conversione: il convertitore "a contatore" e il convertitore "ad approssimazione successiva". Lo schema di base per la conversione riguarda il **comparatore analogico**.



### Funzionamento in breve:

La tensione in incognita  $V_x$  è connessa ad un ingresso, la tensione di riferimento, variabile nel tempo, è connessa all'altro ingresso. Se l'ingresso  $V_x$  è maggiore della tensione di riferimento l'uscita è alta e corrisponde al valore logico 1, altrimenti è bassa e corrisponde al valore logico 0. Nell'effettuare la conversione, la tensione di



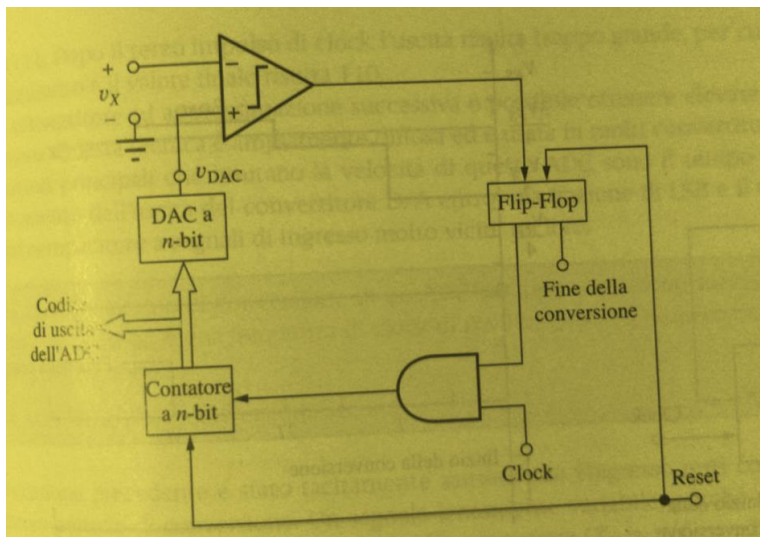
riferimento viene variata finché non raggiunge l'ingresso, entro l'errore di quantizzazione. Idealmente, il convertitore sceglie un insieme di coefficienti binari **bi** in modo tale che la differenza tra l'ingresso incognito ed il valore quantizzato sia minore o uguale a 0.5 LSB. In altre parole, **bi** viene scelto in modo che:

$$|V_x - V_{fs} \times \sum_1^n b_i \times 2^{-i}| < V_{fs} \div 2^{n+1}$$

La differenza tra il modo di funzionare dei diversi tipi di convertitori risiede nella tecnica utilizzata per variare la tensione di riferimento al fine di determinare l'insieme dei coefficienti  $\{B_i, i = 1, \dots, n\}$

## Convertitori a contatore:

Il modo più semplice di generare la tensione di riferimento è utilizzare un convertitore digitale-analogico. Si può infatti utilizzare un DAC a n-bit per generare ciascuno dei  $2^n$  valori discreti in uscita applicando l'adeguato codice di ingresso. Un modo diretto per misurare la tensione incognita di ingresso è di confrontarla in sequenza con ciascuna delle possibili uscite di un DAC. Collegando l'ingresso del DAC l'uscita di un contatore binario a n bit si realizza il confronto step-by-step con la tensione incognita. Vediamo di capirne il funzionamento:



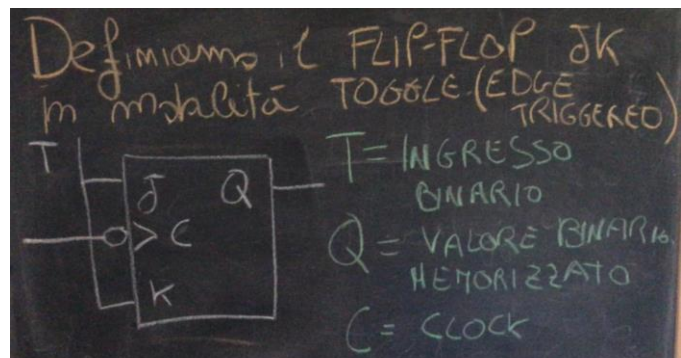
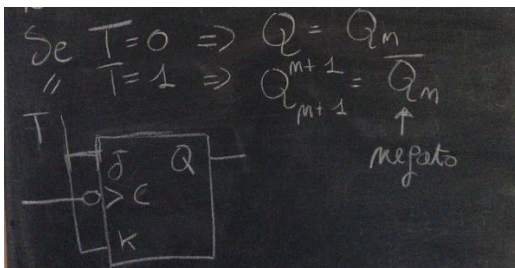
La conversione A/D inizia quando un impulso (reset) abilita il conteggio tramite il flip-flop e contemporaneamente azzerà l'uscita del contatore. Il clock è connesso al contatore (contatore di impulsi di clock) attraverso una porta AND, dunque solo quando il flip-flop (che è un elemento bistabile, cioè può memorizzare solo 0 o 1) ha in uscita 1, al contatore arriverà il treno di impulsi di clock. Infatti, se l'uscita del flip-flop è 0 avremo:  $1 \&\& 0 = 0$  e  $0 \&\& 0 = 0$ . Dunque, il contatore in input non avrà nulla da contare. Ad ogni impulso di clock (sul fronte di discesa per come tratterò il contatore successivamente) il contatore incrementa la sua uscita, che sarà una

sequenza crescente di word binarie. L'uscita del DAC, dunque, si presenterà come la transcaratteristica che abbiamo visto all'inizio della trattazione (avrà un andamento crescente a gradini). Quando l'uscita del DAC, che ricordiamo essere il valore  $V_{ref}$  del comparatore, supera il valore di ingresso incognito  $V_x$ , l'uscita del comparatore cambia stato e, tramite il flip-flop, blocca il conteggio andando a memorizzare il valore 0. Il cambiamento di stato dell'uscita del comparatore indica che la conversione è completa. A questo punto, l'uscita del contatore binario, che resterà stabile ad un valore, visto che non gli vengono forniti più impulsi di clock da contare, rappresenta il valore digitale della tensione incognita di ingresso. Nel dettaglio possiamo vedere che:

➔ Ipotezzando il segnale in ingresso alla porta AND dal flip-flop come 1, il contatore avrà segnale di reset disabilitato e inizierà da 0 a contare fino al verificarsi di uno dei seguenti eventi: o finiscono i bit a disposizione, poiché il valore di ingresso ha un'escursione eccessiva ed allora andrà a clippare la forma d'onda, o viene arrestato attraverso un clock nullo in ingresso dal comparatore poiché ha raggiunto il valore desiderato.

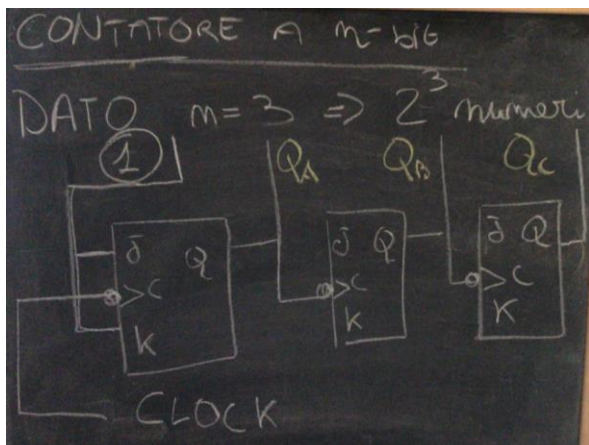


Un contatore può esser creato utilizzando più elementi bistabili (sulla base di un segnale pilota, il clock, memorizzano 0 o 1). L'elemento che utilizzerò è un flip-flop JK (che rappresentano simbolicamente le due porte di ingresso) in modalità toggle (avrà gli ingressi J e K collegati, quindi avranno sempre lo stesso valore che poniamo pari ad 1) edge triggered (cioè ogni elemento bistabile reagisce ad una transizione da alto a basso del clock). Un numero **N** di bistabili mi permette di contare fino a  $2^N$ , visto che ogni bistabile può memorizzare un bit.



Come possiamo realizzare un contatore a 3-bit?

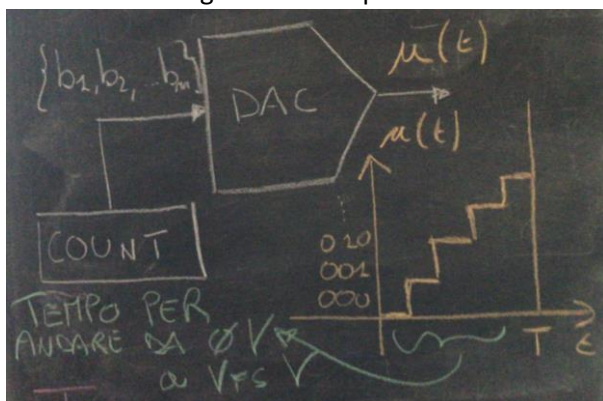




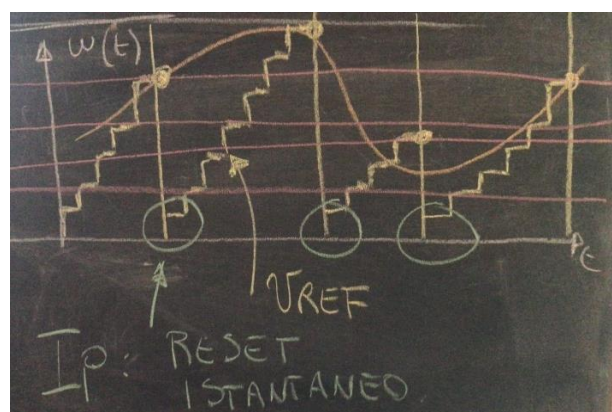
Definiti i valori  $Q_A$ ,  $Q_B$ , e  $Q_C$  che rappresentano i bit della word da creare, si può verificare che in questa connessione, dove il valore del bistabile **bi** diventa il segnale di controllo in ingresso al bistabile **bi+1**, i valori di  **$Q_C Q_B Q_A$**  andranno da 000 a 111. Consideriamo un'iterazione del processo: sul fronte di discesa del primo clock da contare (dopo un messaggio di reset non rappresentato) abbiamo che tutti gli ingressi JK valgono 1 dunque seguendo quanto rappresentato prima  **$Q_C Q_B Q_A$**  valgono 000. Sul fronte di discesa del secondo clock, avendo in ingresso al primo flip-flop 1, il valore in uscita sarà il negato del precedente e dunque varrà 1. Un passaggio da 0 a 1 di  $Q_A$  non

