

EsD1: Caratterizzazione di porte logiche e semplici circuiti logici.

Gruppo 1.AC
Matteo Rossi, Bernardo Tomelleri

12 aprile 2022

Misura componenti dei circuiti

Resistenze [$k\Omega$]	R	σR
R_{pot1}	9.53	0.08
R_{pot2}	9.78	0.08

Tabella 1: Valori di resistenza misurati per i componenti passivi dei circuiti studiati.

Riportiamo per completezza anche il valore della tensione continua di alimentazione per i circuiti integrati misurata con il multmetro

$$V_{CC} = 4.99 \pm 0.03V$$

Nota sul metodo di fit

Per determinare i parametri ottimali e le rispettive covarianze si è implementato in Python un algoritmo di fit basato sui minimi quadrati mediante la funzione *curve_fit* della libreria SciPy.

Parte A: Caratteristiche fisiche delle porte logiche

Si studia il comportamento delle porte NOT TTL contenute nel circuito integrato SN7404 misurando le tensioni e correnti di operazione e verificando che queste rientrino nelle specifiche tecniche riportate nel Data-Sheet del chip in fig. 1.

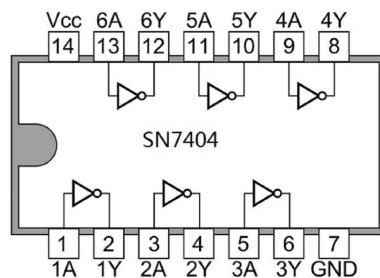


Figura 1: Schema e pedinatura del chip integrato quad-NOT SN7404

1 Tensioni di operazione

Per prima cosa misuriamo i valori delle tensioni di soglia in ingresso e in uscita (e verifichiamo che rispettino le specifiche di buon funzionamento del DS) da cui è possibile ottenere una misura del Noise Margin delle porte.

Dalle specifiche del DS si ha che i valori attesi sono: (riportati in tabella 2)

in cui V_O e V_I sono definite come le tensioni in uscita e in ingresso dalla porta logica (le altre diciture indicano se la grandezza a cui facciamo riferimento corrisponde a uno stato logico alto (H) o basso (L) e quali sono i massimi o minimi valori garantiti dal costruttore).

Parameter	min	typ	max	[Unit]
V_{CC}			7	V
V_I			5.5	V
V_{OH}	2.4	3.4		V
V_{OL}		0.2	0.4	V
V_{IH}	2			V
V_{IL}			0.8	V
I_{IH}			40	μA
I_{OH}			-0.4	mA

Tabella 2: Valori delle tensioni e correnti di operazione indicati sul datasheet dell'integrato SN7404.

1.a Misura delle tensioni di soglia dal grafico $V_{out}(V_{in})$

Generiamo una rampa di tensione compresa tra 0-5 V e la inviamo all'ingresso di una porta NOT per osservare i segnali generati in uscita, così da ottenere un grafico di V_{out} in funzione di V_{in} .

Abbiamo quindi utilizzato i cursori per misurare V_{OH} e V_{OL} , cioè la tensione che raggiunge l'uscita in saturazione per valore logico H e L rispettivamente, mentre per misurare V_{IL} e V_{IH} abbiamo misurato le tensioni in ingresso per cui inizia e finisce la commutazione dell'uscita secondo lo schema in fig. 2:

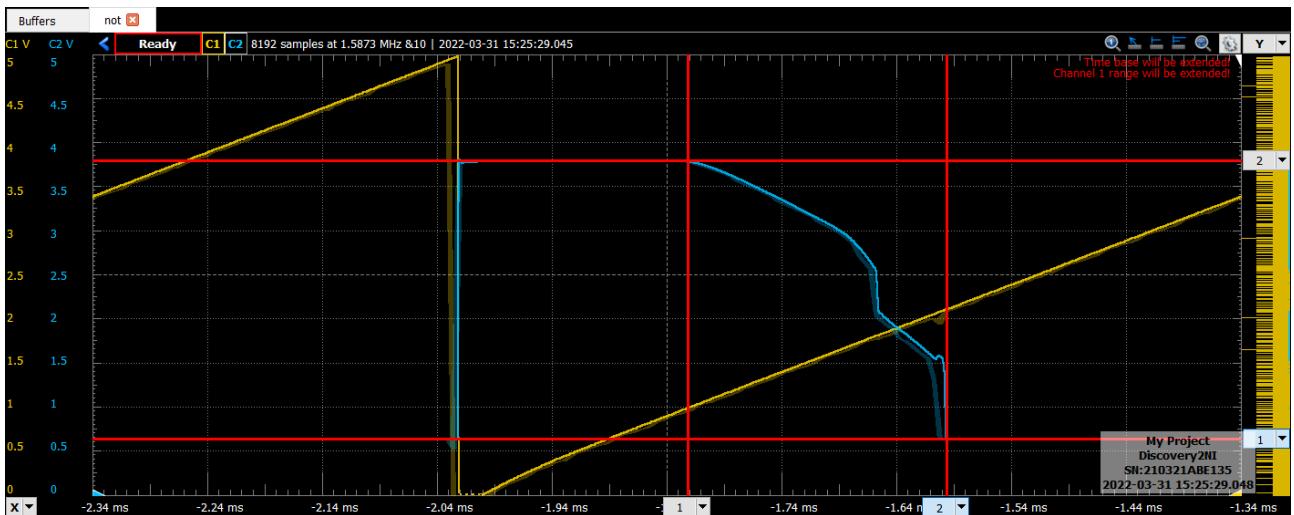


Figura 2: Schema di come sono stati utilizzati i cursori per misurare le tensioni V_{OH} , V_{OL} (cursori orizzontali) e V_{IH} , V_{IL} (aiutandosi con i cursori verticali a individuare la fase di transizione H->L)

$$\begin{aligned} V_{OH} &= 3.78 \pm 0.03 \text{ V} \\ V_{OL} &= 632 \pm 4 \text{ mV} \\ V_{IH} &= 2.10 \pm 0.02 \text{ V} \\ V_{IL} &= 979 \pm 9 \text{ mV} \end{aligned}$$

$$\begin{aligned} V_{OH} &= 3.76 \pm 0.03 \text{ V} \\ V_{OL} &= 341 \pm 3 \text{ mV} \\ V_{IH} &= 1.81 \pm 0.02 \text{ V} \\ V_{IL} &= 772 \pm 8 \text{ mV} \end{aligned}$$

Sempre con i cursori abbiamo misurato $V_{OH,\min}$ dato che il valore minimo per lo stato alto dell'uscita risultava più basso di quanto misurato prima; di conseguenza abbiamo spostato il cursore nel punto in cui la tensione in uscita cambia in maniera più repentina:

Successivamente si sono utilizzati i valori misurati già elencati prima per ottenere le nostre stime delle tensioni di soglia. Per il primo integrato:

$$\begin{aligned} V_{OH,\min} &= 2.53 \pm 0.03 \text{ V} \\ V_{IH,\min} &= 2.10 \pm 0.02 \text{ V} \end{aligned}$$

$$\begin{aligned} V_{IL,\max} &= 979 \pm 9 \text{ mV} \\ V_{OL,\max} &= 632 \pm 4 \text{ mV} \end{aligned}$$

mentre per il secondo:

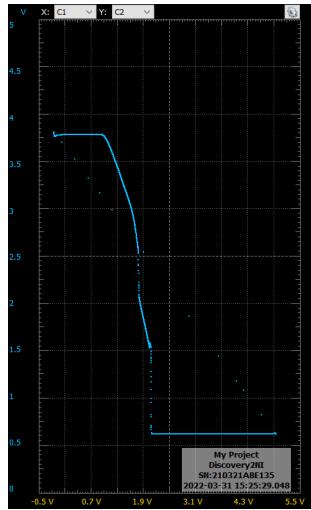


Figura 3: Grafico XY di V_{out} in funzione di V_{in}

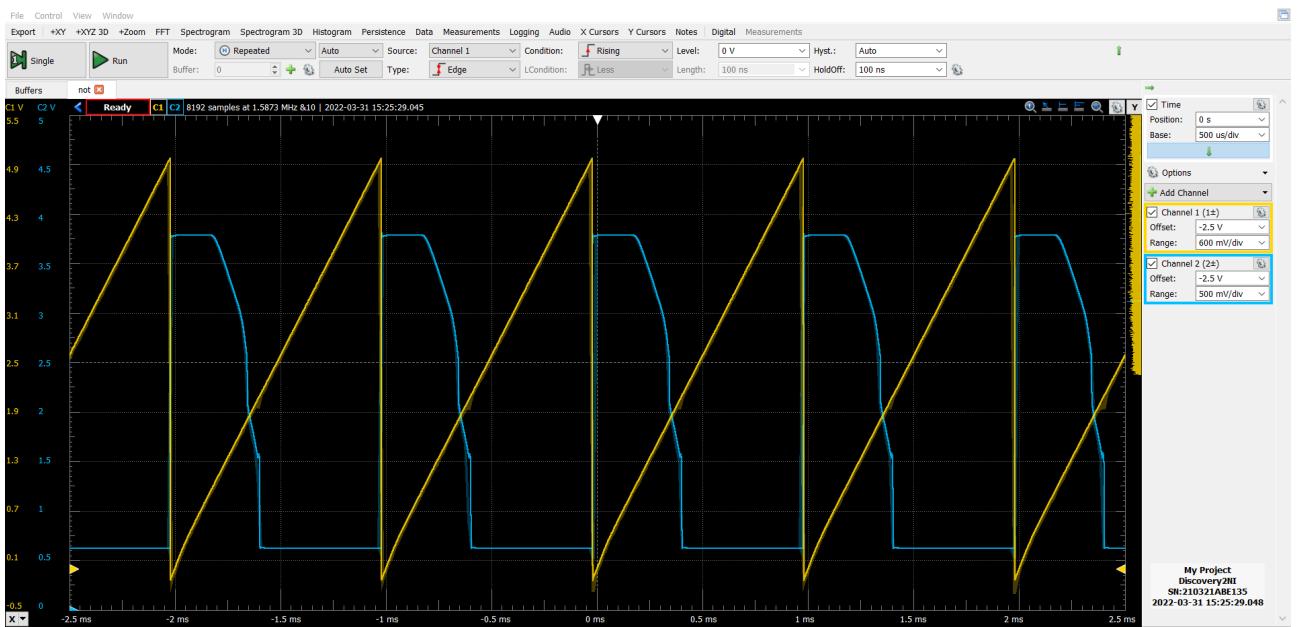


Figura 4: Acquisizione all'oscilloscopio dell'andamento nel tempo dei segnali V_{in} (CH1, una rampa da 0 a 5 V di frequenza $f = 1$ kHz) e V_{out} (CH2)

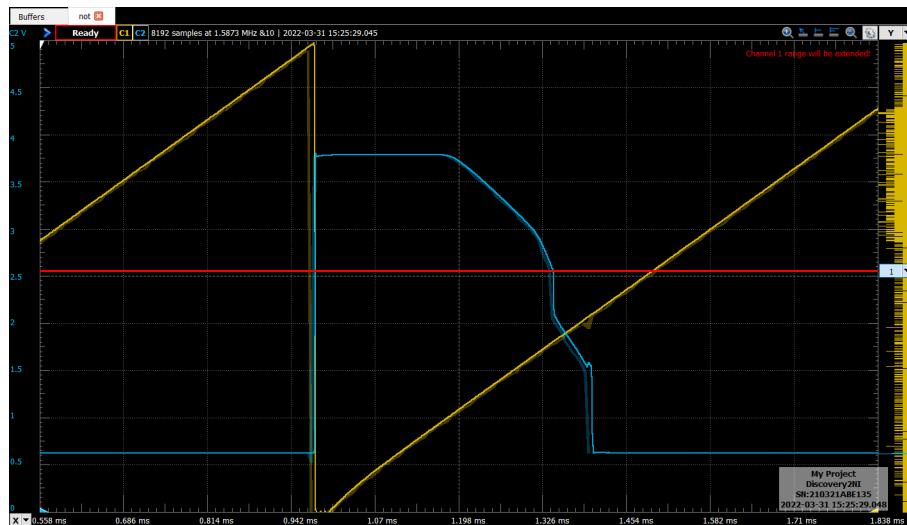


Figura 5: Schema di come è stata presa la misura di $V_{OH,min}$ tramite l'utilizzo dei cursori. Si è scelto come valore minimo la tensione dell'uscita subito prima del punto di discesa più ripida.

$$V_{OH,min} = 2.40 \pm 0.03 \text{ V}$$

$$V_{IH,min} = 1.81 \pm 0.02 \text{ V}$$

$$V_{IL,max} = 772 \pm 8 \text{ mV}$$

$$V_{OL,max} = 341 \pm 3 \text{ mV}$$

Da cui troviamo le nostre stime dei valori delle soglie di rumore (Noise Margin High e Low) per le porte NOT studiate, definite come

$$NM_H = V_{OH,min} - V_{IH,min} = (0.43 \pm 0.04) \text{ V}$$

$$= (0.60 \pm 0.03) \text{ V}$$

$$NM_L = V_{IL,max} - V_{OL,max} = (0.35 \pm 0.02) \text{ V}$$

$$= (0.43 \pm 0.02) \text{ V}$$

1.b Valori attesi per le soglie di rumore

Estrapolando i corrispettivi valori dal datasheet, e assumendo che ogni porta logica abbia i medesimi parametri delle altre presenti nello stesso integrato possiamo allora ricavare i valori attesi dei Noise Margin High e Low:

$$NM_H = 2.4 - 2 = 0.4 \text{ V}$$

$$NM_L = 0.8 - 0.4 = 0.4 \text{ V}$$

1.c Confronto dei risultati

Le nostre misure risultano compatibili con le aspettative entro gli intervalli ammessi dal datasheet per il buon funzionamento del componente, dunque anche le grandezze derivate risultano in accordo con le aspettative.

