POLITECNICO
MILANO 1863

Progetto di reti Logiche

Funzione seno in virgola fissa

Lorenzo Bardelli 10831941

Sommario

Introduzione

Il modulo progettato calcola il seno di un angolo ricevuto in input mediante interpolazione lineare basandosi su valori noti della funzione.

I valori prestabiliti del seno sono relativi agli angoli multipli di 8 del primo quadrante oltre che agli angoli pari a 89° e 90°.

Il modulo ricorre inoltre le seguenti identità trigonometriche:

$$\sin(\theta) = \sin(180 - \theta) \ \ se \ \ \theta \in (90; 180]$$
$$\sin(\theta) = -\sin(\theta - 180) \ \ se \ \theta \in (180; 270]$$
$$\sin(\theta) = \ -\sin(360 - \theta) \ \ se \ \theta \in (270; 360]$$

Grazie alle quali è sufficiente conoscere i valori della funzione da 0° a 90° per poter ricondursi al risultato per qualsiasi angolo.

L'angolo in input è intero e può variare secondo la specifica da 0 a 359, è rappresentabile quindi su 9 bit (sono rappresentabili anche valori superiori a 359 ma produrranno una segnalazione di risultato invalido).

Il valore del seno in output è rappresentato su 10 bit in virgola fissa così organizzati:

i primi 2 bit più significativi per la parte intera, i restanti 8 bit per la parte decimale.

La funzione utilizzata per l'interpolazione è quella classica della retta passante per due punti:

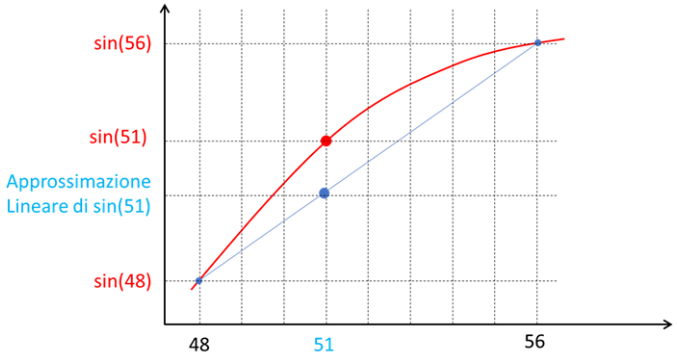
$$y = y_0 + \frac{(x - x_0)(y_1 - y_0)}{8}$$

Dove

y è il valore approssimato del seno al per l'angolo x
y_0, y_1 sono i valori della funzione per l'angolo multiplo di 8 precedente e successivo all'angolo x
x_0 è il multiplo di 8 precedente per x

Si noti che non compare il termine $x_1 - x_0$ perché, dati gli intervalli considerati, è sempre = 8.

Esempio di interpolazione per $\sin 51$:



Specifica

Il sistema progettato è composto da diversi sotto moduli, realizzati principalmente con componenti standard, che implementano le seguenti macro-procedure:

Un componente trasla l'angolo in base alle necessità e genera i flag di segno e di errore.

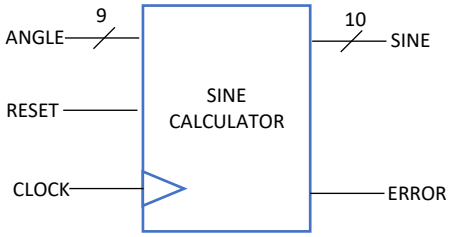
Un altro identifica i multipli di 8 che definiscono il range dell'approssimazione e gestisce la casistica legata ad angoli superiori a 87°.

Delle lookup-table restituiscono i valori del seno per gli angoli così trovati e l'interpolatore lineare calcola il valore della funzione nel punto (angolo) desiderato (in caso di angoli = 88°, 89° , 90° e loro associati restituisce il valore del seno senza interpolare).

Questo ultimo modulo si occupa anche di cambiare di segno il valore approssimato, in base al flag calcolato a monte di tutta la procedura.

Interfaccia del sistema

L'interfaccia del top-module è la seguente:



nome	dimensione	direzione
ANGLE	9 bit	Input
RESET	1 bit	Input
CLOCK	1 bit	Input
SINE	10 bit	Output
ERROR	1 bit	Output

Sono presenti gli ingressi per i segnali di CLOCK e RESET (asincrono).

