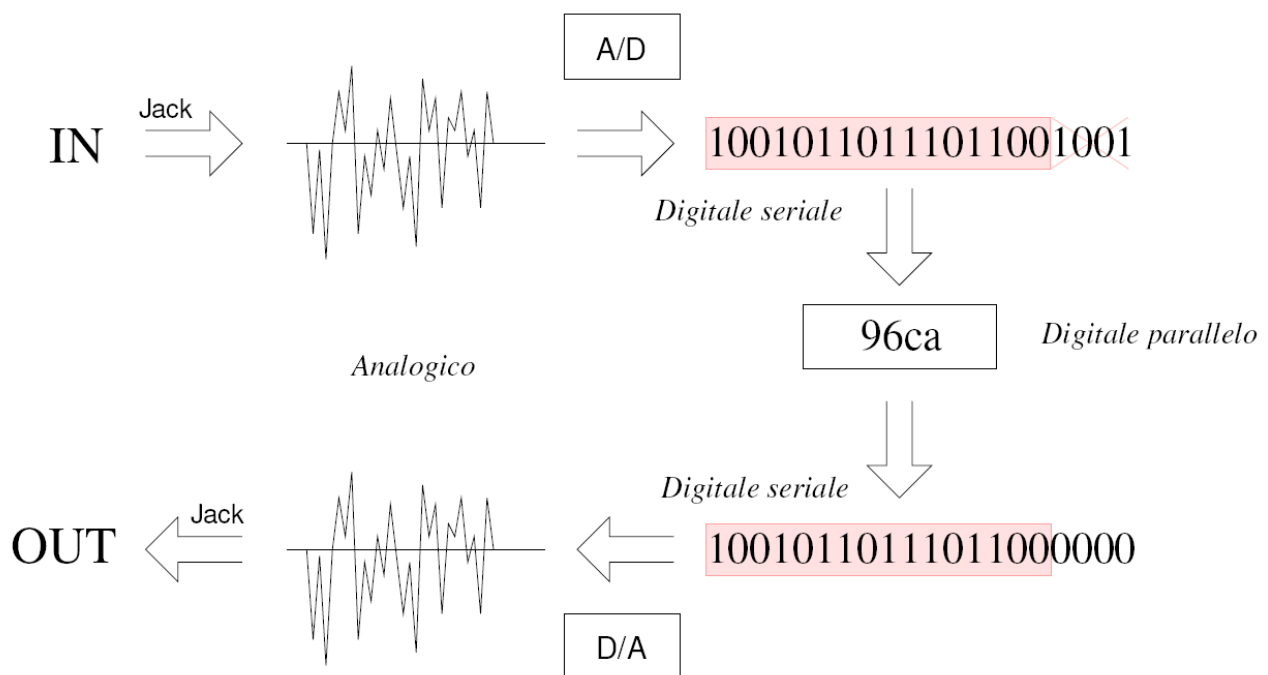


CONVERTITORI AUDIO

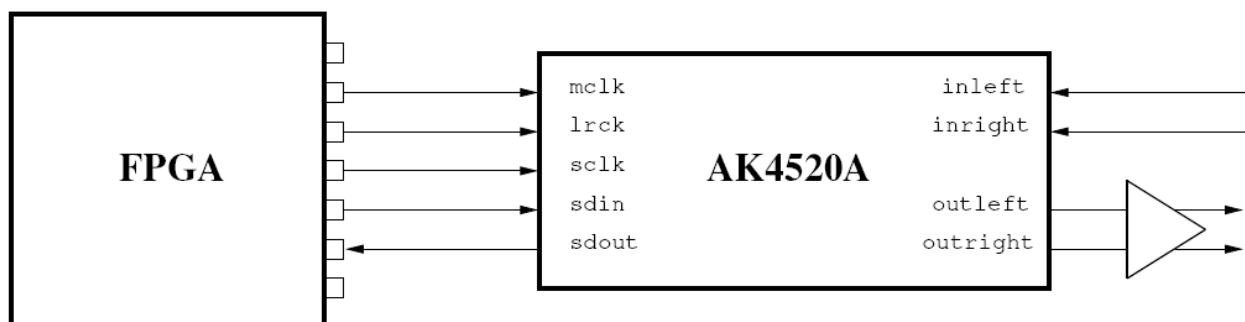
IDEA DI BASE

Questa sezione si occupa di pilotare i *convertitori audio* presenti sulla scheda *Xess XSA-50*, convertire tramite opportuni *registri* il *segnale seriale* d'ingresso in pacchetti da 16 bit (pronti per una successiva elaborazione), quindi rigenerare un segnale seriale e portarlo in uscita. Quindi, inizialmente, si supporrà di eliminare la parte centrale del progetto per realizzare un semplice in/out audio.