2 Misura (statica) del Fan-out

Chiamiamo Fan-Out il massimo numero di porte che una singola porta è in grado di pilotare rimanendo entro le specifiche di funzionamento del datasheet, lo si è definito come

$$FO = \left| \frac{I_{IH,max}}{I_{OH,max}} \right|$$

2.a Misura della corrente in ingresso alla porta NOT nello stato alto

Misuriamo la corrente $I_{IH,max}$ per entrambi gli integrati collegando l'amperometro in serie tra l'ingresso della porta logica e l'uscita del canale WaveGen1 dell'AD2, quindi generando in questo canale una tensione continua di livello alto pari a 5 V, da cui si trova

$$I_{IH,1} = 16 \pm 1 \mu\text{A}$$

$$I_{IH,2} = 10 \pm 1 \mu\text{A}$$

Che risultano essere compatibili entro i limiti del datasheet.

2.b Misura della corrente in uscita dalla porta NOT per VOH tipico

Successivamente abbiamo inviato all'ingresso della porta un segnale DC a 0 V (sempre utilizzando il canale WG1) e abbiamo inserito un potenziometro da 10 kΩ in serie all'uscita di questa per misurare la corrente che scorre attraverso questa "resistenza di carico" regolabile in modo da avere quando una tensione in uscita dalla porta corrispondente al valore tipico di $V_{OH} = 3.40 \pm 0.03$ V.

$$I_{OH,1} = 495 \pm 4 \mu\text{A}$$

$$I_{OH,2} = 315 \pm 3 \mu\text{A}$$

2.c Misura della corrente in uscita dalla porta NOT per VOH misurato

Per completezza abbiamo impostato il potenziometro in modo che la tensione V_{OH} risultasse (entro l'incertezza) pari a $V_{OH,min}$ e $V_{OH,sat}$ misurati precedentemente e si è ripetuta la misura di I_{OH} per entrambi gli integrati:

Riportiamo le nostre misure per $V_{OH,sat} = 3.78 \pm 0.03$ V

$$\begin{aligned}I_{OH,1} &= 453 \pm 4 \mu\text{A} \\I_{OH,2} &= 275 \pm 2 \mu\text{A}\end{aligned}$$

quindi per $V_{OH,min} = 2.50 \pm 0.02$ V

$$\begin{aligned}I_{OH,1} &= 782 \pm 6 \mu\text{A} \\I_{OH,2} &= 552 \pm 5 \mu\text{A}\end{aligned}$$

2.d Misura indiretta del fan-out

Dalle misure di corrente sui due integrati ricaviamo finalmente che:

$$\begin{aligned}\text{FO}_1 &= 31 \pm 2 \\ \text{FO}_2 &= 32 \pm 3\end{aligned}$$

Mentre usando le correnti misurate per gli altri valori di V_{OH} :

$$\begin{aligned}\text{FO}_{1,\text{sat}} &= 28 \pm 2 \\ \text{FO}_{2,\text{sat}} &= 28 \pm 3\end{aligned}$$

$$\begin{aligned}\text{FO}_{1,\text{min}} &= 49 \pm 3 \\ \text{FO}_{2,\text{min}} &= 55 \pm 6\end{aligned}$$

2.e Valore del fan-out atteso e confronto

Dalle specifiche del DS elencate sopra in 2 risulta che

$$\begin{aligned}I_{IH,\text{max}} &= -0.4 \text{ mA} \\I_{OH,\text{max}} &= 40 \mu\text{A} \\ \text{FO} &= 10\end{aligned}$$

I valori da noi trovati risultano sensibilmente più alti delle aspettative e non compatibili con queste, ma perlomeno rimangono compatibili tra di loro.

3 Tempi di propagazione

Vogliamo ora misurare i tempi di propagazione della porta logica NOT osservando (tramite un oscilloscopio da banco) l'andamento nel tempo dei segnali nei pin di ingresso uscita della porta durante le transizioni di stato per poi paragonarli ai valori riportati sul datatsheet in analogia a quanto fatto finora.

3.a Definizione e valori attesi dei tempi di propagazione

Dal datasheet TI fornito le misure di tempi di propagazione per la commutazione dell'uscita tra gli stati alto->basso t_{PHL} e basso->alto t_{PLH} sono definite come l'intervallo di tempo tra il passaggio della forma d'onda in ingresso e di quella in uscita dal livello di tensione iniziale al punto medio tra lo stato iniziale e finale, secondo lo schema in 6

Prendiamo quindi come definizione di tempo di propagazione il lasso di tempo che trascorre tra quando V_{in} diventa pari a $V_{I,med}$ e V_{out} diventa pari al suo $V_{O,med}$, dove i valori medi tra stato basso e stato alto rispettivamente per l'ingresso e per l'uscita sono definiti intuitivamente dalle formule

$$\begin{aligned}V_{O,med} &= \frac{V_{OH} + V_{OL}}{2} \\V_{I,med} &= \frac{V_{IH} + V_{IL}}{2}\end{aligned}$$

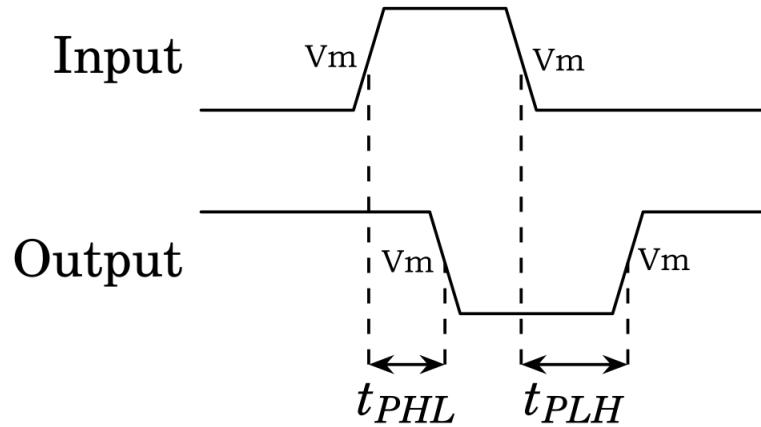


Figura 6: Diagramma dei tempi di propagazione per una porta NOT

Dal datasheet si ricavano come tempi di propagazione attesi:

$$t_{PHL,max} = 22 \text{ ns}$$

$$t_{PLH,max} = 15 \text{ ns}$$

3.b Misura dei tempi con l'oscilloscopio

Come visto anche nella sezione 1, inviando in ingresso un'onda quadra compresa tra 0 e 5 V, all'uscita della porta troviamo la stessa onda compresa tra $V_{OH} \approx 3.6$ V e $V_{OL} \approx 0.3$ V; infatti misurando con i cursori e con l'oscilloscopio troviamo come

$$V_{O,med} = 1.95 \pm 0.02$$

$$V_{I,med} = 2.50 \pm 0.03$$

Quindi, una volta collegati i due canali all'ingresso (CH1) e all'uscita (CH2) del NOT gate, abbiamo scelto come evento di trigger il passaggio del segnale di ingresso per $V_{I,med}$, abbiamo dunque cambiato il fronte tra salita/discesa per scegliere se misurare il tempo di propagazione H->L o L->H.

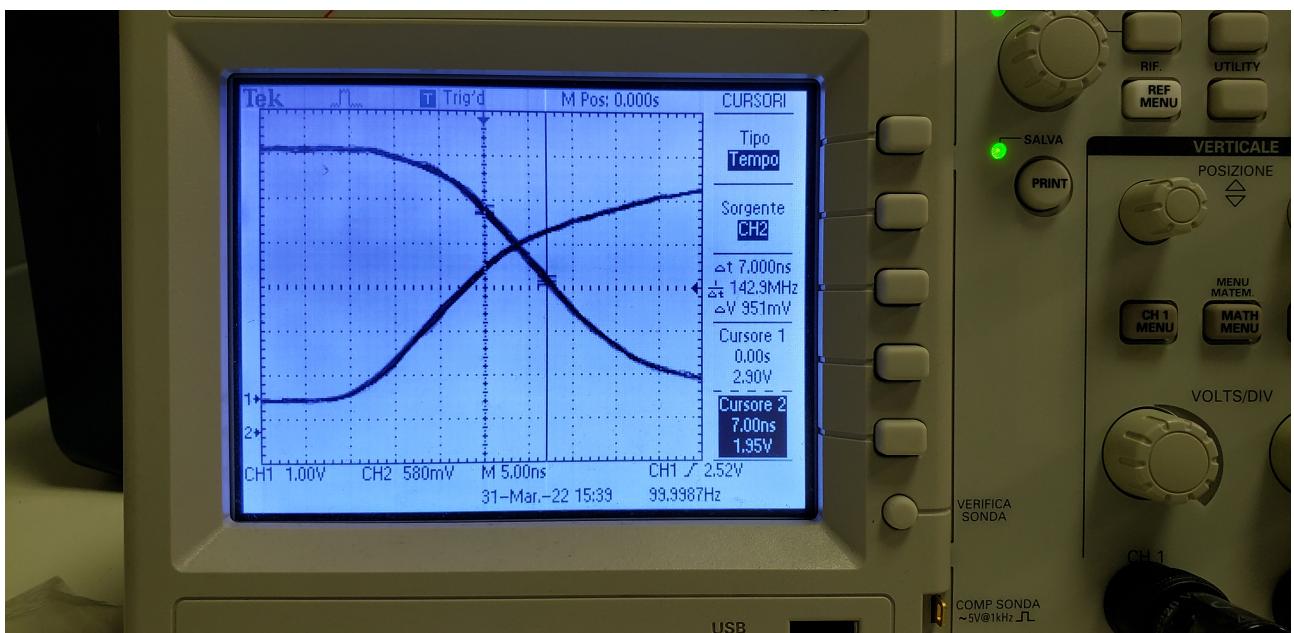


Figura 7: Acquisizione tramite oscilloscopio digitale della transizione da L a H per il primo integrato