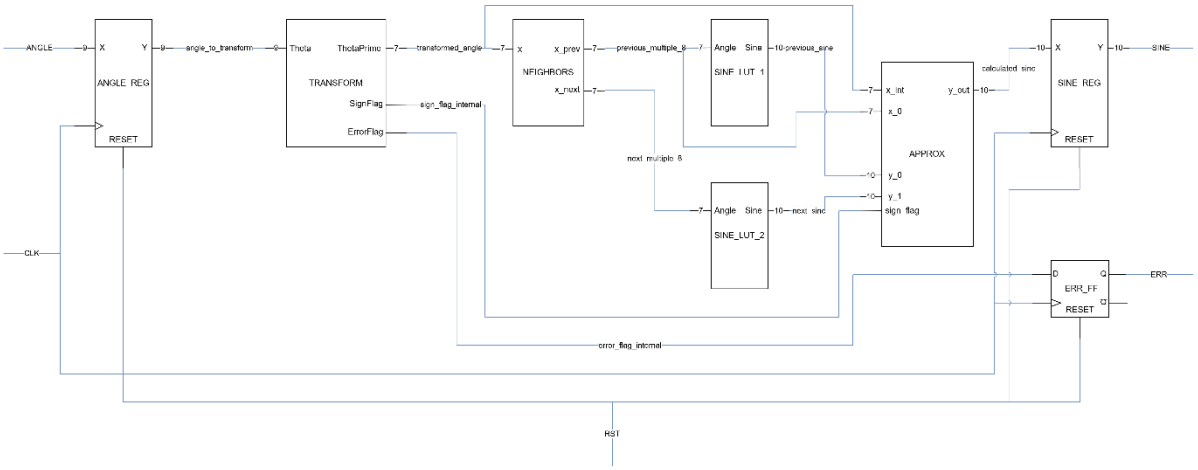
Il segnale ANGLE da 9 bit è l'ingresso su cui fornire il valore intero dell'angolo di cui si vuole calcolare il seno.

Il segnale SINE è l'uscita a 10 bit su cui viene fornito il valore decimale segnato del seno calcolato, rappresentato in fixed-point (2 bit parte intera + 8 bit parte decimale).

Il bit di ERROR segnala un errore dovuto ad un angolo in ingresso fuori dal range [0;359].

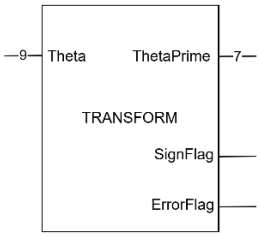
Architettura del sistema

La seguente figura rappresenta l'architettura del componente come interconnessione di sottocomponenti:



Angle Transformer

Questo modulo si occupa di traslare l'angolo inserito per riportarlo nel range 0-90 e generando il flag di segno associato.



nome	dimensione	direzione
Theta	9 bit	Input
ThetaPrime	7 bit	Output
SignFlag	1 bit	Output
ErrorFlag	1 bit	Output

Possiede un input su 9 bit su cui va fornito l'angolo intero in codifica binaria.

In output restituisce l'angolo traslato in binario e i flag di segno e di errore.

Utilizza adder e comparatori per calcolare le differenze con gli angoli che delimitano i quaranti e stabilire in quale range si trova l'angolo.

La logica di selezione infine riporta in uscita il risultato corretto secondo questo criterio:

R3 se IS_THIRD_ QUADRANT = 0[1]