altera  $Q_B$  e  $Q_C$  che ricordiamo reagire solo alle transizioni alto basso. Adesso  **$Q_C Q_B Q_A = 001$** . Alla successiva transizione del clock avremo che  $Q_A$  tornerà a 0 quindi  $Q_B$  da 0  $\Rightarrow$  1 visto che il suo segnale di controllo (che non è il clock ma l'uscita del primo bistabile) passerà da 0 a 1 (questo non altera  $Q_C$  che non reagisce a transizioni basso alto). Abbiamo ottenuto la word 010. Si può verificare che il conteggio arriverà fino alla word 111 (ricordando che il bit più significativo è  $Q_C$  e non  $Q_A$ ).

Capito il funzionamento del contatore, e sapendo che un DAC opera una interpolazione per impulsi (ognuno pesato rispetto alla significatività del bit che rappresenta) avremo un segnale di tensione di riferimento in ingresso al comparatore che sarà una rampa a gradini ascendente.



A questo punto la modalità di funzionamento del comparatore ci permette di capire che l'approssimazione a gradini finisce quando  $V_{ref}$  supera la tensione analogica incognita  $V_x$  di un fattore inferiore a 0.5 LSB, perché lo stato di uscita passerà da 1 a 0 bloccando il conteggio attraverso il flip-flop, e resettando il contatore al valore iniziale. Uno specifico quanto di tempo, che dovrà esser introdotto nella figura successiva, sarà dedicato a queste operazioni ed alla lettura della word del contatore nel momento dell'arresto. Tale word rappresenta proprio il valore digitale cercato!!!



Vanno evidenziate parecchie caratteristiche di questo tipo di convertitore:

1) La lunghezza del ciclo di conversione è variabile ed è proporzionale alla tensione incognita; il massimo tempo di conversione  $T_t$  si ha in corrispondenza del valore di fondo scala del segnale di ingresso e corrisponde a  $2^N$  periodi di clock, ovvero:

$$T_t \leq (2^N) * T_c(\text{periodo di clock})$$

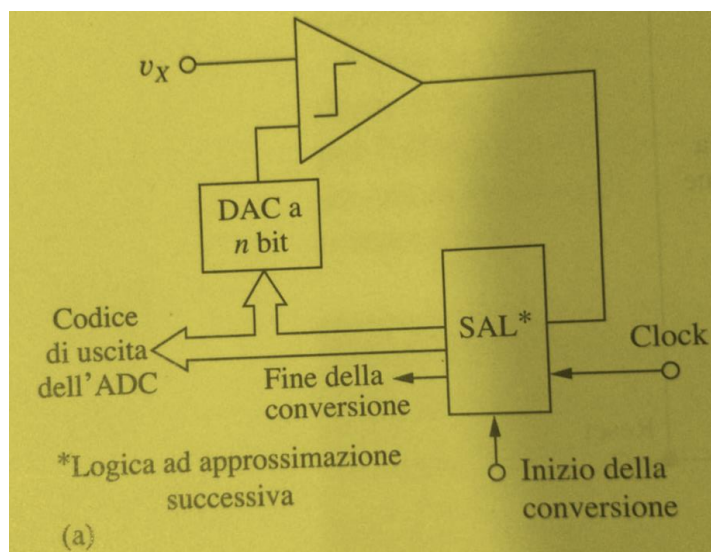
2) L'uscita binaria del contatore rappresenta la più piccola uscita del DAC che è maggiore della tensione incognita; questo valore non è necessariamente la tensione di uscita del DAC più vicina alla tensione incognita, come sarebbe auspicabile.

3) È di semplice ed economica realizzazione.

4) È relativamente lento nel processo di conversione per un'assegnata velocità del DAC.

## Convertitori ad approssimazione successiva:

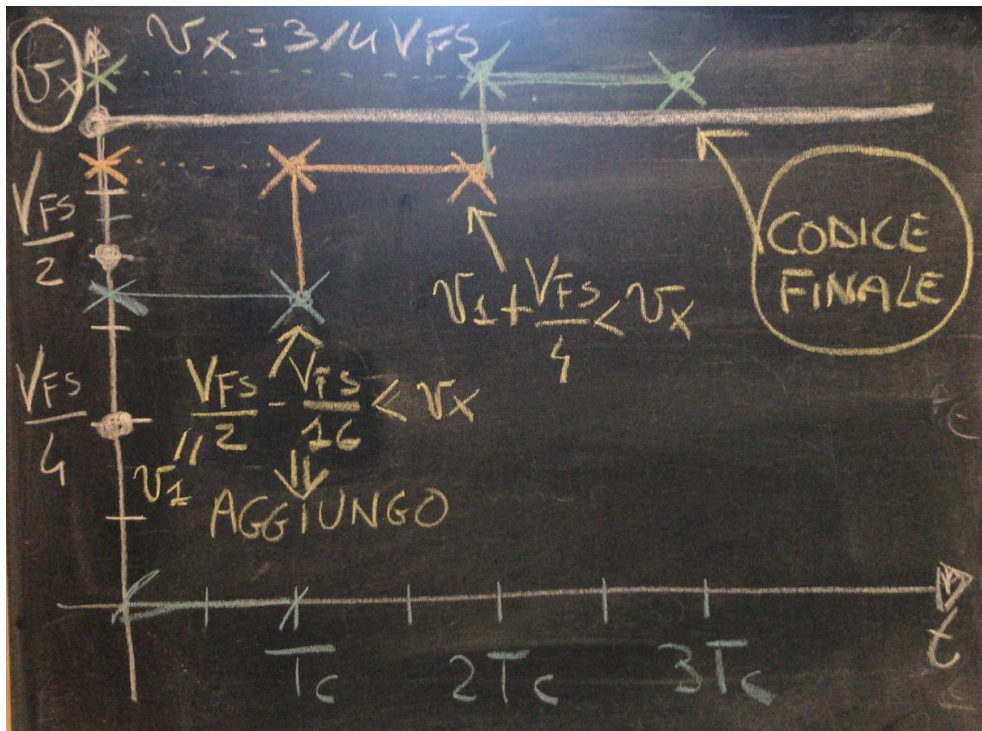
Il convertitore ad approssimazione successiva utilizza una strategia più efficiente per variare la tensione di riferimento del comparatore, richiedendo solo  $N$  impulsi di clock per effettuare una conversione a  $N$ -bit. Viene utilizzata una ricerca binaria per trovare la migliore approssimazione della tensione di ingresso  $V_x$ . Consideriamo un esempio pratico con 3 bit (8 livelli di quantizzazione) e con una tensione di offset iniziale pari a  $V_{fs}/16$  (come abbiamo visto nella trattazione dell'errore, l'offset è ricorrente quindi consideriamolo nella trattazione) cioè considereremo tutte le tensioni di uscita dal DAC con un valore pari al valore della frazione della tensione di offset che rappresentano, meno un fattore  $V_{fs}/16$ . Se l'offset crea confusione possiamo analizzare il sistema non considerandolo e quindi facendo un'analisi uguale a quella che seguirà senza considerare il fattore additivo (o sottrattivo) di 1 LSB.



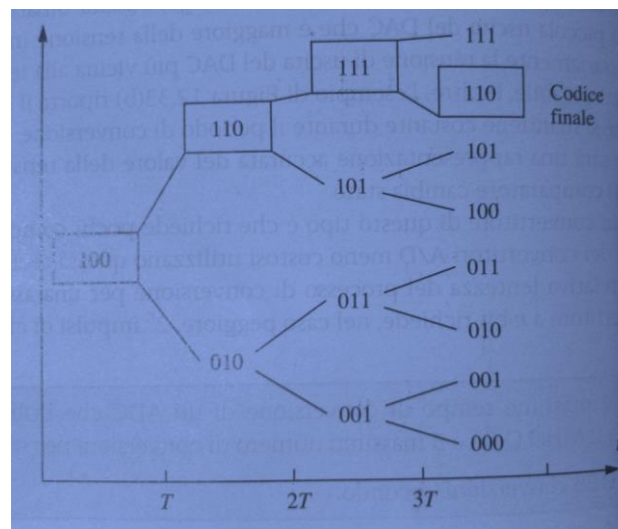
Dopo il segnale di inizio conversione, un circuito logico fissa l'uscita del DAC a  $(V_{fs}/2 - V_{fs}/16)$  e, dopo un tempo di assestamento, controlla l'uscita del comparatore (stiamo considerando come nel caso precedente 3 bit e 0.5 LSB di offset). Al successivo impulso di clock l'uscita del DAC viene incrementata di  $V_{fs}/4$  se l'uscita del comparatore era 1 o decrementata di  $V_{fs}/4$  se l'uscita del comparatore era 0. Viene quindi controllata di nuovo l'uscita del comparatore e il successivo impulso di clock provoca un incremento o un decremento del DAC di  $V_{fs}/8$ . Viene poi eseguito un terzo confronto. L'uscita binaria finale resta immutata se  $V_x$  è maggiore dell'uscita finale del DAC o diminuita di 1 LSB in caso contrario.