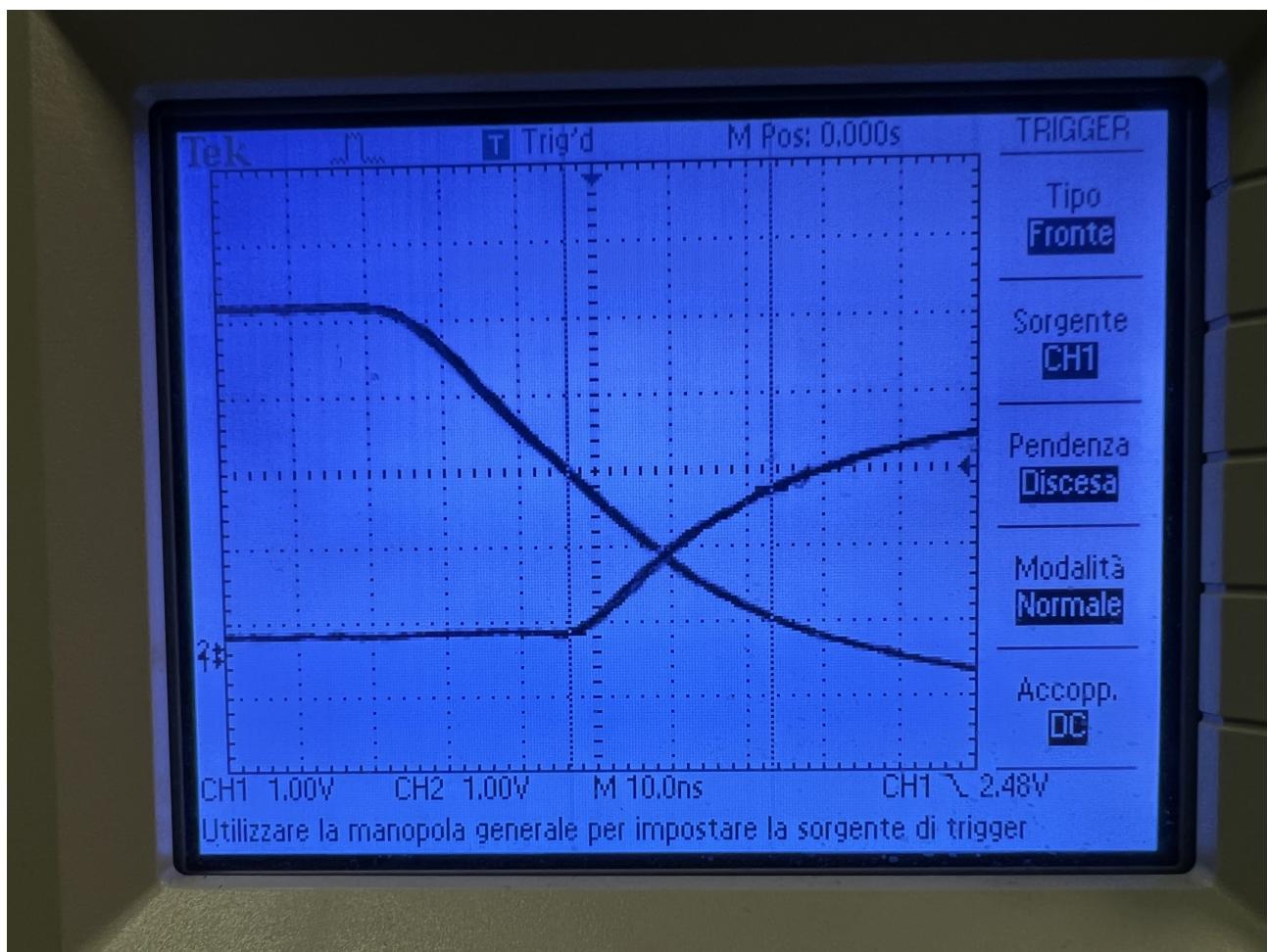


Figura 8: Acquisizione tramite oscilloscopio digitale della transizione da L a H per il secondo integrato

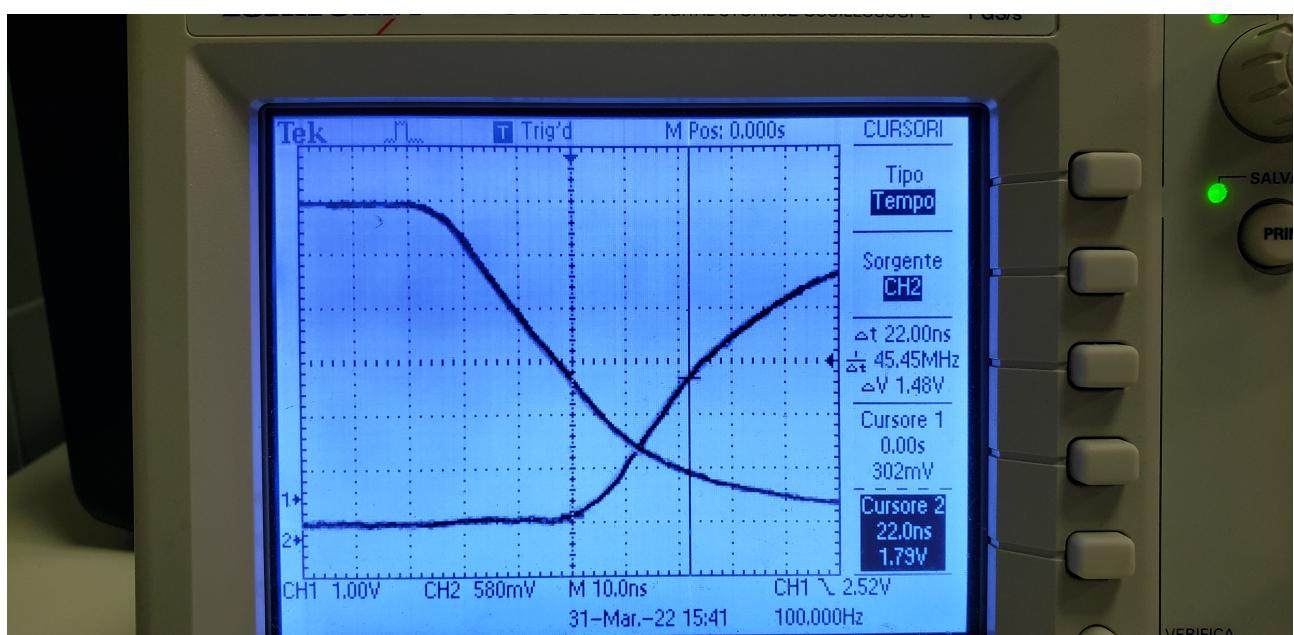


Figura 9: Acquisizione tramite oscilloscopio digitale della transizione da H a L per il primo integrato

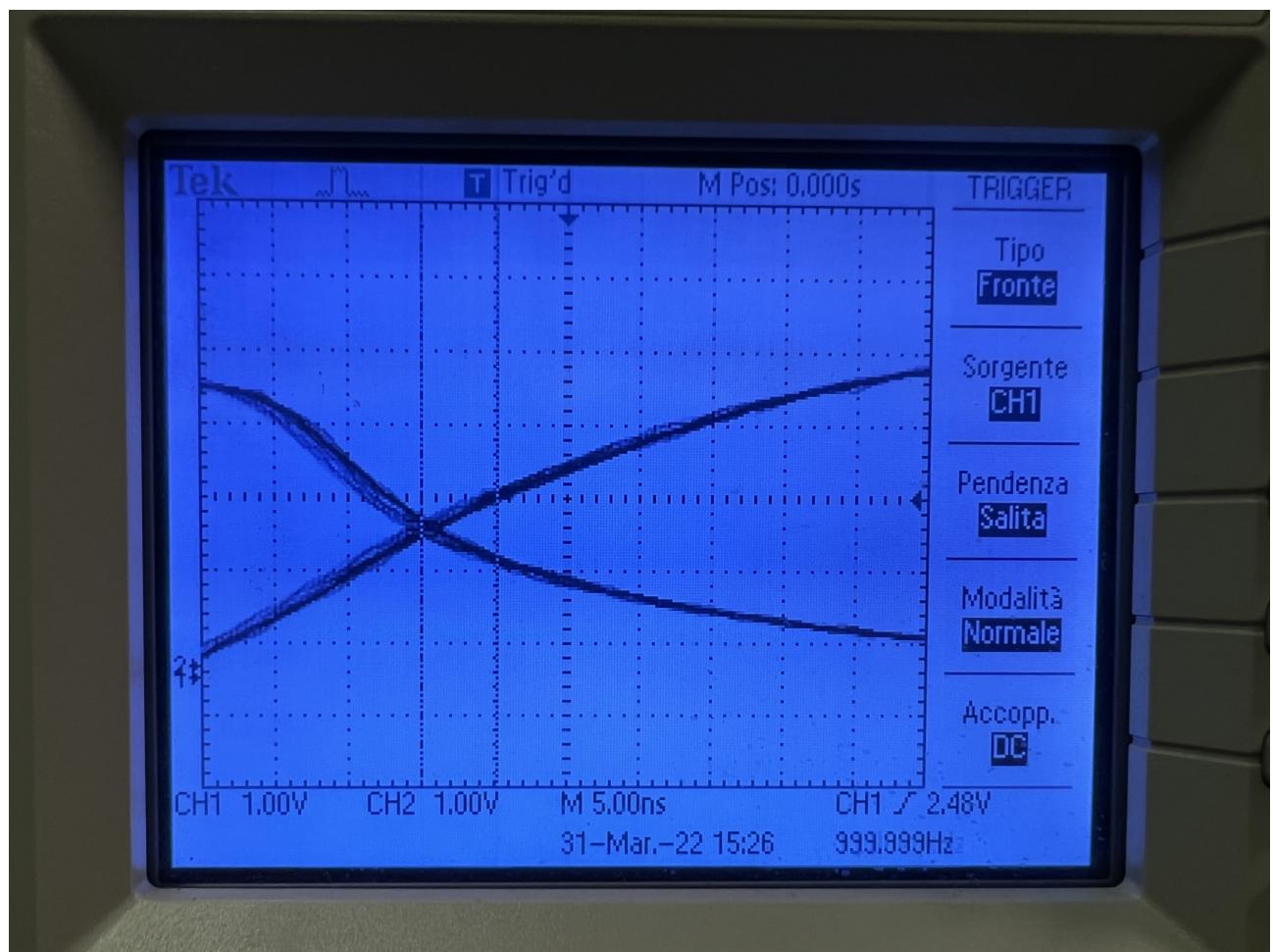


Figura 10: Acquisizione tramite oscilloscopio digitale della transizione da H a L per il secondo integrato

Dalle misure prese con i cursori dell'oscilloscopio (a cui associamo come incertezza il contributo dato dalle specifiche del datasheet, tenendo conto dell'instabilità delle tracce sullo schermo) troviamo

$$t_{PLH} = 5.2 \pm 1.2 \text{ ns}$$

$$t_{PHL} = 25 \pm 2 \text{ ns}$$

$$t_{PLH} = 7.0 \pm 1.2 \text{ ns}$$

$$t_{PHL} = 22 \pm 2 \text{ ns}$$

3.c Confronto con i valori attesi

Le misure dei tempi di propagazione nella transizione basso->alto risultano compatibili con i valori tipici, quindi sicuramente compresi entro l'intervallo massimo garantito dal costruttore.

Mentre per quanto riguarda la transizione opposta, abbassando la scala dei tempi osserviamo che i segnali visualizzati dall'oscilloscopio risultano simili alle curve caratteristiche di carica e scarica di un condensatore (riportata in sezione 3.c).