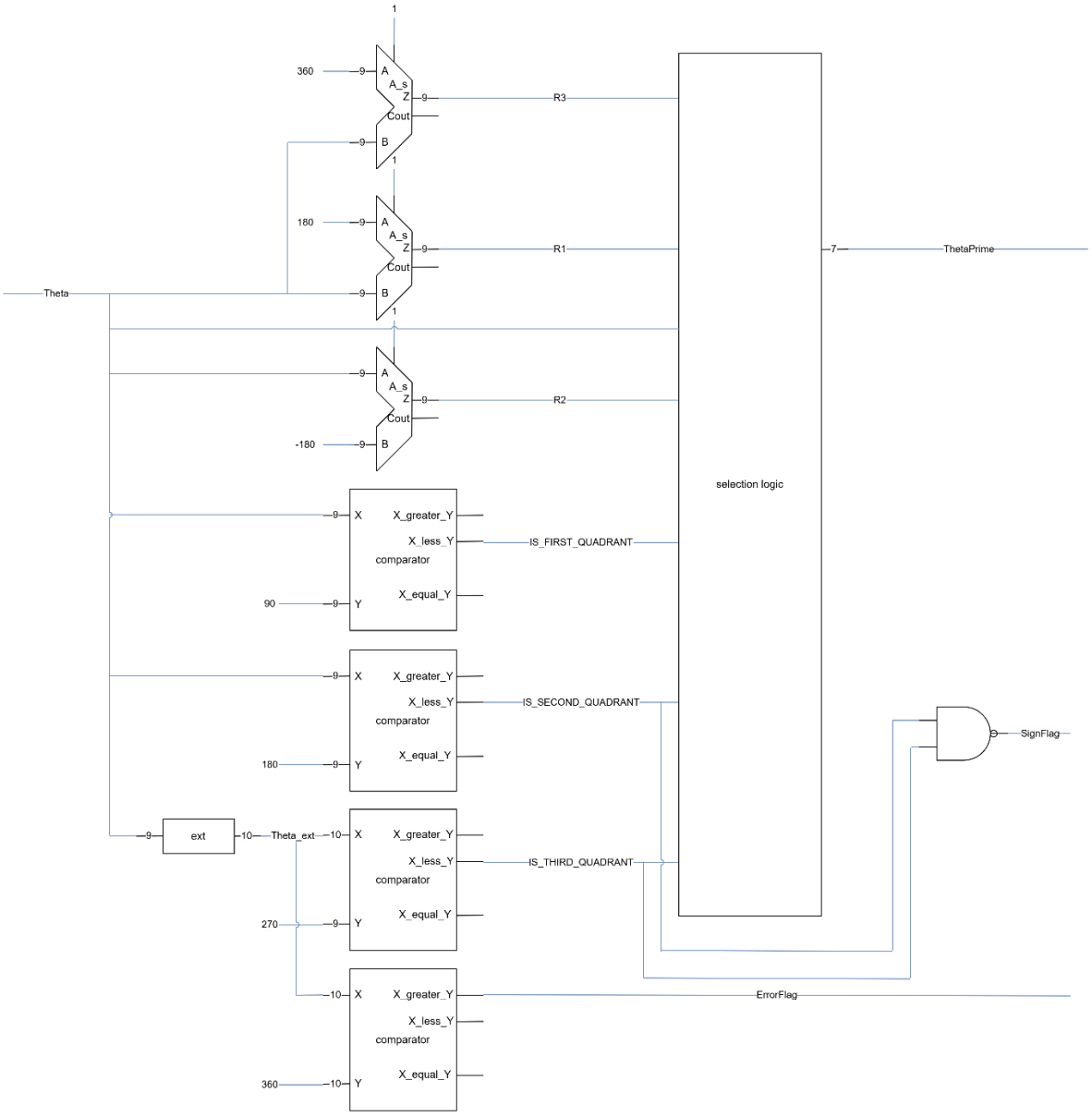
R2 se IS_SECOND_ QUADRANT = 0

R1 se IS_FIRST_ QUADRANT = 0

Theta altrimenti.

Il flag di segno è sollevato se l'angolo appartiene al III o IV quadrante (vedi identità trigonometriche) e quindi se IS_SECOND_QUADRANT = 0 o IS_THIRD_QUADRANT = 0.

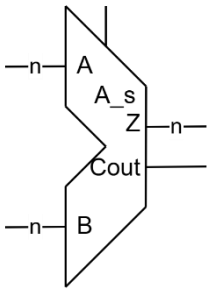
Di seguito la struttura interna del traslatore:



[1] I flag interni che indicano il quadrante dell'angolo sono da interpretare nel seguente modo:
= 0 se l'angolo è in un quadrante superiore a quello in esame,
= 1 se l'angolo è < dell'angolo che delimita il quadrante considerato.