IL CONVERTITORE AK4520A

La scheda in dotazione possiede un convertitore stereo *AK4520A* che accetta in *ingresso* due *segnali analogici* (canale destro e sinistro) dal *jack J1*, *converte* il segnale da *analogico* a *digitale* e spedisce il *segnale digitalizzato* all'*FPGA* in modo *seriale*; in maniera duale *accetta* dall'*FPGA* un flusso *seriale* di bit e lo converte nei due *segnali digitali* in *uscita* che escono dalla scheda attraverso il *jack J2* (dopo essere stati amplificati da un amplificatore integrato).



I CLOCK INTERNI

Il convertitore in questione utilizza tre clock: *lrck*, *sclk* e *mclk* pilotati dall'*FPGA*:

- *lrck* è il *segnale* che seleziona il *canale* sinistro (1) o destro (0): questo *clock* determina la *frequenza di campionamento* in quanto dice al convertitore quando prendere ogni campione;
- *sclk* è usato per *sincronizzare* le sequenze di bit: deve avere almeno tanti cicli (per canale) quanti il bitrate di ogni campione prelevato, il che significa che se campiono a 20 bit deve essere almeno 40 volte la frequenza di campionamento;
- *mclk* è il *master clock* che viene usato per *sincronizzare* le operazioni interne al *convertitore*.

Benché l'AK4520A supporti quattro modi diversi di gestire i dati seriali attraverso i suoi pin (*DIF1* e *DIF0* per il bitrate, *CMODE* per la frequenza di campionamento e di conseguenza per il master clock), la scheda non dà la possibilità di agire su questi piedini avendoli già collegati il primo a *DVDD*, il secondo e il terzo a *massa* (si vedano i *data sheet*). Di conseguenza i vincoli sono:

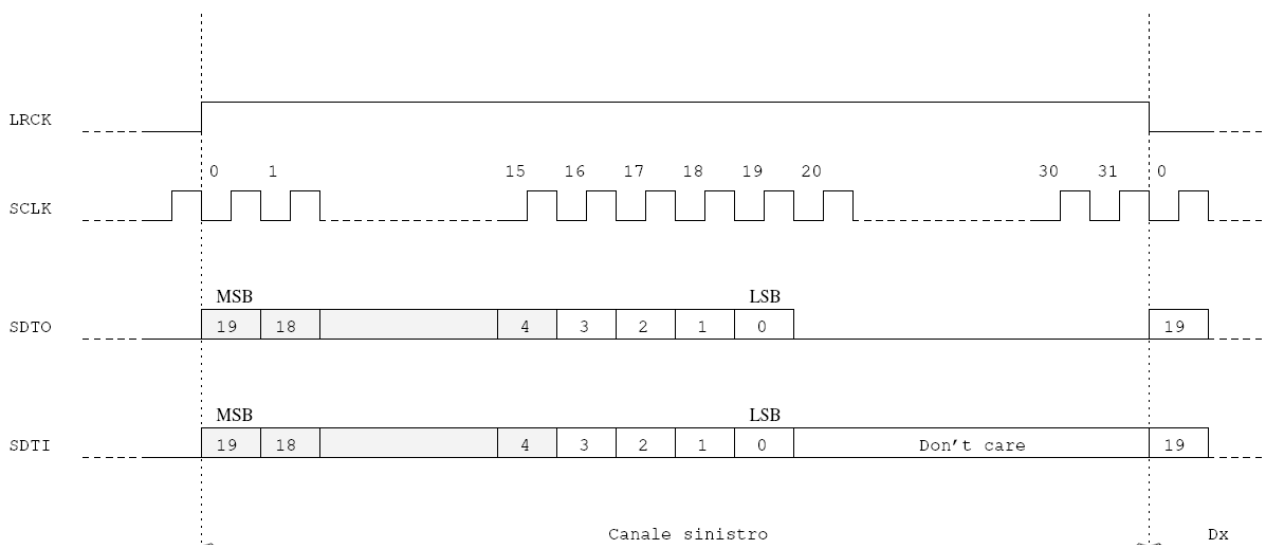
- *sdto* (ADC): 20 bit, MSB justified;
- *sdti* (DAC): 20 bit, MSB justified;
- *sclk* $\geq 40fs$ (fs = frequenza di campionamento);
- *mclk*: 256fs.