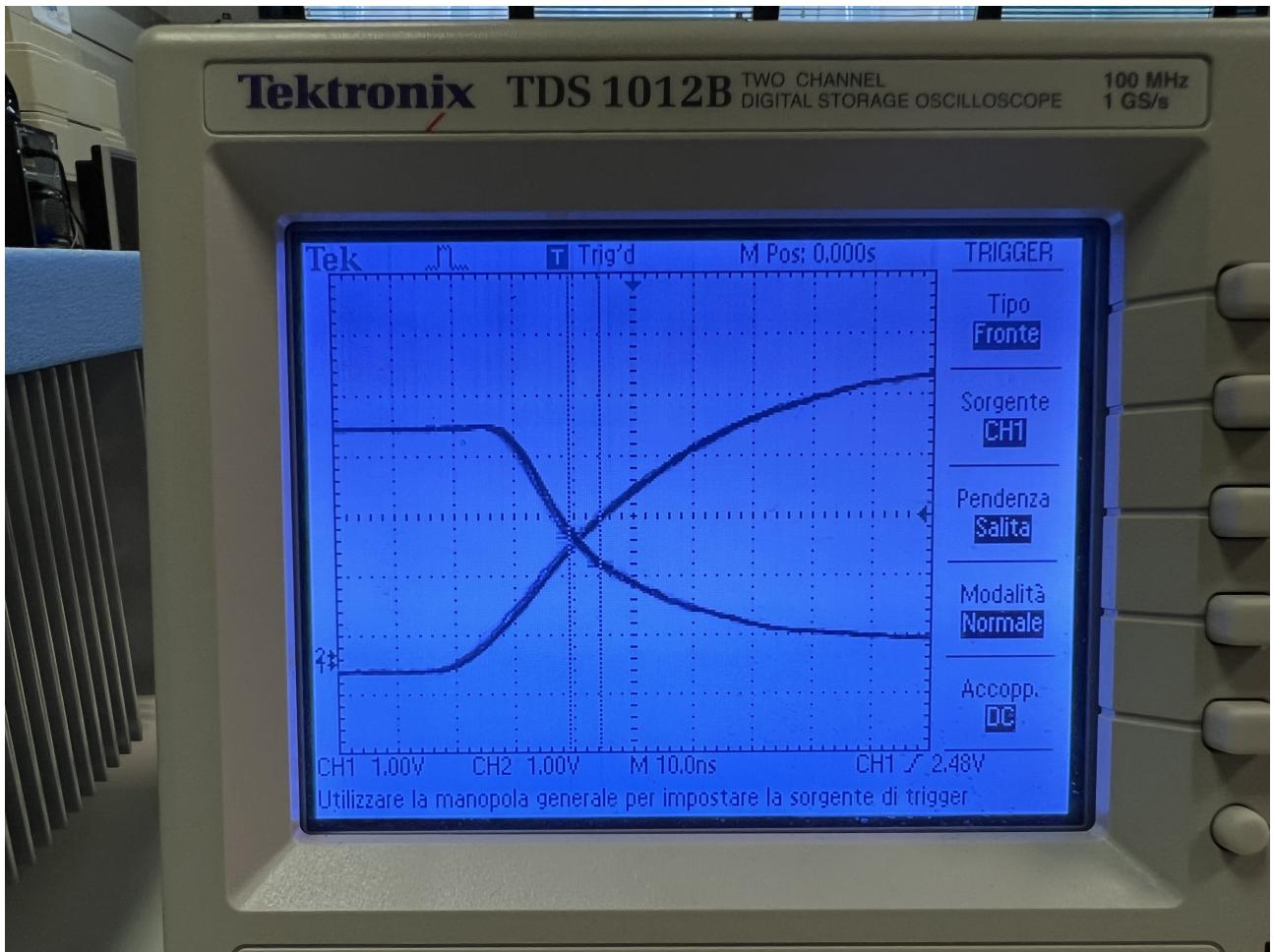


Figura 11: Acquisizione con oscilloscopio digitale con un fondo scala dei tempi più piccolo, in modo da osservare qualitativamente come la propagazione del segnale assomigli molto al grafico della tensione ai capi di un condensatore durante il ciclo di carica/scarica

Effettivamente a causa di questa discesa molto lenta del segnale in uscita abbiamo misurato dei tempi di propagazione t_{PHL} per il primo integrato al limite massimo riportato nel datasheet e addirittura oltre le specifiche per il secondo.

Rimane difficile valutare se questo tempo di discesa lungo sia dovuto ad accoppiamenti capacitivi spuri fra la basetta, i cavi di connessione o interne agli integrati usati, per cui se la definizione di tempo di propagazione non risulta ben posta prendiamo le nostre misure come stime indicative dei valori effettivi, che risultano compatibili con quanto dichiarato nel datasheet con uno scarto del 14 %.

Parte B: Circuiti logici elementari con sole porte NAND

La caratteristica più fondamentale delle porte NAND è la loro universalità, infatti è possibile realizzare qualsiasi tipo di circuito logico tramite combinazione di sole porte NAND (o NOR). In questa parte intendiamo costruire e verificare il funzionamento di circuiti equivalenti a porte OR, XOR e multiplexer a partire da soli chip NAND SN74LS00.

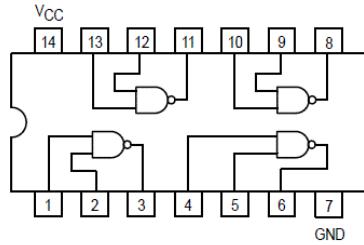


Figura 12: Schema e pedinatura del circuito integrato SN74LS00 quad-NAND

4 Tabella di verità

4.a Verifica statica della tabella di verità NAND

Vogliamo verificare preliminarmente che la porta NAND funzioni correttamente grazie alla funzione StaticIO inviando a DIO 0 e DIO 1 due segnali di tipo switch in tutte le loro possibili combinazioni. I primi due canali sono collegati agli ingressi di una stessa porta NAND sotto, mentre da DIO 2 in modalità LED misurare il valore in uscita da questa.

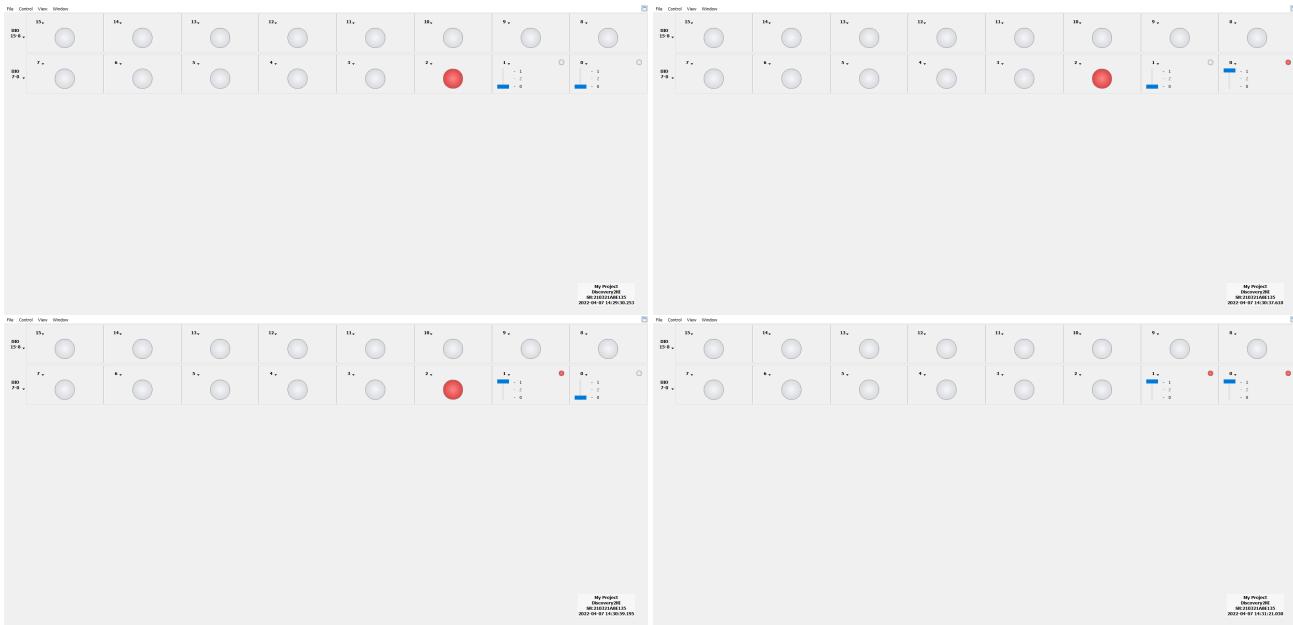


Figura 13: Acquisizione tramite StaticIO delle possibili configurazioni della porta NAND, in funzione dei valori di input

4.b Verifica del funzionamento con Logic Analyzer

Definiamo dentro lo strumento Patterns (generator) di WaveForms 2 segnali di clock; uno di frequenza 100 Hz (in uscita dalla porta DIO0) e l'altro a 200 Hz (in uscita dalla porta DIO1) in modo che la loro combinazione produca tutte le coppie di valori ottenibili con 2 bit.

Dunque si sono pilotati i due ingressi di una stessa porta NAND collegando i due clock ad un ingresso ciascuno e tramite lo strumento Logic (Analyzer) si sono acquisiti gli andamenti nel tempo dei due segnali in

ingresso e di quello in uscita dalla porta studiata (riportata in fig. 14) Il circuito risulta essere funzionante e

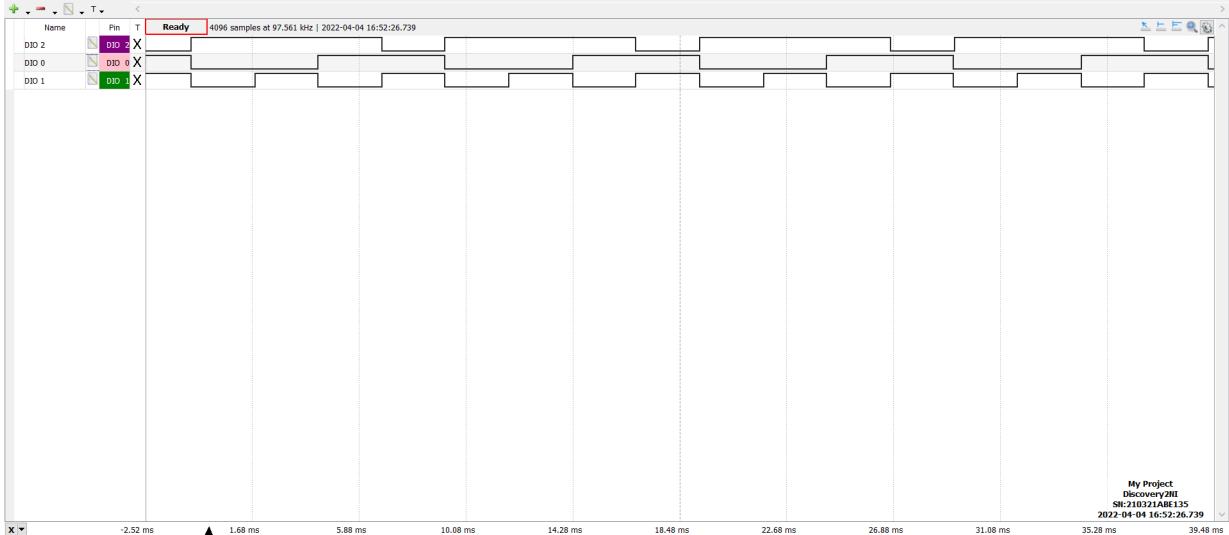


Figura 14: Acquisizione di Logic di una porta NAND guidata da due segnali da DIO0 (100 Hz) e DIO1 (200 Hz); in uscita viene letto da DIO2

l'output risulta essere L se e solo se i pin 1 e 2 valgono entrambi H, proprio come da aspettativa.

5 Costruzione di circuiti con porte NAND

5.a Porta OR

Come primo circuito si costruisce una porta OR: detti A e B gli ingressi e Y l'uscita, la funzione OR nella notazione dell'algebra di Boole si indica come

$$Y = A + B$$

Sfruttando la legge di De Morgan si ottiene la stessa relazione in termini di operatori NAND:

$$Y = A + B = \overline{\overline{A} \cdot \overline{B}} \quad (1)$$

Sapendo che per ottenere un NOT basta collegare lo stesso ingresso a un NAND gate, si può disegnare lo schema logico del circuito come in fig. 15:

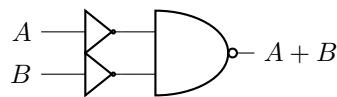


Figura 15

Che corrisponde nella nostra implementazione reale con l'integrato SN74LS00 allo schema circuitale in fig. 16.

Utilizzando le funzioni Patterns e Logic abbiamo inviato dai pin DIO0 e DIO1 2 segnali di clock di frequenza 50 e 100 Hz e abbiamo utilizzato la porta DIO2 per verificare che l'output del circuito combaciasse con i valori attesi dalla tabella di verità della funzione logica OR (come si vede in fig. 17) Si vede quindi che il circuito ha il funzionamento aspettato con l'output L se e solo se i due input sono entrambi L.