CLA

Per realizzare il traslatore sono stati utilizzati dei carry-lookahead adder data loro rapidità.

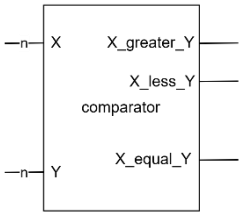


nome	dimensione	direzione
A	n bit	Input
B	n bit	Input
Add_sub	1 bit	Input
Z	n bit	Output
Cout	1 bit	Output

l'architettura è stata resa generica a n bit per permettere il riutilizzo in altri componenti ed è stata implementata per eseguire anche la sottrazione.

Comparator

questo componente stabilisce se l'input X sia >,< o = all'input Y, entrambi numeri binari generici su n bit.



nome	dimensione	direzione
X	n bit	Input
Y	n bit	Input
X_greater_Y	1 bit	Output
X_less_Y	1 bit	Output
X_equal_Y	1 bit	Output

Viene utilizzato nello specifico per controllare il quadrante dell'angolo e per selezionare l'output dell'interpolatore nel caso di angoli = 89°,90° che non necessitano di interpolazione.

L'architettura è stata resa generica dato che nei due componenti che lo utilizzano si necessita di confronti di numeri di dimensioni differenti.

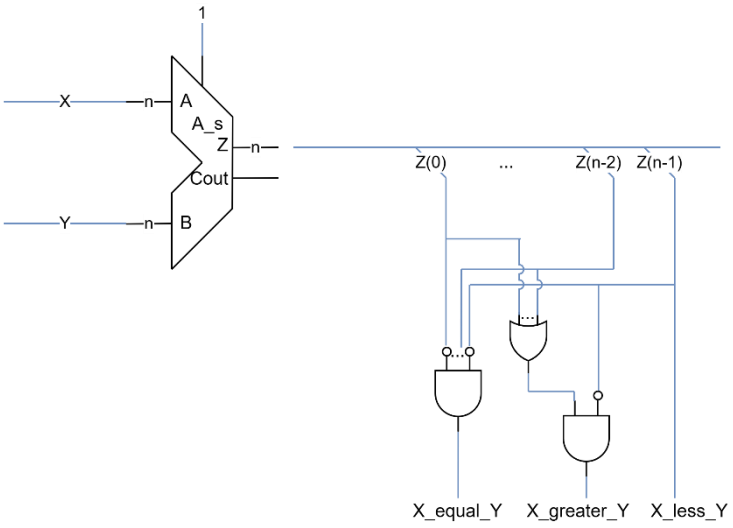
le uscite logiche sono calcolate con le seguenti formule:

$x > y \leftrightarrow x - y > 0$, che a livello di bit significa valutare: $\overline{z_{n-1}} * (Z_0 + Z_1 + \cdots + Z_{n-2})$

$x < y \leftrightarrow x - y < 0$, che a livello di bit significa valutare: z_{n-1}

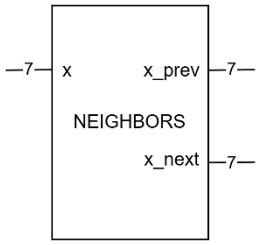
$x = y \leftrightarrow x - y = 0$, che a livello di bit significa valutare: $(\overline{Z_0} * \overline{Z_1} * \ldots * \overline{Z_{n-1}})$

struttura interna del componente:



Multiple 8

Il modulo multiple 8 restituisce i multipli di 8 adiacenti al numero fornito.



nome	dimensione	direzione
x	7 bit	Input
x_prev	7 bit	Output
X_next	7 bit	Output

l'ingresso x è compreso tra 0 e 90 e rappresentato su 7 bit.

Le uscite sono anch'esse su 7 bit.

In caso l'ingresso sia superiore a 87 (quindi idealmente nei casi 88, 89 e 90) le uscite replicano semplicemente l'ingresso.