La conversione è quindi completa alla fine del terzo impulso di clock, per il convertitore a tre bit, o alla fine di  $N$  impulsi di clock per un convertitore a  $N$ -bit.

Schematicamente quello che facciamo, per quantizzare e codificare un valore di  $V_x$  (da considerare come un valore della tensione di ingresso al comparatore in uno specifico istante, quello in cui preleviamo il campione da  $V_x$ ) possiamo rappresentarlo come una ricerca algebricamente additiva (andiamo sempre a sommare o a sottrarre al valore della stima  $i$ -esima il valore della stima  $i$ -esima dimezzato per calcolare la stima  $(i+1)$ -esima, fermandoci alla stima  $N$ -esima, con  $N$  numero di bit della codifica, cioè fermandoci al valore algebricamente additivo di 1 LSB) che dopo un numero prestabilito di clock raggiunge un valore stabile di uscita che rappresenta il valore cercato:



Che in termini di word codificate significa:



Possiamo notare che la presenza dell'offset è poi considerata nel valore finale trovato.

Anche per questo convertitore possiamo evidenziare alcune caratteristiche:

1) È possibile ottenere elevate velocità di conversione. Questa tecnica è ampiamente diffusa ed è usata in molti convertitori a 8 e 16 bit. I fattori principali che limitano la velocità di questo ADC sono il tempo richiesto per l'assestamento dell'uscita del convertitore D/A entro una frazione di LSB e il tempo di risposta del comparatore a segnali di ingresso molto vicini tra loro (visto che lavora con differenze tra le tensioni ai suoi morsetti si intende per differenze molto piccole).

2) Un segnale  $V_x$  variabile deve soddisfare una certa condizione affinché possa essere efficacemente convertito: non deve variare di più di 0.5 LSB durante l'intervallo di conversione ( $T_t = N * T_c$ ). Questo ci permette di limitare l'errore di stima a  $0.5 \text{ LSB} = V_{fs} / (2^{(n+1)})$ .



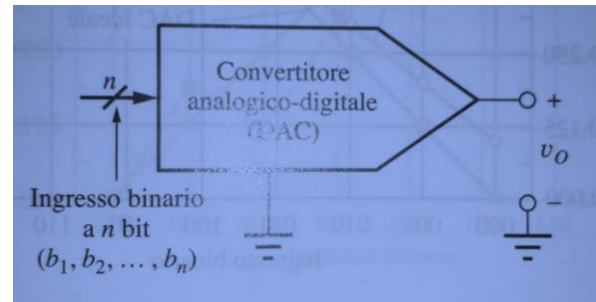
## 6. Conversione digitale-analogica.

Il convertitore digitale-analogico spesso indicato come convertitore D/A o DAC, è l'interfaccia tra i segnali digitali ed i segnali continui. Il convertitore D/A accetta in ingresso una informazione digitale, spesso in forma binaria, e genera in uscita una tensione o una corrente che può essere usata per un controllo elettronico o per visualizzare informazioni.

### Aspetti fondamentali dei convertitori D/A:

Nel DAC una parola binaria di  $n$ -bit ( $b_1, b_1, \dots, b_n$ ) viene combinata con una tensione di riferimento  $V_{ref}$  per ottenere l'uscita analogica. Assumendo di considerare una conversione in un segnale di tensione, l'ingresso digitale è considerato una frazione binaria.

Assumendo un'uscita in tensione si ha:

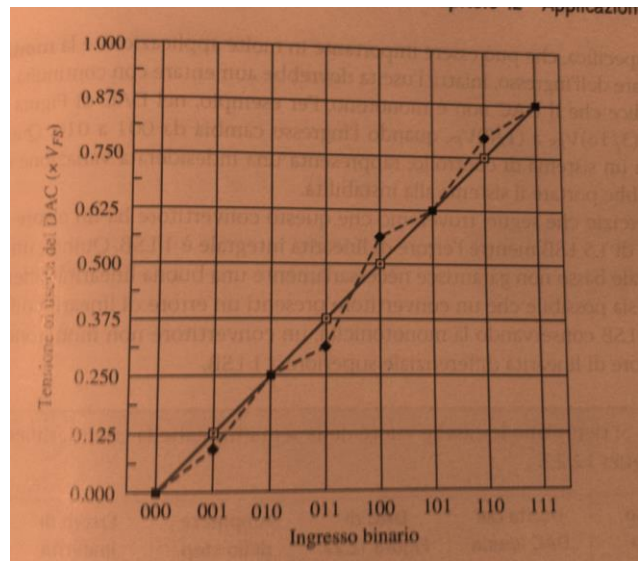
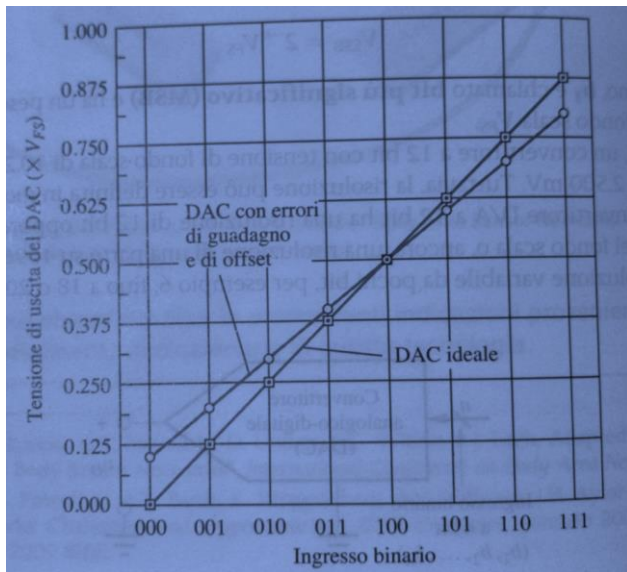


$$v_O = V_{FS}(b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}) + V_{OS} \quad \text{per } b_i \in \{1, 0\}$$

$V_{OS}$  rappresenta la tensione di offset, cioè la tensione del convertitore che compare in uscita quando l'ingresso digitale è nullo. Come già visto per l'ADC possiamo definire la più piccola variazione di tensione all'uscita del DAC che si ha quando il bit meno significativo (LSB)  $b_n$  passa da 0 a 1. Questa variazione di tensione viene indicata come risoluzione del convertitore ed è data da:  $V_{lsb} = (2^{-n}) * V_{fs}$ . All'altro estremo,  $b_1$  è chiamato bit più significativo (MSB) e ha peso pari a metà della tensione di fondo scala  $V_{fs}$ .

### Errori in un convertitore A/D:

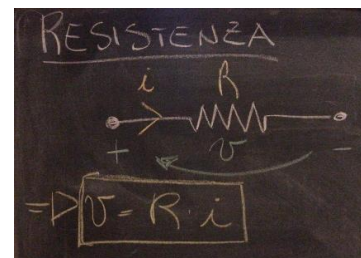
Nella ricostruzione del segnale analogico gli errori che si presentano sono dello stesso tipo di quelli già visti in precedenza. Questo ci permette di capire che questi errori sono dovuti alla natura delle componenti elettroniche e che non possono essere azzerati, diciamo che "ci si deve convivere". Facendo un'analisi più accurata si presentano: **errori di guadagno**, intesi come la deviazione della pendenza della caratteristica del convertitore reale rispetto a quello ideale; **errore di offset**, menzionato ormai più e più volte; **errore di linearità integrale**, inteso come una misura della deviazione dell'uscita del convertitore dalla linea retta che meglio interpola i valori della tensione di uscita; **errore di linearità differenziale**, che consiste nella massima differenza tra l'incremento effettivo di tensione e l'incremento ideale di 1 LSB. Di seguito i grafici che riproducono tali errori (notare come questi siano transcaratteristiche esattamente simmetriche rispetto a quelle precedentemente trattate e come il concetto di errore a cui facciamo riferimento sia del tutto analogo seppur considerando questa volta un ingresso binario ed un'uscita interpolata a impulsi).



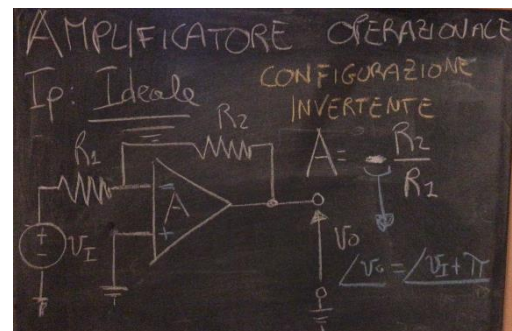
## Tecniche di base per la conversione digitale-analogica:

Per illustrare i principi di funzionamento dei convertitori DAC, andremo a capire il funzionamento di due realizzazioni circuitali di base e molto conosciute. Le implementazioni di riferimento saranno dunque: DAC realizzato a resistori scalati e DAC realizzato con schema R-2R (che sarà quello utilizzato nella simulazione finale). La scelta di questi due approcci si basa su un vantaggio: andremo a trattare attraverso una rete "lineare" (tra virgolette perché la rete che costituisce l'amplificatore internamente non lo è), quindi senza appellarci a trasformate o equazioni differenziali, il processo di conversione in esame. Gli elementi da conoscere per la comprensione di questi due circuiti saranno i seguenti:

➔ In un modello circuitale a parametri concentrati un resistore lo trattiamo in modo molto semplice andando ad utilizzare la legge di Ohm che ci dice che la tensione ai capi di una resistenza è pari al prodotto del valore di tale resistenza per la corrente che la attraversa (se i segni di tensione e corrente sono orientati come in figura). Il resistore è uno degli elementi base che andremo ad utilizzare.