5.b Circuito selettore a due vie (multiplexer)

Realizziamo un circuito che permetta di assegnare all'uscita il valore di uno dei due ingressi a singolo bit tramite il valore di un terzo ingresso. Indichiamo con A , B e C gli ingressi e con Y l'uscita. In notazione logica il circuito implementa la funzione

$$\begin{cases} C = 0 \implies Y = A \\ C = 1 \implies Y = B \end{cases}$$

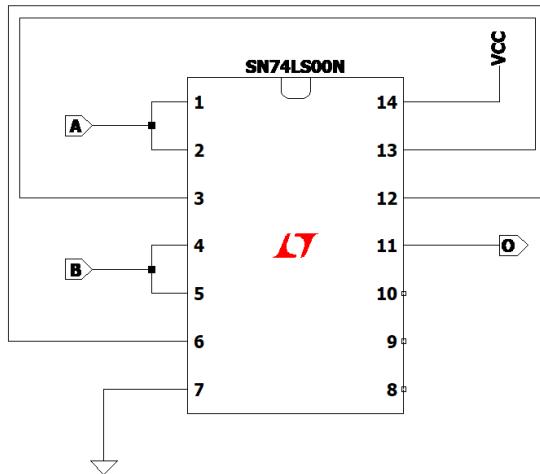


Figura 16: Schema circuitale utilizzato per costruire un OR GATE: A e B sono i segnali di input, mentre $O \equiv Y$ qui è l'output

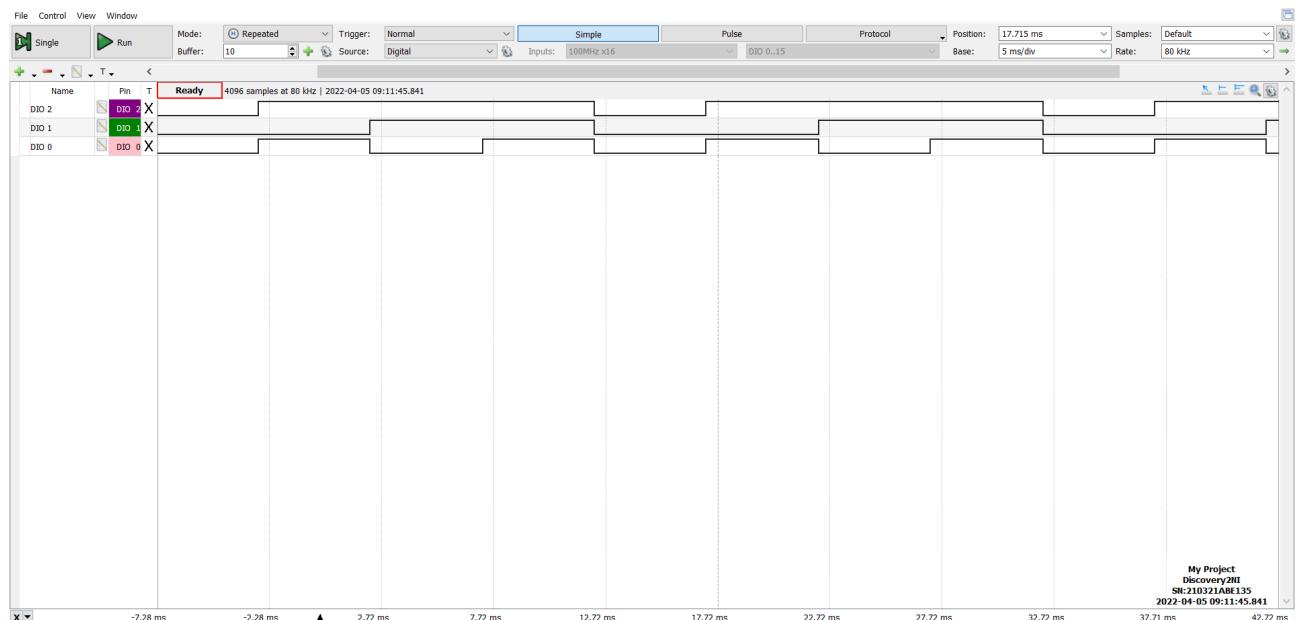


Figura 17: Acquisizione di Logic per il circuito OR costruito tramite NAND: DIO0 e DIO1 sono i segnali in input, DIO2 è l'output

AB				
	00	01	11	10
C	0	0	1	1
	1	0	1	0

Tabella 3: Mappa di Karnaugh per un circuito selettore a due vie

C	A	B		Y
0	0	0		0
0	0	1		0
0	1	0		1
0	1	1		1
1	0	1		0
1	1	0		0
1	1	1		1

Tabella 4: Tabella di verità del circuito selettore a due vie

che corrisponde alla mappa di Karnaugh riportata in tabella 3 e alla tabella di verità scritta in forma esplicita in tabella 4

Da queste possiamo ricavare che l'uscita del circuito deve soddisfare la relazione

$$Y = A \cdot \overline{C} + B \cdot C \quad (2)$$

sempre con la legge di De Morgan si arriva ad una riscrittura dell' eq. (2) in termini di NAND

$$Y = A \cdot \overline{C} + B \cdot C = \overline{\overline{(A \cdot \overline{C})}} \cdot \overline{\overline{(B \cdot C)}}$$

Che si traduce nello schema logico e nell'implementazione reale del circuito riportate in fig. 18 e fig. 19 rispettivamente

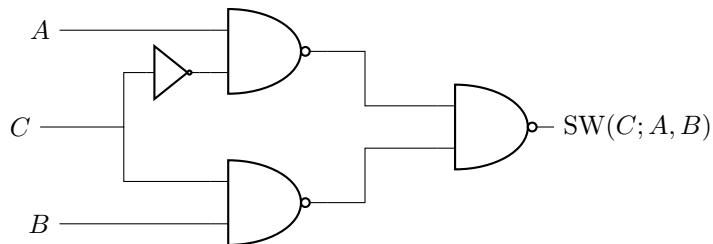


Figura 18: Diagramma logico del circuito selettore

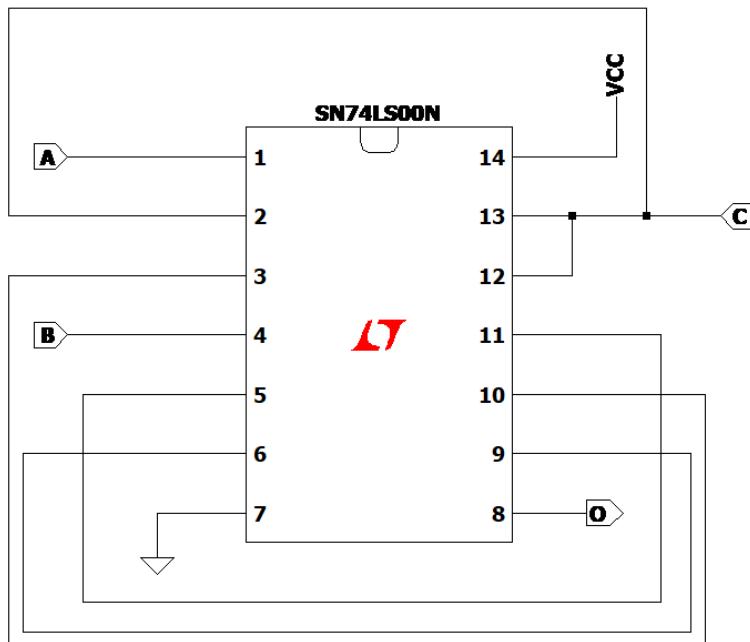


Figura 19: Schema circuitale di un multiplexer a 2 input, controllato dal valore logico di switch C

Per dimostrare il corretto funzionamento del circuito riportiamo la schermata di configurazione di Patterns, in cui si vede come sono stati impostati gli ingressi e l'acquisizione di Logic, da cui si può verificare il corretto funzionamento in uscita dal circuito al variare dei valori logici dei 3 ingressi.

Si inviano all'ingresso del circuito un segnale di clock di frequenza 100 Hz (B), uno a 50 Hz A e infine uno a 25 Hz C per esplorare tutte le possibili combinazioni possibili per i 3 bit di ingresso (cfr. fig. 20). Osserviamo (da fig. 21) che il circuito funziona come da aspettative, restituendo cioè il segnale in A nel caso in cui $C = 0$ e il valore di B quando $C = 1$.

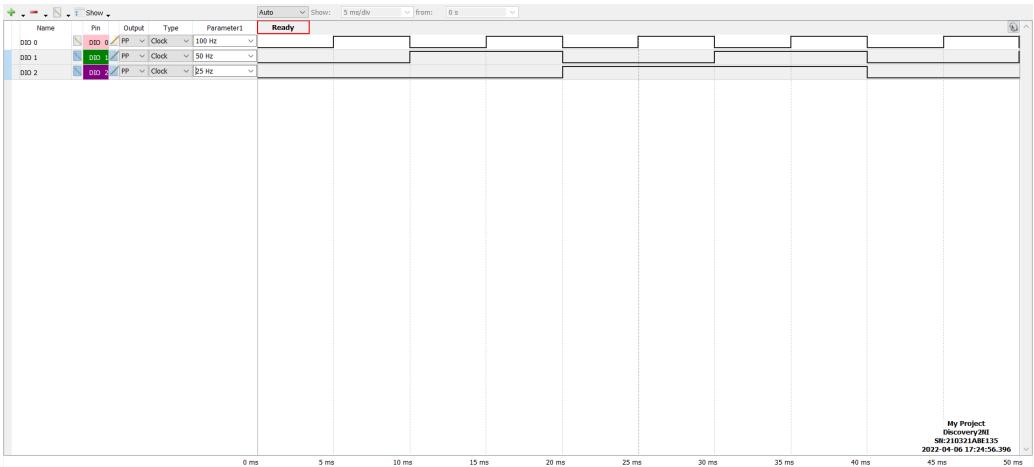


Figura 20: Schermata di configurazione degli ingressi in Patterns al multiplexer: DIO 0 \equiv B, DIO 1 \equiv A, DIO 2 \equiv C.

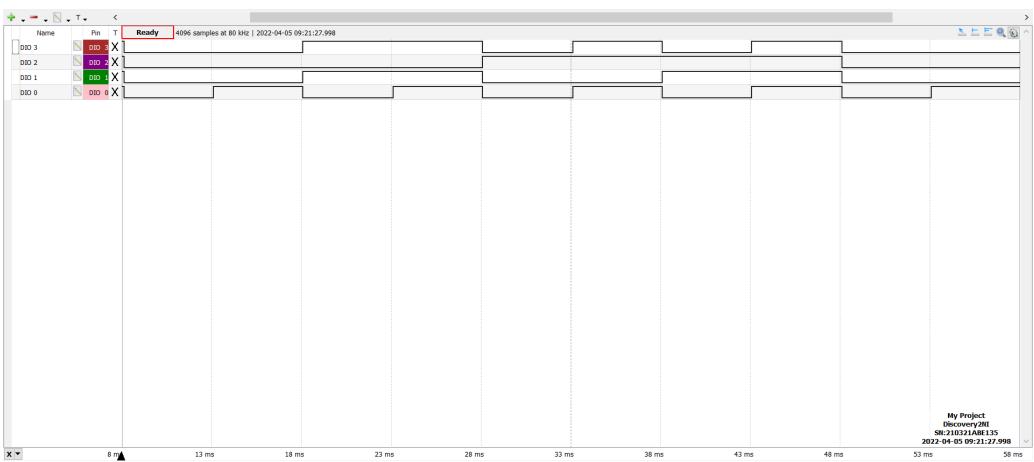


Figura 21: Acquisizione di Logic per il multiplexer: DIO 0 \equiv B, DIO 1 \equiv A, DIO 2 \equiv C, DIO 3 \equiv Y.

5.c Porta XOR

Un circuito XOR si può realizzare in maniera ottimale con 4 porte NAND partendo dalla sua equazione caratteristica e manipolandola con le leggi di De Morgan:

$$\begin{aligned}
 A \oplus B &= (A \cdot \bar{B}) + (\bar{A} \cdot B) = (A \cdot \bar{A} + A \cdot \bar{B}) + (B \cdot \bar{B} + \bar{A} \cdot B) \\
 &= A \cdot (\bar{A} + \bar{B}) + B \cdot (\bar{B} + \bar{A}) = A \cdot \overline{(A \cdot B)} + B \cdot \overline{(B \cdot A)} \\
 &= \overline{A \cdot \overline{(A \cdot B)}} \cdot \overline{B \cdot \overline{(B \cdot A)}}
 \end{aligned}$$

Questo corrisponde allo schema logico e all'implementazione reale con il nostro chip riportate nelle fig. 22 e fig. 23.