In particolare in questo sotto-progetto si userà un *sclk* pari a 64fs, quindi i vari clock avranno un valore di:

- *clk*: 50MHz (il clock globale del progetto);
- *mclk*: 12.5MHz (256fs);
- *sclk*: 3.125 MHz (64fs);
- *lrck*: 48.828KHz (fs).

LA CODIFICA DEL SEGNALE

Come si è già detto, il segnale esce dal convertitore *A/D* trasformato in un treno di 20 bit, ordinati dal più significativo al meno significativo. La seguente figura mostra la sincronizzazione di questo processo:

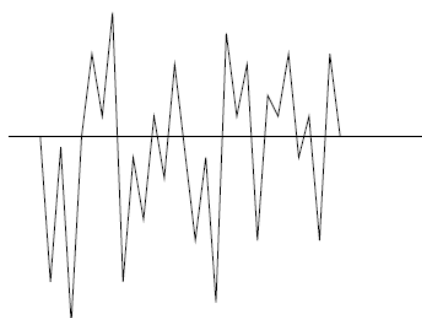


- il segnale *lrck* passa allo *stato logico alto*, quindi i convertitori di ingresso e di uscita si occupano del segnale sul canale sinistro;

- il treno di 20 bit inizia ad uscire dal convertitore *A/D* non appena *lrck* passa da uno *stato logico*, ad esempio, *basso* (canale destro) a quello *alto* (canale sinistro);
- allo stesso modo il convertitore *D/A* in uscita si aspetta che il treno di bit inizi ad arrivare nel momento in cui *lrck* cambia *stato*;
- i bit entrano (ed escono) *sincronizzati* con il segnale *sclk*.

Ogni fs^{-1} secondi il convertitore *A/D* campiona la linea d'ingresso e codifica il campione su 20 bit in *complemento a 2*. Il convertitore codifica con 0x7FFFF(@20bit) ogni tensione pari o maggiore a 5V e con 0x80000(@20bit) ogni tensione pari o minore a -5V; idealmente l'assenza di segnale dovrebbe venir codificata con 0x00000(@20bit). L'*offset* in continua viene rimosso dal *filtro passa-alto* interno al *AK4520A*.

limite max ----- 0x7FFFF(@20bit) = 01111111111111111111



0x00000(@20bit) = 00000000000000000000

limite min ----- 0x80000(@20bit) = 10000000000000000000

FREQUENZA DI CAMPIONAMENTO E BITRATE

Ci sono due caratteristiche del progetto che possono sembrare inusuali: la *frequenza di campionamento* a 48,828 kHz (anziché i classici 44'100) e la decisione di utilizzare solo 16 dei 20 bit messi a disposizione dai convertitori. Queste due caratteristiche sono frutto di una scelta esclusivamente pratica:

- generare un *clock* di 48,828 kHz a partire da un *master clock* di 50 MHz è estremamente più semplice che generarne uno di 44,100 kHz: nel primo caso infatti è sufficiente dividere il clock per una potenza di due ($50'000'000 / 1'024 = 48'828,125$);
- la scelta di utilizzare solo 16 bit deriva dal fatto (banale) che risulta molto più comodo lavorare con 2 byte anziché con due byte e mezzo... (si pensi ad esempio alle *memorie SDRAM* presenti sulle scheda *XSA*, organizzate in celle da 16 bit); in questo caso vengono trascurati i 4 bit meno significativi, forzati a zero all'uscita (come mostrato nella prima figura di questa sezione).