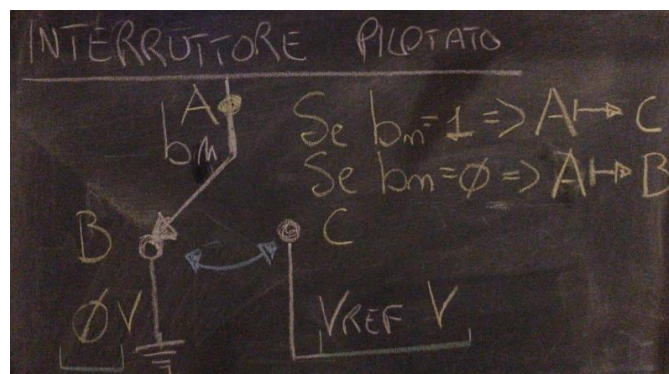


➔ Il secondo componente fondamentale è l'amplificatore operazionale, che considereremo ideale. Data una tensione di ingresso  $V_i$  questo restituisce in uscita  $V_o$  amplificata di un coefficiente  $R_2/R_1$  in opposizione di fase rispetto a  $V_i$  (il guadagno  $A$ ). Viene detto in configurazione invertente poiché infatti alimentiamo al morsetto "-" invertente. L'appellativo invertente al morsetto è dovuto al fatto che dato un segnale  $V_i$ ,  $V_o$  sarà sempre in opposizione di fase rispetto a  $V_i$ .





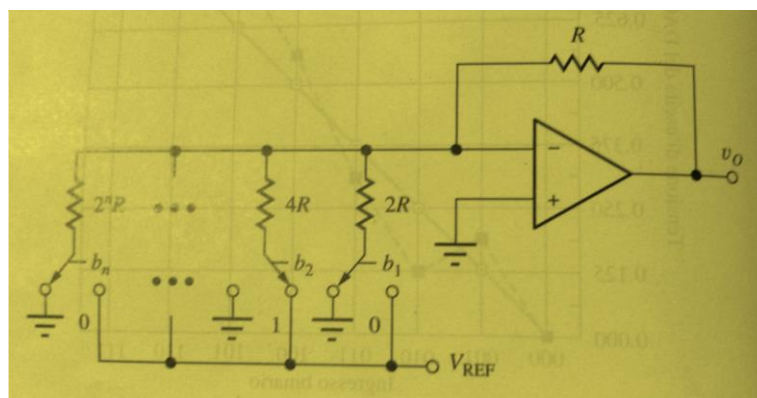
→ L'ultimo componente utilizzato è un interruttore. Questo ha lo scopo di collegare due sotto-rami di circuito differenti sulla base del valore di pilotaggio, che nel nostro caso sarà un bit. Un interruttore di questo tipo può esser implementato grazie a svariate componenti elettroniche MOSFET, BJT, ecc...



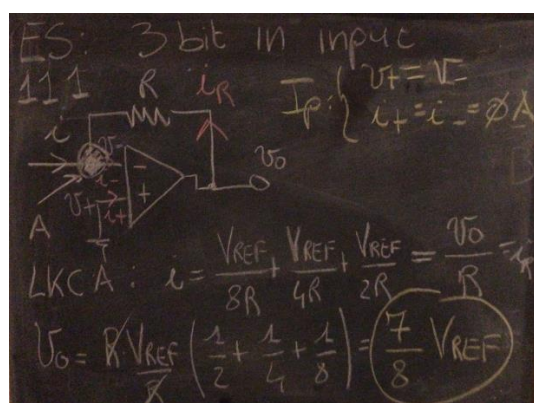
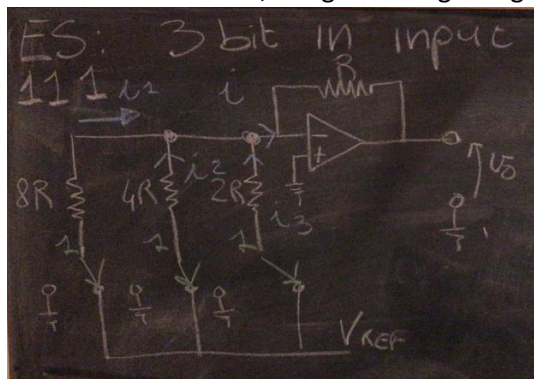
## DAC a resistori scalati:

L'ingresso binario  $\{b_1, b_2, \dots, b_n\}$  controlla gli interruttori, un 1 indica che la resistenza è connessa a  $V_{ref}$  mentre 0 indica che è connessa al potenziale nullo. I resistori adiacenti sono scalati di un fattore due, in modo che ciascuno di essi fornisce il corretto contributo all'uscita:

$$V_o = (b_1/2 + b_2/2^2 + \dots + b_n/2^n) * V_{ref}$$



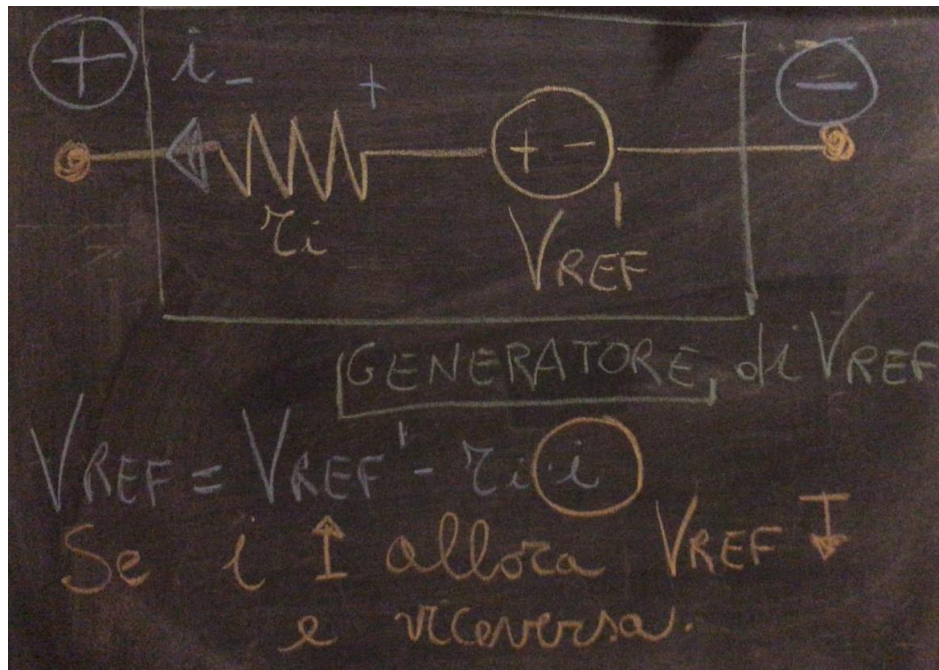
Quando i rapporti tra resistori non sono mantenuti in maniera esatta, il convertitore presenta errori di linearità differenziale, integrale e di guadagno. Ecco per linee generali il funzionamento.



Prendiamo sempre in esame word da tre bit, e vediamo quanto vale l'uscita quando abbiamo gli interruttori tutti su  $V_{ref}$  vista la parola in ingresso 111. Per analizzare il circuito, utilizziamo il modello ideale dell'amplificatore. Dunque, valgono sempre due ipotesi: il potenziale ai due morsetti di ingresso è uguale (corto circuito virtuale); le correnti entranti sono nulle (vedere immagine successiva). Sotto queste ipotesi è facile verificare quanto valga  $i$ , intesa come somma di  $i_1$ ,  $i_2$  e  $i_3$  che sono immediatamente identificabili. Facendo una legge di Kirchhoff al nodo indicato con A ci si accorge immediatamente che la corrente che scorre su  $R$  debba esser uguale a  $i$  (il nodo A non può accumulare carica) e dall'uguaglianza otteniamo la relazione ingresso uscita tale da mostrarci che a meno di un LSB l'uscita assume il valore di  $V_{ref}$ . Questa analisi, è valida in questi termini ipotizzando l'amplificatore operazionale IDEALE, dunque considerando valide le ipotesi in giallo in figura. Analizzando la configurazione in un ambiente di simulazione sarà evidente una piccola discordanza dei valori di  $V_o$  ottenuti rispetto a questo modello, dovuta alle non idealità che stiamo trascurando.

Con questo tipo di realizzazione si presentano diversi problemi. La difficoltà principale è legata al mantenimento degli esatti rapporti tra i resistori in un ampio intervallo di valori. Bisogna, inoltre, osservare

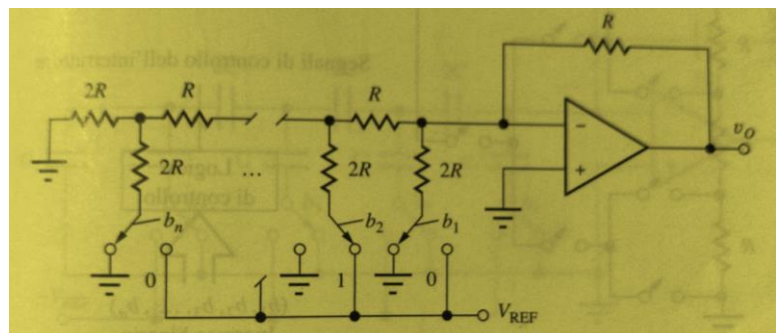
che la corrente erogata dal generatore di ingresso  $V_{ref}$  dipende dalla configurazione binaria di ingresso (cioè dal numero di resistori a cui viene collegata la linea che cambia di word in word grazie agli interruttori). Questa variazione di corrente erogata provoca una diversa caduta di tensione sulla resistenza interna del generatore, producendo una variazione della tensione di riferimento che può condurre a errori, dipendenti dai dati in ingresso, spesso chiamati **errori di sovrapposizione**.