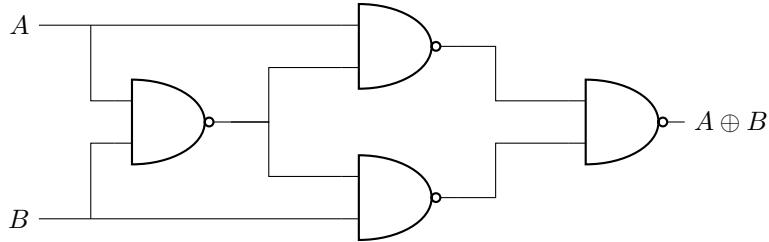


Figura 22: Circuito logico di una porta XOR costruita con sole porte NAND

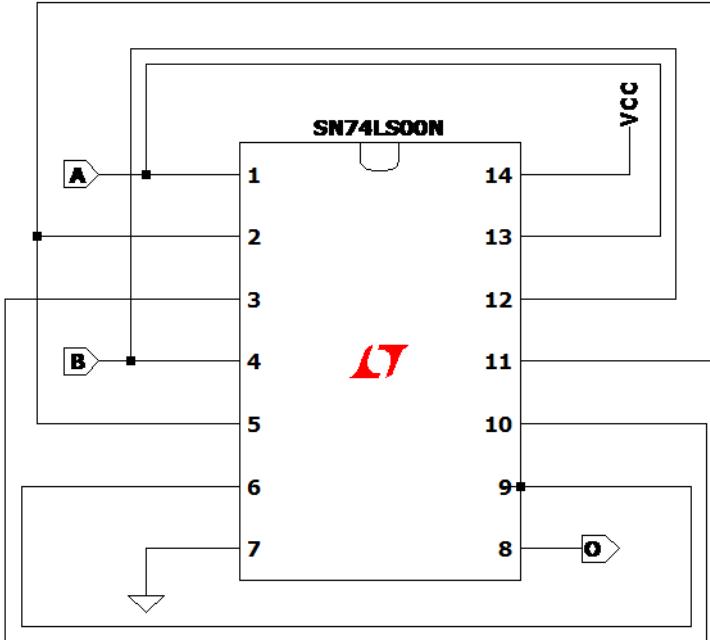


Figura 23: Schema circuitale utilizzato per costruire uno XOR GATE tramite NAND

Verifichiamo il buon funzionamento del circuito XOR inviando ai due ingressi gli stessi segnali di clock usati per la porta OR e osservando i valori dell'uscita Y con il Logic Analyzer in maniera del tutto analoga a prima. Dall'acquisizione in 24 osserviamo che il circuito riproduce il comportamento atteso, in quanto l'output risulta essere L se e solo se entrambi gli input A e B assumono lo stesso valore.

Parte C: Circuiti logici complessi a più chip

6 Convertitore Gray-Binario

Vogliamo costruire un circuito che converte un valore a 4 bit dalla codifica Gray in codice Binario utilizzando un integrato di tipo SN74LS86 a porte XOR (in fig. 25).

Un convertitore Gray-Binario può essere schematizzato come in fig. 26, di nuovo il nostro obiettivo è verificare che tale circuito funzioni correttamente.

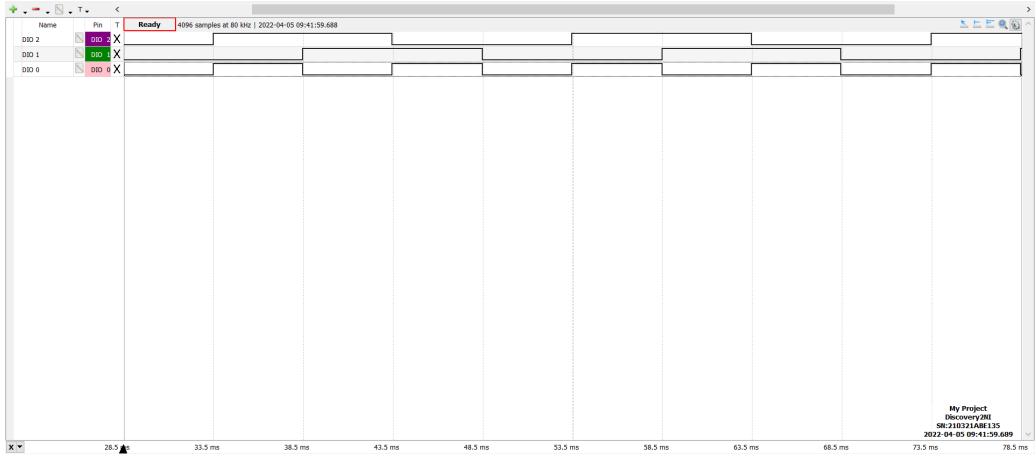


Figura 24: Acquisizione Logic della Porta XOR: DIO 0 e DIO 1 sono i segnali di input, DIO 2 è l'uscita dallo XOR

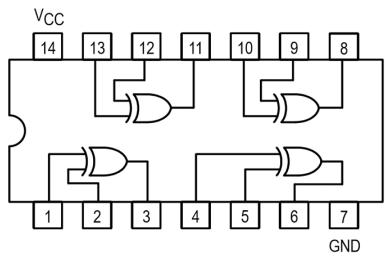


Figura 25: Schema e pedinatura del circuito integrato SN74LS86 quad-XOR

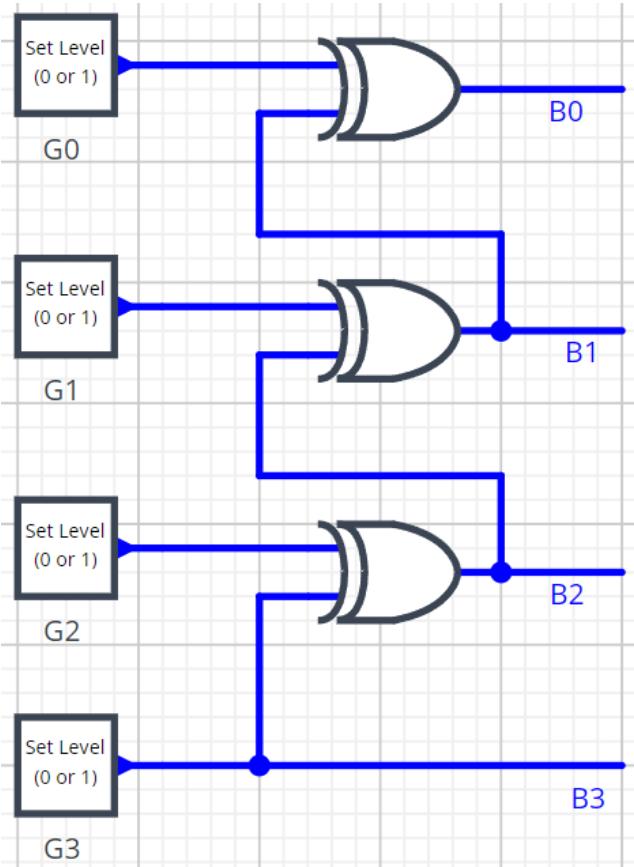


Figura 26: Schema logico di un circuito convertitore Gray->Binario

6.a Verifica del disegno circuitale

Preliminamente possiamo calcolare i valori attesi in uscita per almeno 4 diversi stati in ingresso come verifica dello schema circuitale in fig. 26

G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	1	1	1	1	0	1	0

6.b Verifica del corretto comportamento del convertitore G->B

Come si può vedere esplicitamente dalla 5 il codice Gray differisce dal codice binario in quanto si passa da un numero intero al successivo modificando un solo bit per volta. Dunque dalla stessa possiamo vedere quali sono

Codice binario	Codice Gray
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

Tabella 5: Conteggio sequenziale a 4 bit nelle due codifiche.

i valori attesi per l'uscita del circuito al variare dei valori in ingresso.

Dunque, per conferma del corretto funzionamento del convertitore come prima possiamo confrontare le uscite ottenute da un'acquisizione con Logic Analyzer con i valori riportati in tabella 5 generando all'ingresso con Patterns un contatore Gray.

Per una scala dei tempi ristretta, abbiamo triggerato quando il MSB del bus a codice Gray cade da H a L: in questo modo riusciamo a osservare la transizione da 15 a 0 (1000 in Gray equivale a 1111 in binario) e in particolare il tempo di propagazione che intercorre tra quando il segnale in ingresso scende a 0 e quando effettivamente in uscita si registra corrispondente valore in binario (riportiamo una serie di acquisizioni di questa transizione in fig. 28 e fig. 29)

Dalla figura fig. 29 notiamo che anche se il bit 3 del segnale Gray e il bit 3 del codice binario sono direttamente collegati tra loro (cioè senza nessuna porta logica tra i due), quest'ultimo percepisce un ritardo pari a circa la risoluzione temporale dell'AD2 (10 ns).

Inoltre i successivi bit del codice binario (2, 1 e infine 0) risultano anch'essi ognuno in ritardo rispetto al bit precedente a cui è collegato in cascata da una porta logica XOR.

Controllando sul datasheet i tempi di propagazione per i gate XOR usati si ricava che nel caso in cui l'altro ingresso si trovi in stato logico basso, $t_{PHL,typ} = 10$ ns e $t_{PHL,max} = 17$ ns.

Considerando quindi che il ritardo totale (da quando il bit MSB Gray scende a 0, fino a quando il bit LSB Binario scende anch'esso a 0) risulta pari a circa 50 ± 10 ns, questo risulta compatibile con il ritardo atteso; ovvero quello accumulato dalle 3 porte XOR di (≈ 30 ns) in aggiunta a quello del primo bit binario (MSB) di 10 ns.

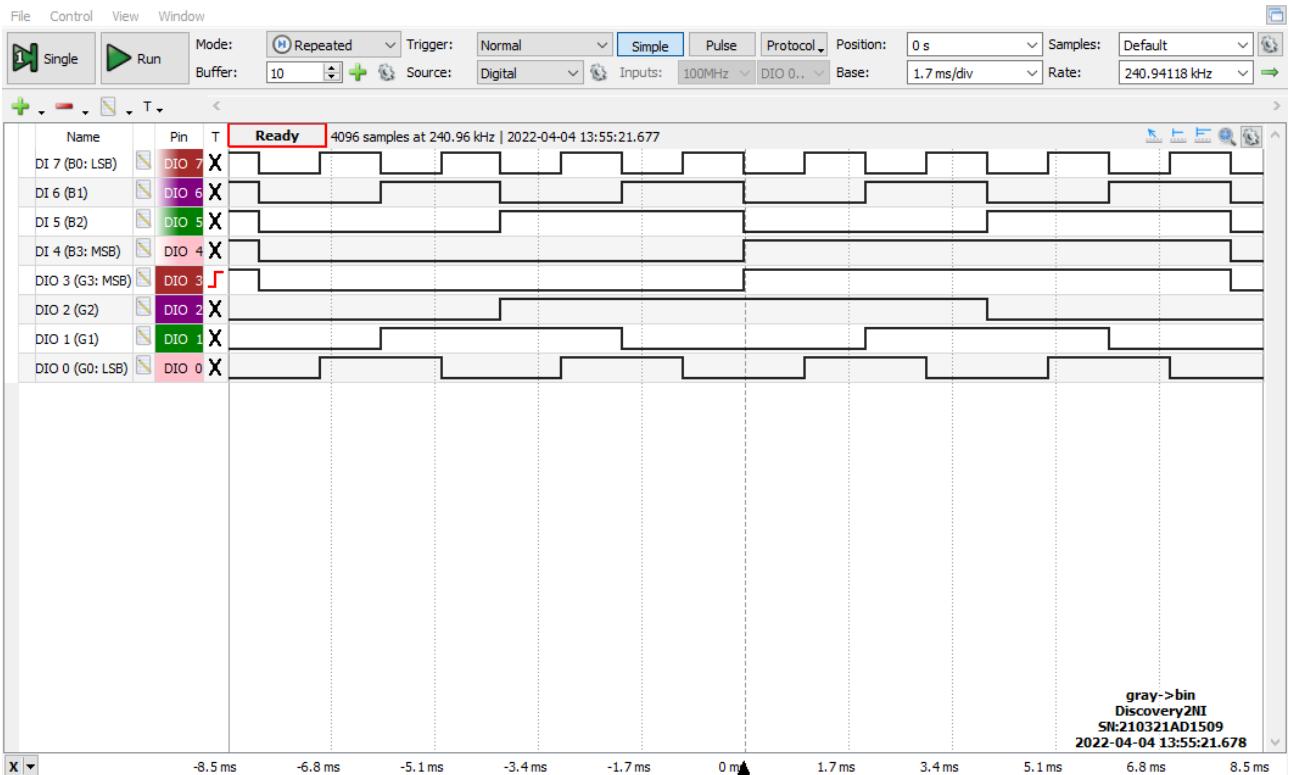


Figura 27: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dal convertitore Gray-binario.

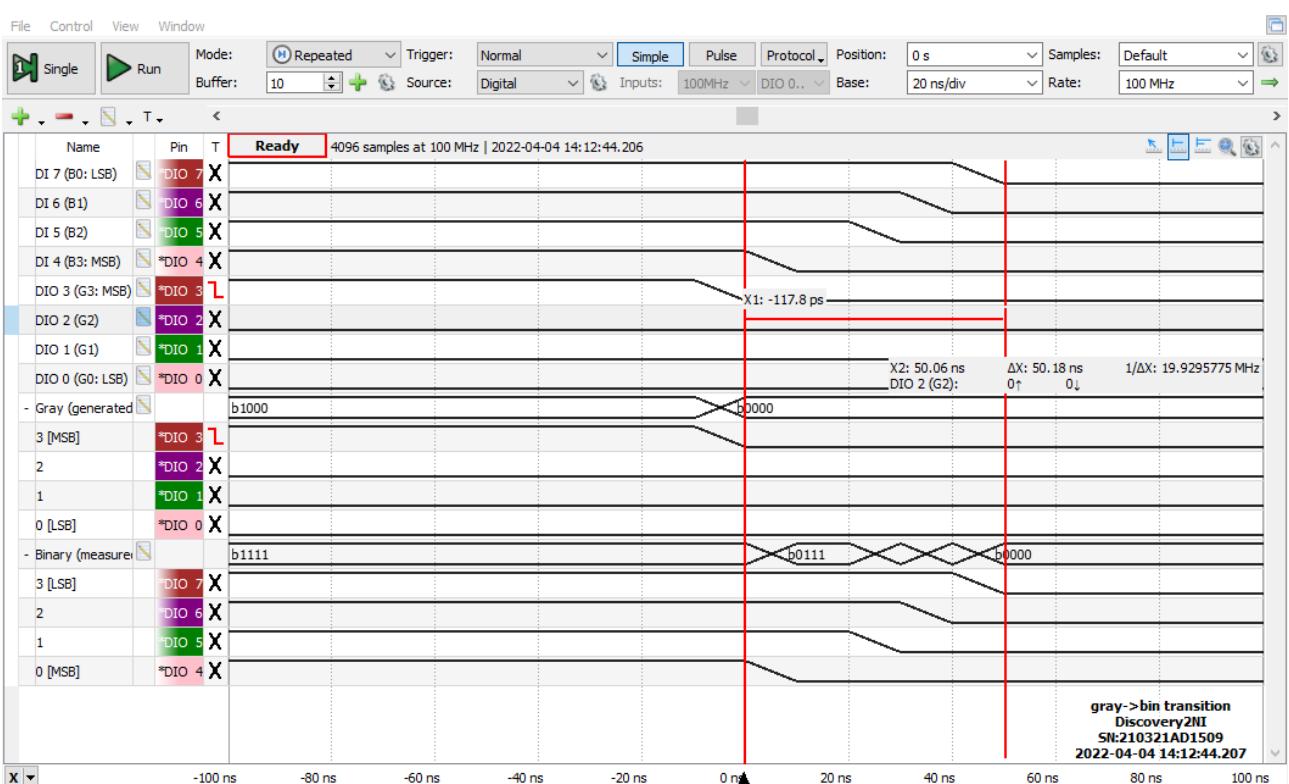


Figura 28: Acquisizione del Logic Analyzer durante la transizione dal numero 15 (b1111) al numero 0 (b0000) su scala dei tempi pari a 20 ns.

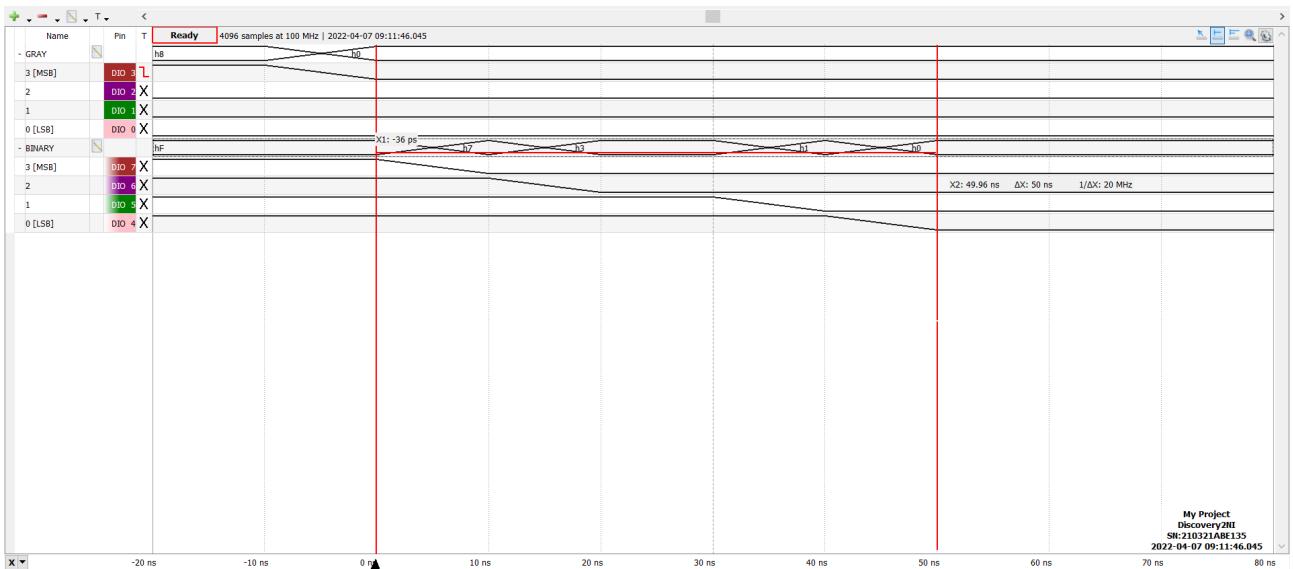


Figura 29: Transizione dal 15 allo 0 su scala temporale pari a 10 ns.

Il ritardo "fantasma" misurato con l'AD2 presente dal momento in cui il bit 2 Binario scende a 0 fino a quando anche il bit 1 scende a 0 (indicativamente in fig. 29 tra i 20 e i 30 ns dopo il trigger) si ipotizza sia da attribuire alla scarsa risoluzione dell'AD2 e ad una porta logica con tempo di propagazione particolarmente più alto rispetto alle altre (da quanto osservato ci aspettiamo sia lo XOR che in uscita ha il bit B1).

7 Sommatore a 2 bit

7.a Costruzione e analisi del circuito

Vogliamo costruire un sommatore a due bit utilizzando le porte logiche contenute nei chip integrati SN74LS08 (quad-AND), SN74LS32 (quad-OR) e SN74LS86 (quad-XOR). Il circuito da noi costruito e studiato è riportato in fig. 34.

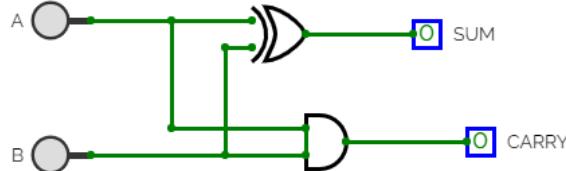


Figura 30: Schema logico del circuito di un half adder

7.b Verifica del funzionamento degli Half-Adder e Full-Adder

Si vuole quindi verificare il funzionamento dei due circuiti Half Adder e Full Adder schematizzati nelle fig. 30 e fig. 31.

Per verificare il funzionamento del primo è stato sufficiente generare con Patterns un bus contatore binario a 2 bit, inviare i segnali dei due bit a ciascuno degli ingressi dell'HA e osservarne il comportamento con Logic (come in fig. 32).

Per verificare il funzionamento del Full Adder abbiamo generato con Patterns un bus contatore (sempre in binario) a 3 bit: 2 di questi (i due bit meno significativi) sono stati utilizzati come ingressi A e B , mentre il MSB è stato utilizzato come bit di CARRY IN. Riportiamo l'andamento dell'uscita del FA in funzione dei segnali in ingresso acquisita dal Logic Analyzer in fig. 33. Dai risultati ottenuti con le acquisizioni di Logic si verifica che entrambi i circuiti funzionano come ci si aspetta.

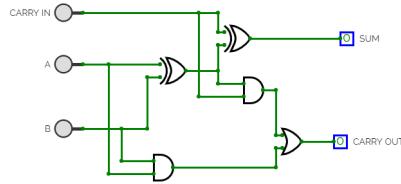


Figura 31: Schema logico del circuito di un full adder.