## DAC con rete R-2R:

Diamo un'occhiata al funzionamento generale, che, una volta capito l'approccio precedentemente visto, risulta essere di più facile comprensione visto che le componenti utilizzate sono le medesime.

La rete R-2R elimina la necessità di un ampio intervallo di valori di resistori. È adatta a una realizzazione integrata in quanto chiede il controllo del rapporto tra due soli resistori. È possibile dimostrare, applicando il teorema di Thevenin passo dopo passo da sinistra verso destra lungo i nodi della rete,



che il contributo di ciascun bit si riduce di un fattore due andando dal MSB a LSB. Anche in questo approccio c'è una dipendenza di  $V_{ref}$  dall'ingresso binario, dunque possiamo avere errori di sovrapposizione. Senza entrare in merito al funzionamento del teorema di Thevenin, analizzando la configurazione 000 è evidente che la resistenza vista dall'amplificatore operazionale è proprio pari a  $R$  (basta eseguire una serie di equivalenze parallelo e serie per ottenere questo risultato). In realtà, in modo così immediato, abbiamo applicato proprio una parte del teorema di Thevenin per trovare la resistenza equivalente vista dall'amplificatore. Detto ciò, è evidente che l'amplificatore non introduce un guadagno incrementale ma pari a 1. Potremmo verificare per ogni configurazione di bit che la resistenza vista dall'amplificatore (la  $R_1$  della figura esplicativa di tale componente) è sempre la stessa, a cambiare è la tensione  $V_i$  che applichiamo in ingresso.

## Passiamo a LTSPICE:

È possibile attraverso alcuni stratagemmi riuscire in maniera semplice a simulare un DAC con rete R-2R. Per chi non sapesse cosa sia LTSpice, in breve possiamo descriverlo come un ambiente di simulazione di componenti circuitali con specifiche caratteristiche (del tutto analoghe a quelle di acquisto fornite nel mondo reale) di funzionamento in grado di simulare un circuito in termini di analisi transitoria e risposta in frequenza (mediante la rappresentazione in Diagrammi di Bode). Questo tool permette di approcciare alla costruzione del circuito desiderato mediante o l'utilizzo di un'interfaccia grafica o l'utilizzo di script (file con estensione ".cir") contenenti, secondo una specifica sintassi, le direttive per la generazione automatica del circuito desiderato.

**Modus operandi:** il primo passo da compiere è creare una rete R-2R. Per creare il circuito (mostrerò sia la creazione da interfaccia grafica sia da codice) occorre scegliere per prima cosa le componenti da utilizzare. Come amplificatore operazionale ho scelto il modello OPAMP 8092 dal catalogo degli amplificatori offerti per comodità personale vista la mia conoscenza pregressa del datasheet (che è comunque consultabile per ogni componente ed informa su tutto ciò che c'è da sapere per utilizzarlo correttamente). Per quanto riguarda la rete di interruptori possiamo creare una modifica al circuito per far sì di non complicare la rete ma ottenere lo stesso risultato. Infatti, potendo utilizzare generatori di onde quadre, una sequenza binaria in ingresso ai resistori può esser ricreata andando a sfasare nel tempo onde quadre di opportuno periodo. Completata la costruzione della rete confronteremo i risultati con quanto visto prima. Per dimostrare la dipendenza di questi dispositivi dalla profondità di bit utilizzata, genereremo due DAC identici ma uno a 4 bit ed uno a 8 bit e li metteremo a confronto. Andremo ad utilizzare un range dinamico di 0V-3V.

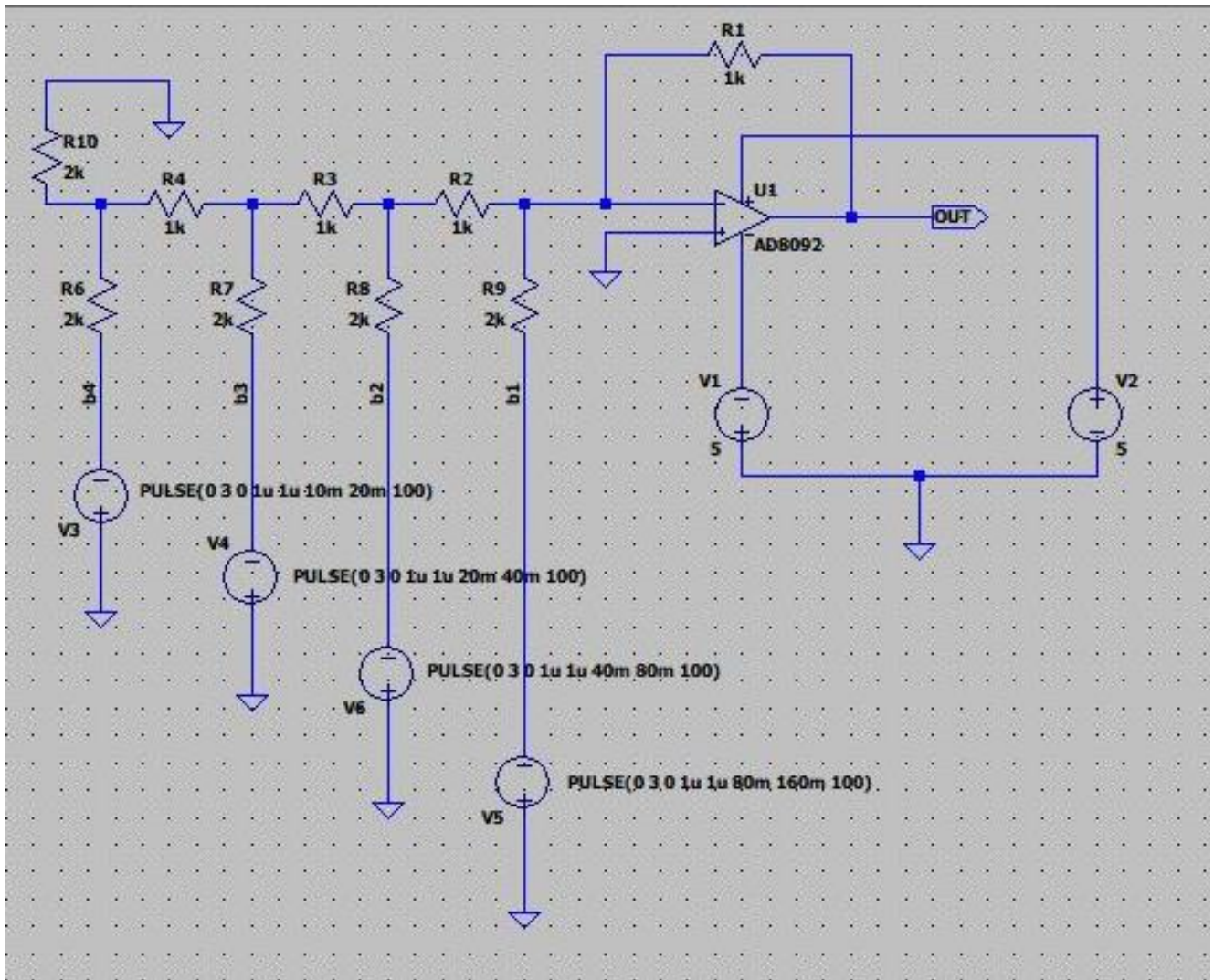
**Implementazione su LTSpice di un DAC a 4 bit (interfaccia e mediante netlist):**