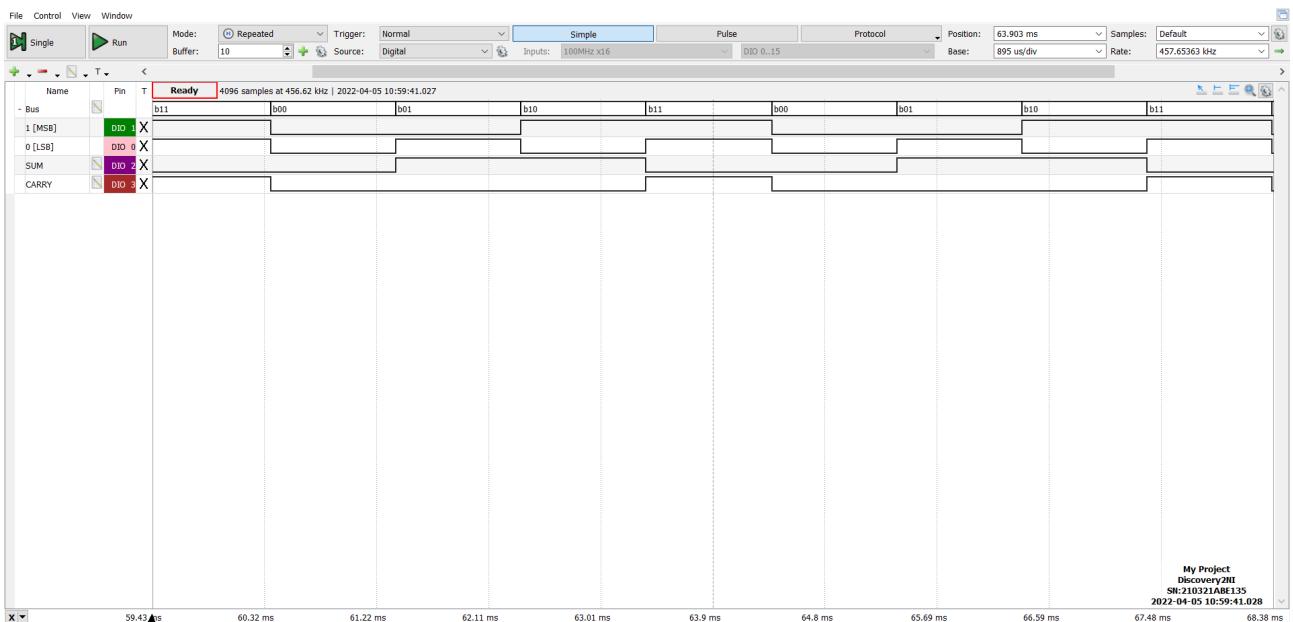


Figura 32: Acquisizione di Logic del funzionamento di un Half Adder

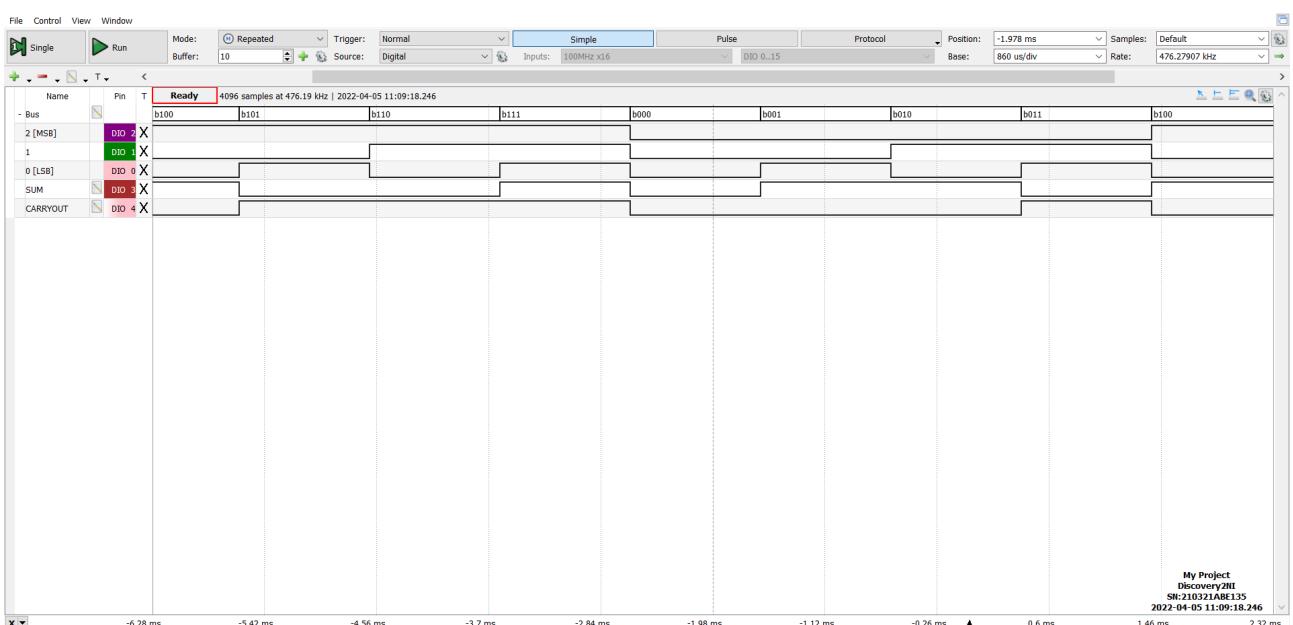


Figura 33: Acquisizione con Logic del funzionamento di un Full Adder

7.c Verifica del comportamento da sommatore con resto

Abbiamo collegato l'uscita CARRY OUT dell'half adder all'ingresso CARRY IN del full adder: così facendo si ottiene un sommatore binario a 2 bit con bit di riporto (che andrà ad indicare un eventuale overflow). Riportiamo lo schema del circuito per completezza in fig. 34.

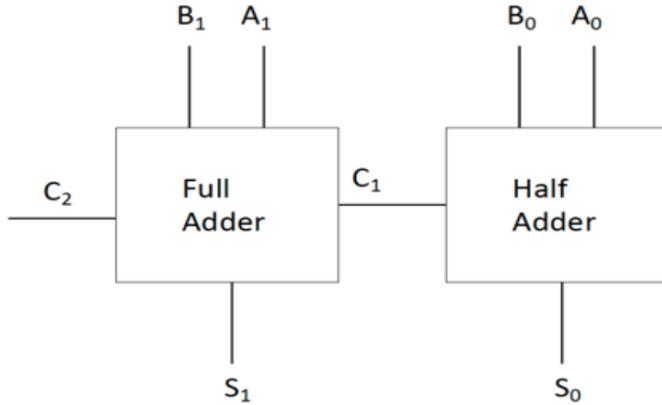


Figura 34: Schema circuitale di un sommatore a 2 bit

A questo punto si procede con la verifica di funzionamento del circuito. Per fare questo si genera con Patterns un Bus contatore a 4 bit che utilizzeremo per inviare segnali corrispondenti a numeri in codice binario in ingresso al sommatore -facendo attenzione a inserire nell'Half Adder i due bit meno significativi di entrambi i numeri, mentre al Full Adder i due bit più significativi- (in particolare i 2 bit meno significativi del bus formeranno il numero A , mentre i restanti rappresenteranno l'intero B .) Si verifica come nei punti precedenti che il circuito

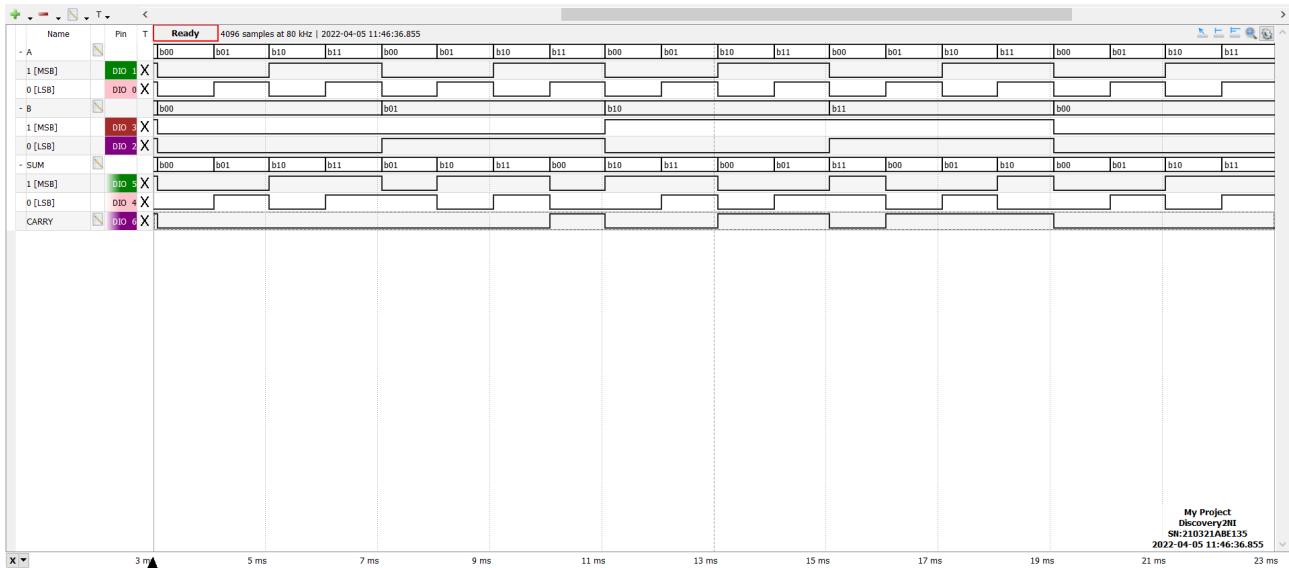


Figura 35: Acquisizione con Logic per il sommatore a 2 bit: DIO 0 e DIO 1 rappresentano il numero A ; DIO 2 e DIO 3 rappresentano B ; DIO 4 (LSB), DIO 5 (MSB) e DIO 6 (bit di overflow) rappresentano il risultato della somma.

sommatore si funziona come atteso dai risultati dell'analisi con Logic riportati in fig. 35.

7.d Verifica funzionamento con led e ROM

Aggiungiamo al circuito 4 led verdi e un led rosso e li colleghiamo all'AD2. Per controllare il loro funzionamento definiamo con ROM in Patterns una tabella di verità, riportata in fig. 36, tale per cui ad ogni ciclo si illuminino un numero di led pari al valore della somma.

Il led rosso viene usato per indicare eventuale overflow, i.e. la possibilità che il risultato sia maggiore o uguale a 4.

	DIO 6	DIO 5	DIO 4	DIO 11	DIO 10	DIO 9	DIO 8	DIO 7
1	1	X	X	1	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	1
4	0	1	0	0	0	0	1	1
5	0	1	1	0	0	1	1	1
6								

Figura 36: Tabella delle verità usata per il controllo dei led.

Si è fatta quindi un’ultima acquisizione tramite Logic includendo anche i segnali inviati ai LED (riportata in fig. 37) Compatibilmente con quanto ci si aspettava l’ultimo LED verde non si accende mai (dato che con 2

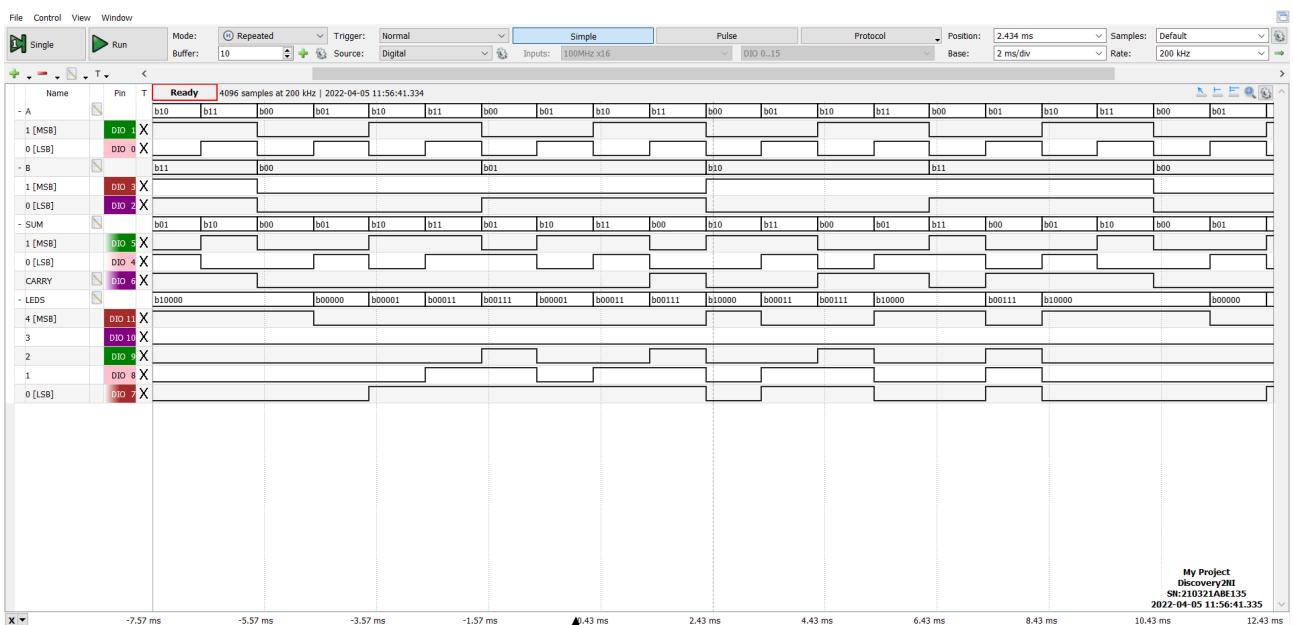


Figura 37: Acquisizione con Logic dell’andamento nel tempo dei vari segnali per il circuito Sommatore a 2 bit. Il risultato in uscita dal circuito pilotano i led secondo la tabella di verità definita in fig. 36.

bit è possibile a rappresentare solo i numeri fino a 3) e osserviamo solo il LED rosso accendersi quando il bit di CARRY ha valore logico alto.

Conclusioni e commenti finali

Si è riusciti a verificare il corretto comportamento delle porte TTL studiate caratterizzandone le tensioni, correnti di operazione e tempi caratteristici di circuiti integrati come il SN7404. Inoltre, è stato possibile verificare il funzionamento di circuiti logici di diversa complessità costruiti con porte NAND, XOR, e OR e si è riusciti ad apprezzare l’effetto dei tempi di propagazione delle porte nella conversione dalla codifica Gray al binario.

Dichiarazione

I firmatari di questa relazione dichiarano che il contenuto della relazione è originale, con misure effettuate dai membri del gruppo, e che tutti i firmatari hanno contribuito alla elaborazione della relazione stessa.