*Netlist:*

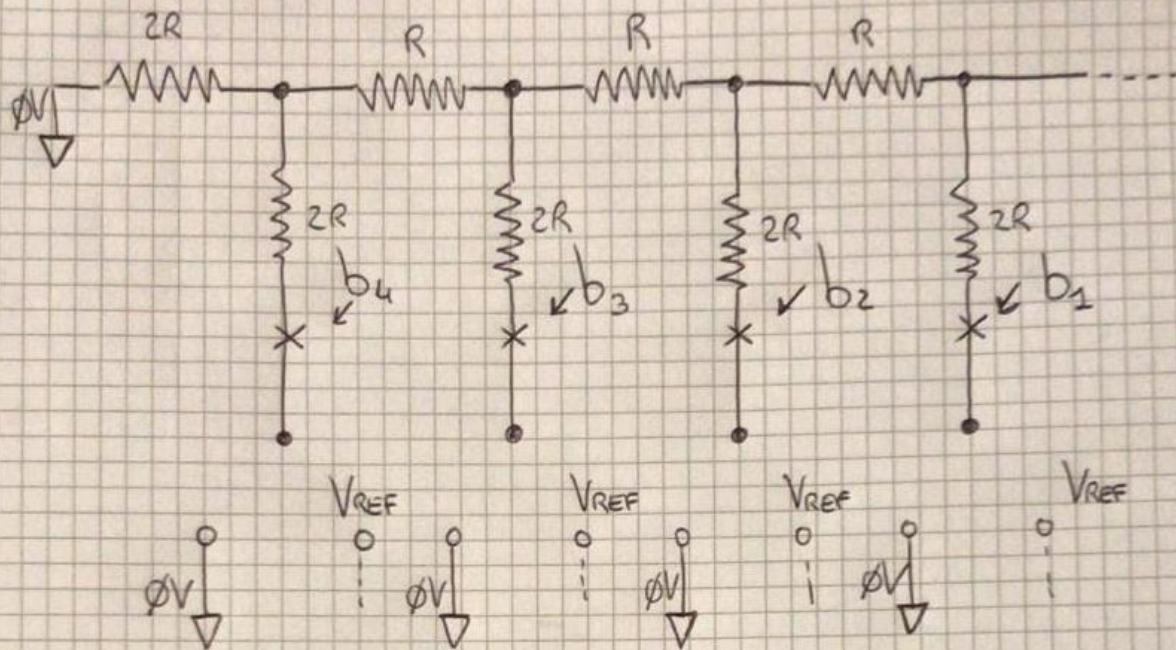
```
* C:\Users\orazi\Desktop\DAC_Orazio.asc
V1 0 N010 5
V2 N002 0 5
XU1 0 N001 N002 N010 OUT AD8092
R1 OUT N001 1k
R2 N001 N009 1k
R3 N009 N008 1k
R4 N008 N007 1k
R6 b4 N007 2k
R7 b3 N008 2k
R8 b2 N009 2k
R9 b1 N001 2k
R10 0 N007 2k
V3 0 b4 PULSE(0 3 0 1u 1u 10m 20m 100)
V5 0 b1 PULSE(0 3 0 1u 1u 80m 160m 100)
V6 0 b2 PULSE(0 3 0 1u 1u 40m 80m 100)
V4 0 b3 PULSE(0 3 0 1u 1u 20m 40m 100)
```



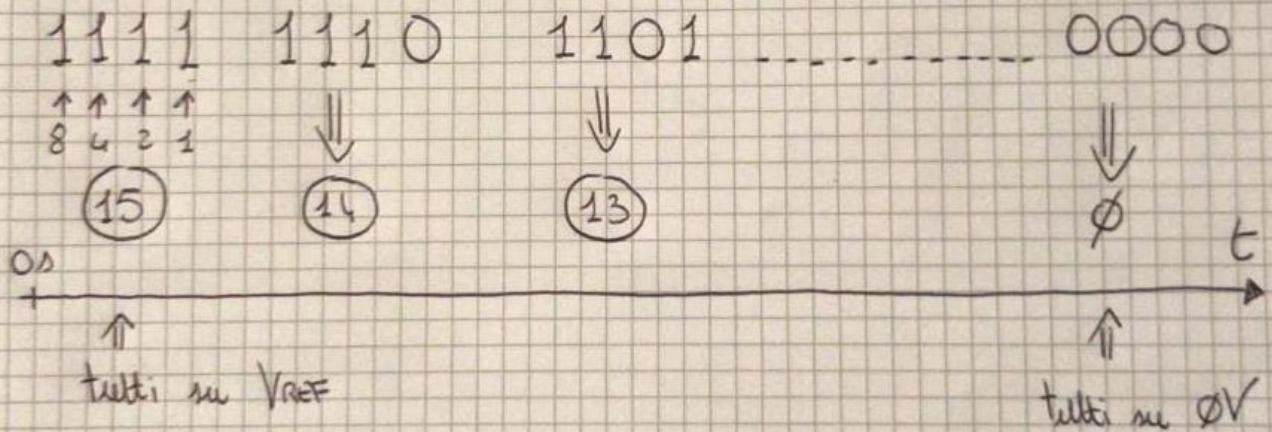
**Circuito creato tramite interfaccia grafica:**



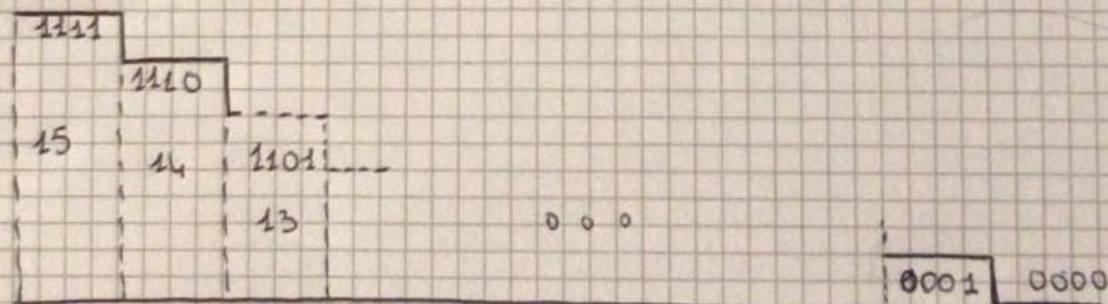
**Descrizione operativa:** L'approccio utilizzato per generare questo convertitore include due elementi diversi rispetto a quelli visti dal punto di vista teorico. Il primo riguarda i due generatori V1 e V2 che applicano una specifica tensione all'amplificatore. Ogni circuito reale deve essere connesso ad una linea di alimentazione. L'amplificatore essendo un elemento "energeticamente attivo" deve esser alimentato, e sulla base della scelta del componente abbiamo differenti specifiche sull'alimentazione (in questo caso  $\pm 5$  V). Il secondo elemento diverso dal modello visto in precedenza è il sistema di generatori di tensione V3,4,5,6. Tutti e quattro questi generatori sono fondamentali per simulare il compito svolto dalla ben più complessa rete di interruttori pilotati dal valore dei bit, permettendoci di analizzare una specifica sequenza di word che in uscita dovrebbe restituire un segnale a dente di sega. Prima di vedere per quale motivo ci aspettiamo un segnale a dente di sega interpolato con impulsi rettangolari in uscita possiamo notare un ultimo aspetto: i generatori sono tutti rivolti con il potenziale negativo verso le resistenze, al contrario di come dovrebbe essere applicata la tensione  $V_{ref}$  nella rete R-2R di partenza. Il motivo di questa scelta è dovuto a quanto visto nella brevissima illustrazione dell'amplificatore operazionale invertente. In quanto tale, per non avere un segnale di uscita in opposizione di fase rispetto a quello che ci aspettiamo di vedere, occorre invertire il segno delle tensioni applicate. Vediamo adesso il funzionamento della sequenza di generatori di onde quadre. Rispetto al circuito ad interruttori che abbiamo visto per avere un segnale a dente di sega in uscita dovremmo avere in ingresso una ripetizione periodica dei valori binari in ingresso dal valore 1111 fino a 000, dove il periodo di ripetizione sarà proprio il periodo del segnale a dente di sega.



Se in ingresso ho:



Il risultato mi aspetto:

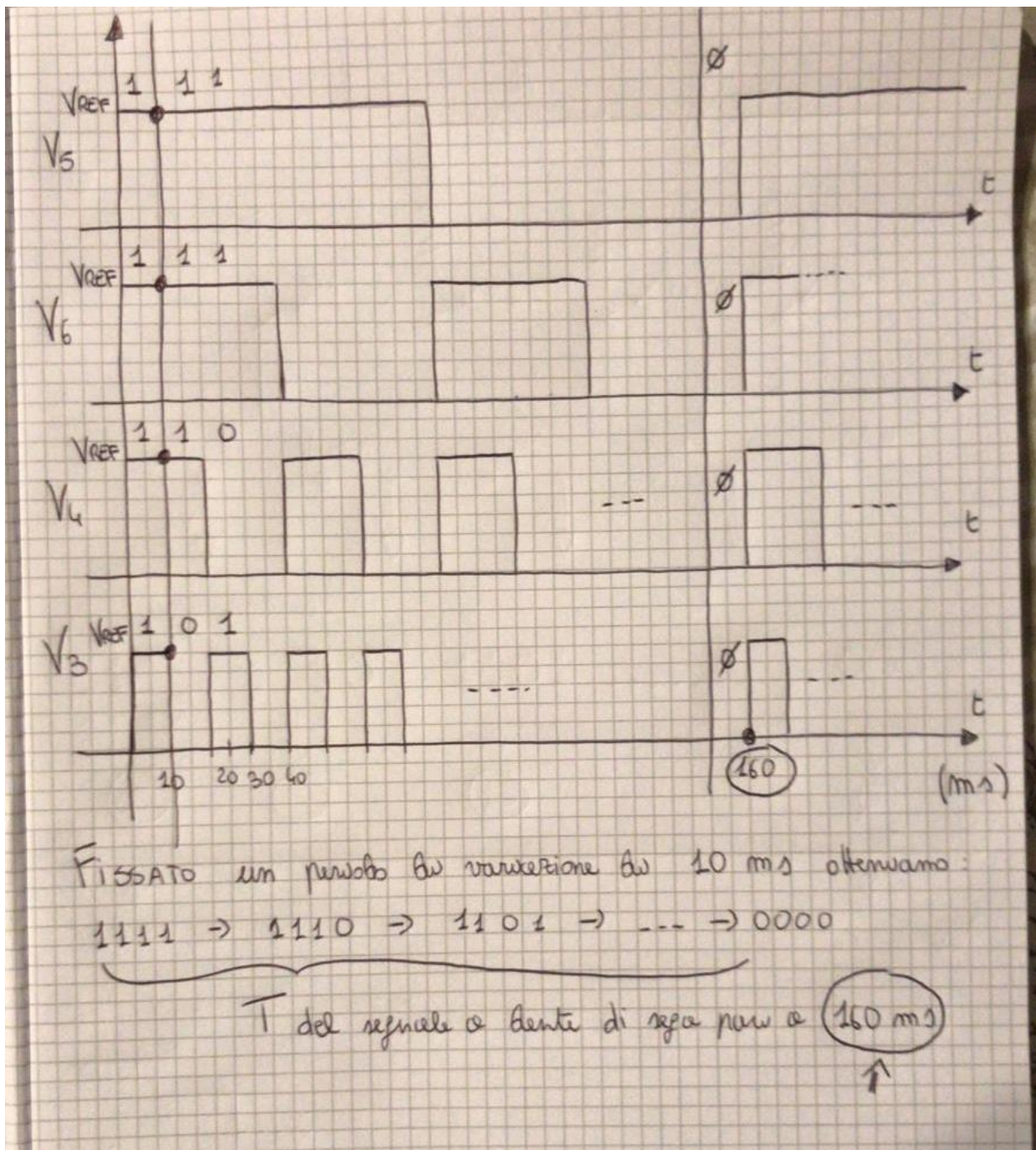


INTERPOLAZIONE:

Con un periodo completo otteniamo un "DENTE" con  $2^n = 4$  impulsi interpolatori.



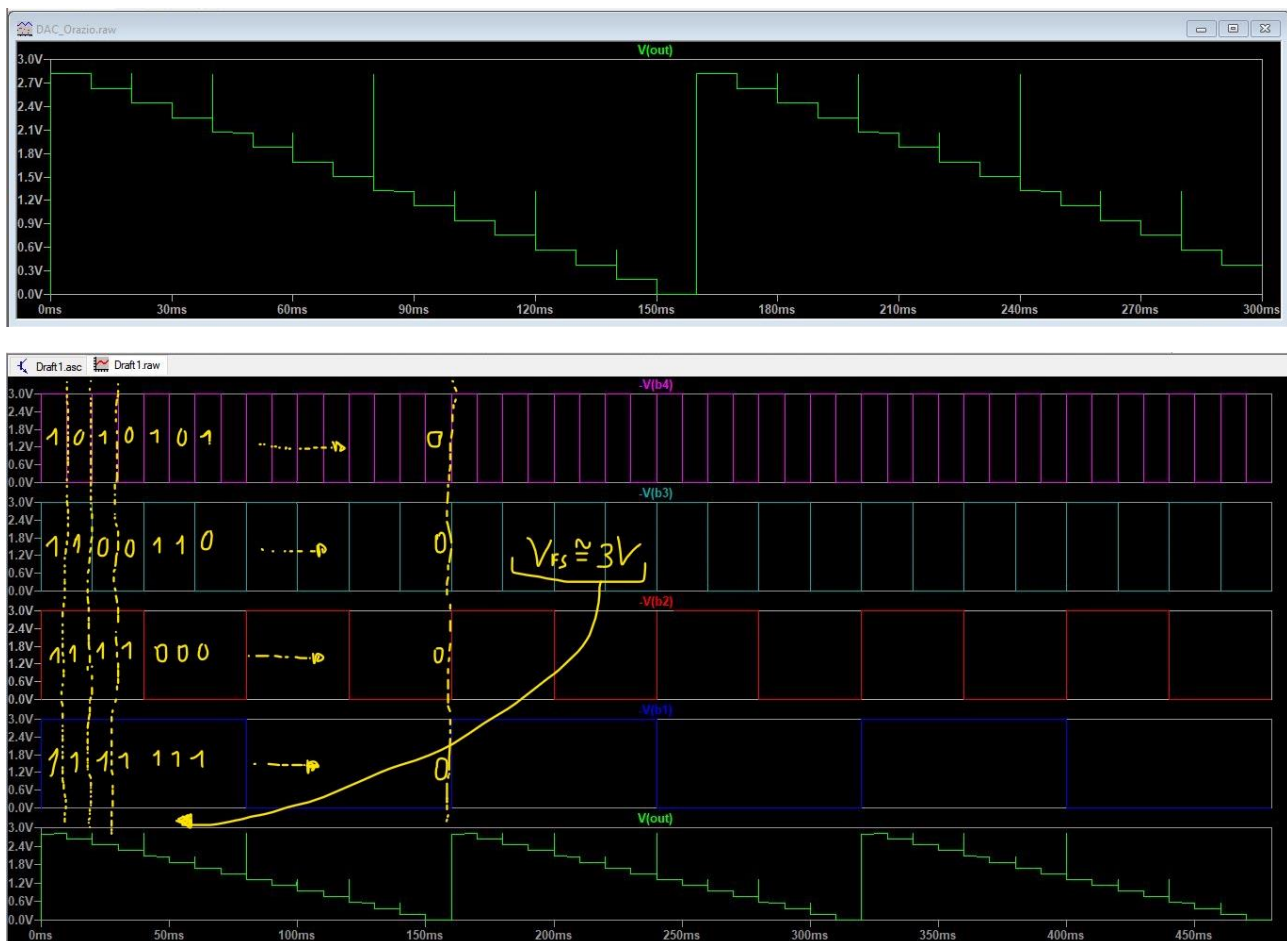
I generatori utilizzati in simulazione (come verrà mostrato anche in seguito utilizzando i grafici delle forme d'onda che possiamo tracciare grazie al tool) non fanno altro che andare a ricreare le medesime sequenze di word binarie in ingresso nel seguente modo:



Quindi i generatori di onde quadre (che oscillano tra  $V_{ref}$  e 0 Volt) attivati contemporaneamente, con periodi che saranno pari al periodo dell'onda da ricreare diviso il peso binario del bit  $i$ -esimo all'interno della word, simulano in modo assolutamente equivalente il compito svolto dalla rete di interruttori, così da poter ottenere il risultato sperato senza dover affrontare effetti secondari non trascurabili dovuto all'utilizzo di MOSFET o BJT per simulare gli interruttori. Quindi abbiamo che ogni generatore è stato dimensionato con un periodo di (con  $i = 1, \dots, n$ ):

$$\text{Periodo generatore } V_x \text{ associato a } B_i = \frac{\text{Periodo onda a dente di sega}}{2^{(i-1)}}$$

## Risultati simulazione:



**Osservazioni:** Dalla simulazione ci accorgiamo che ci sono delle anomalie. Prima anomalia sono questi impulsi nella ricostruzione del segnale a dente di sega. La seconda anomalia è data dal non completo raggiungimento dei 3V in uscita in presenza della configurazione 1111 in ingresso. Le anomalie sono dovute al fatto che la nostra analisi iniziale si è basata su un modello ideale del DAC. Nel mondo reale dell'elettronica occorre fare i conti con effetti molto complessi dovuti alla non idealità delle componenti utilizzate. Anomalie a parte, “chiudendo un occhio 😊”, possiamo notare come quanto pronosticato attraverso calcoli fatti a priori si sia dimostrato non così lontano dal risultato della simulazione. Con un po' di fantasia nel cercare una soluzione adatta alle nostre esigenze siamo riusciti a simulare il funzionamento di un dispositivo reale.

### Se aumentassimo il numero di bit/word e dunque alterassimo di conseguenza la struttura del DAC???

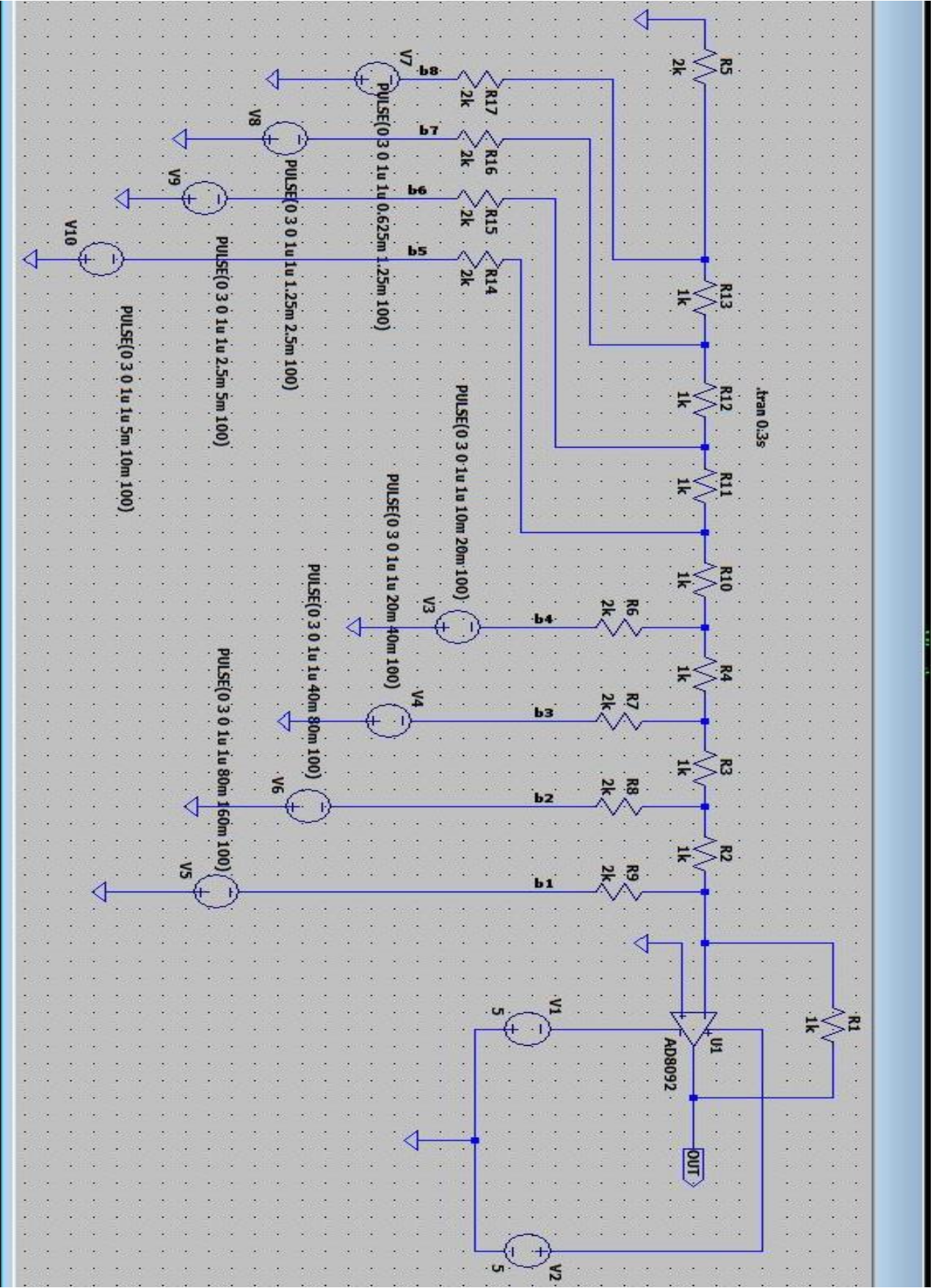
A questo punto, scoperti i trucchi sulla costruzione atta ad ottenere uno specifico risultato in fase di simulazione, ci aspettiamo, a meno di qualche “non idealità”, di ottenere una approssimazione migliore del segnale desiderato. Questo perché, a parità di range dinamico, aumentiamo il numero di bit di codifica e quindi necessariamente riduciamo il passo di quantizzazione. Progettiamo la rete di generatori come fatto in precedenza, quindi applicando sempre la stessa formula per il calcolo del periodo dell'onda associata al singolo generatore di tensione, e osserviamo i risultati ottenuti. Adesso ci aspettiamo di avere 256 stati di quantizzazione, cioè un numero di stati 16 volte maggiore rispetto al caso precedente. (I DAC R-2R a 8 e a 16 bit sono molto diffusi). Quanto stiamo operando è generalizzabile per un convertitore di N bit idealmente (in realtà all'aumentare del numero dei bit poi subentrano effetti secondari non più trascurabili).



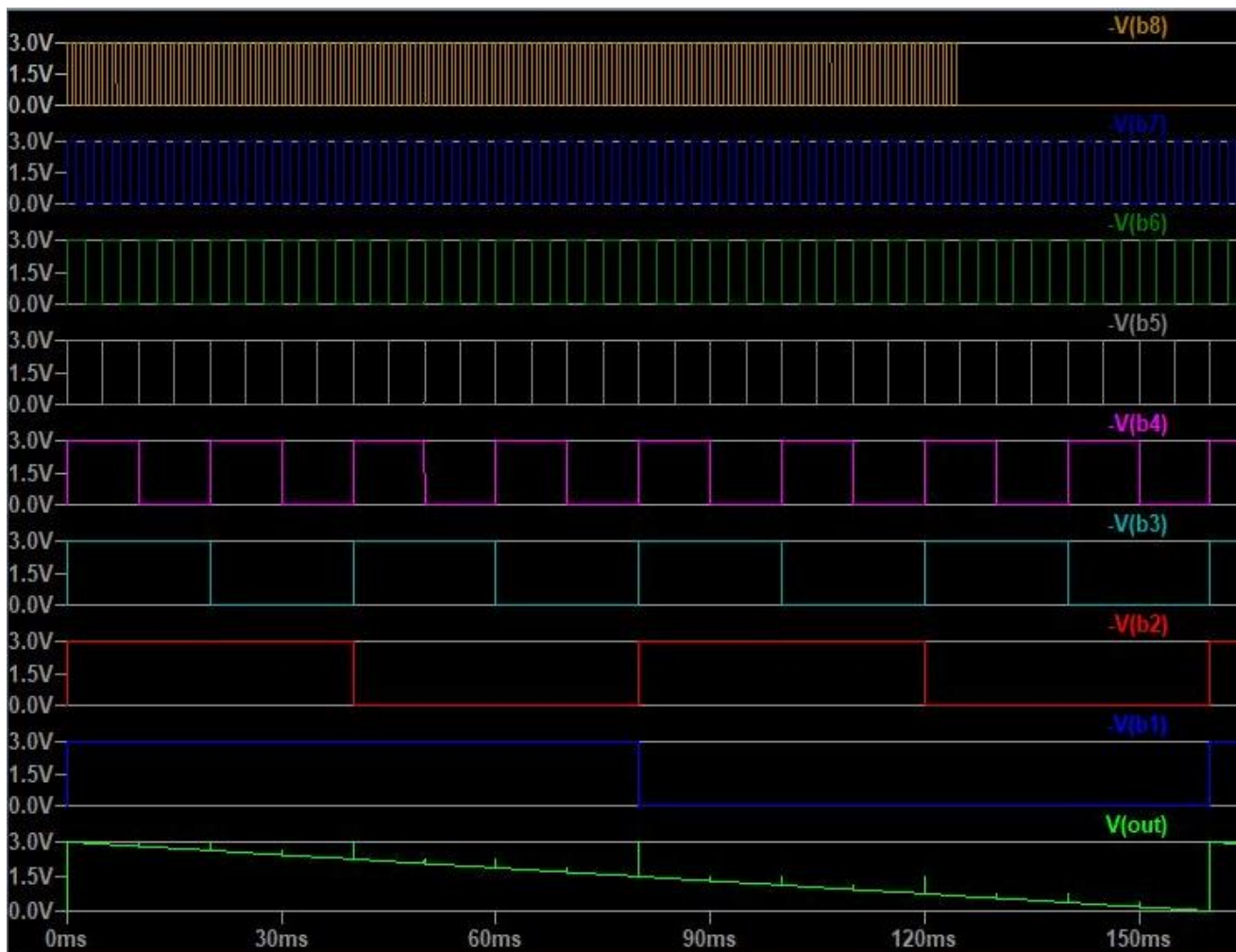
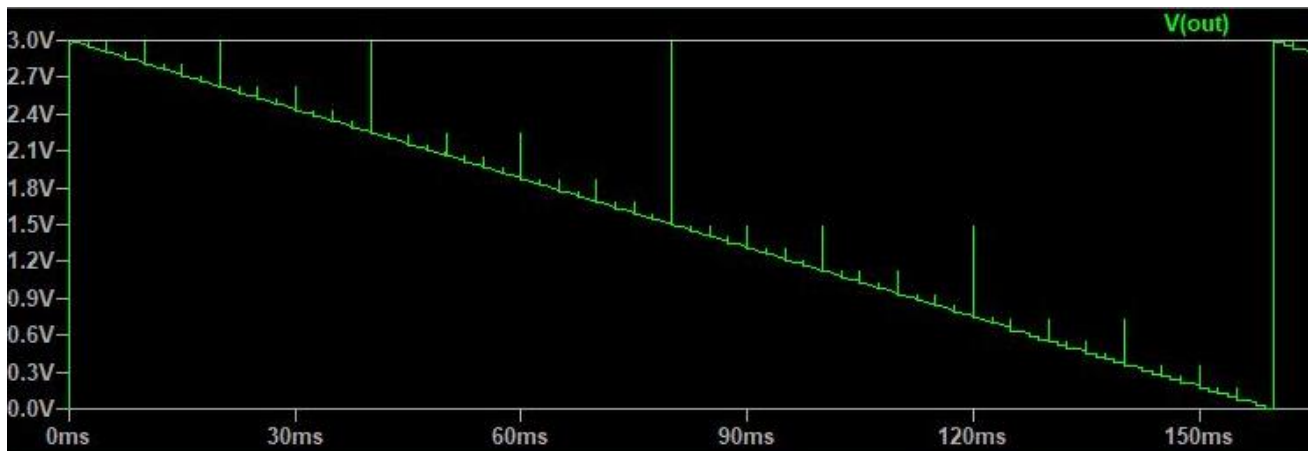
Netlist del nuovo circuito DAC:

```
* C:\Users\orazi\Desktop\DAC_Orazio.asc
V1 0 N010 5
V2 N002 0 5
XU1 0 N001 N002 N010 OUT AD8092
R2 N001 N009 1k
R3 N009 N008 1k
R4 N008 N007 1k
R5 N003 0 2k
R6 b4 N007 2k
R7 b3 N008 2k
R8 b2 N009 2k
R9 b1 N001 2k
V3 0 b4 PULSE(0 3 0 1u 1u 10m 20m 100)
V5 0 b1 PULSE(0 3 0 1u 1u 80m 160m 100)
V6 0 b2 PULSE(0 3 0 1u 1u 40m 80m 100)
V4 0 b3 PULSE(0 3 0 1u 1u 20m 40m 100)
R1 OUT N001 1k
R11 N006 N005 1k
R12 N005 N004 1k
R13 N004 N003 1k
R15 N005 b6 2k
R16 N004 b7 2k
R17 N003 b8 2k
R14 N006 b5 2k
V8 0 b7 PULSE(0 3 0 1u 1u 1.25m 2.5m 100)
V9 0 b6 PULSE(0 3 0 1u 1u 2.5m 5m 100)
V10 0 b5 PULSE(0 3 0 1u 1u 5m 10m 100)
V7 0 b8 PULSE(0 3 0 1u 1u 0.625m 1.25m 100)
R10 N007 N006 1k
.tran 0.3s
.lib ADI.lib
.backanno
.end
```

Circuito al simulatore:



Risultati simulazione:



**Osservazioni:** Possiamo notare come quanto avevamo pronosticato si sia realizzato attraverso la simulazione.

***The end!!!***