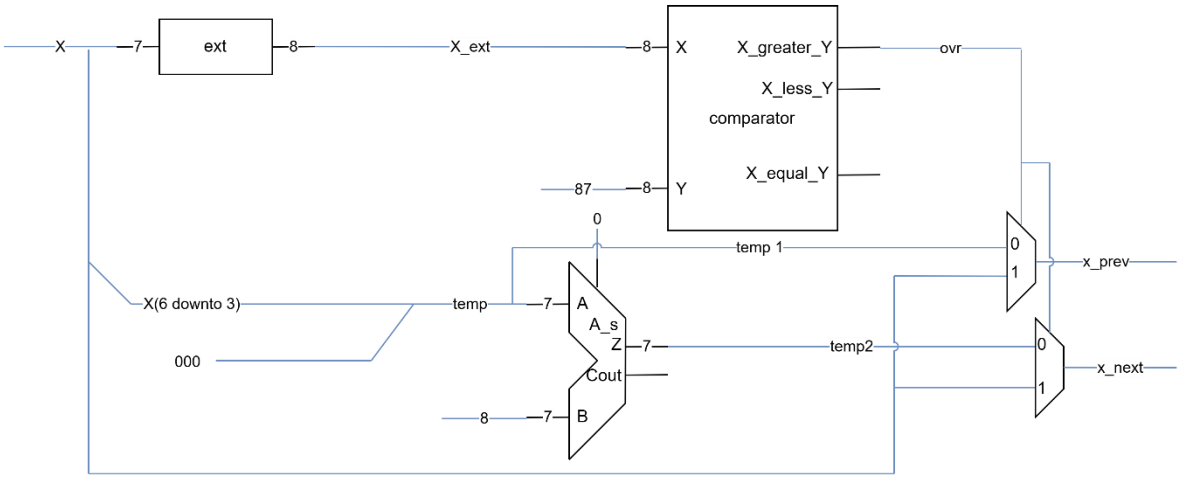
In tutte le altre condizioni il calcolo eseguito per ricavare i multipli adiacenti è il seguente:

$$x_{prev} = \left\lfloor \frac{x}{8} \right\rfloor * 8$$

realizzabile con uno shift logico a destra di 3 bit seguito da uno uguale verso sinistra.

Queste due operazioni possono essere eseguite contemporaneamente semplicemente sostituendo le ultime 3 cifre del numero binario con degli zeri.

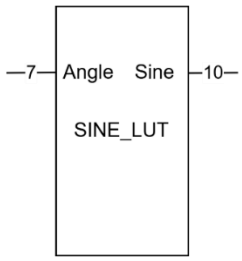
$$x_{next} = x_{prev} + 8$$



Sine LUT

i valori prestabiliti del seno sono codificati in una apposita lookup-table che a fronte di un ingresso binario corretto restituisce in uscita il seno corrispondente su 10 bit in fixed point.

In ogni altro caso (valori non codificati o superiori a 90°) fornisce don't care in uscita.



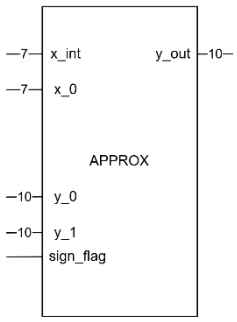
nome	dimensione	direzione
Angle	7 bit	Input
Sine	10 bit	Output

La lookup-table contiene i seguenti valori codificati:

angolo	Valore binario	Valore decimale
0	00.00000000	0
8	00.00100011	0.13671875
16	00.01000110	0.2734375
24	00.01101000	0.40625
32	00.10000111	0.52734375
40	00.10100100	0.640625
48	00.10111110	0.7421875
56	00.11010100	0.828125
64	00.11100110	0.8984375
72	00.11110011	0.94921875
80	00.11111100	0.984375
88	00.11111110	0.9921875
89	00.11111111	0.99609375
90	01.00000000	1
altro	- - - - - - - - - -	d.c.

Linear interpolator

Il componente che effettivamente realizza l'interpolazione è il linear interpolator.



nome	dimensione	direzione
X_int	7 bit	Input
x_0	7 bit	Input
y_0	10 bit	Input
y_1	10 bit	Input
sign_flag	1 bit	Input
y_out	10 bit	Output

Riceve in input l'angolo già traslato e il multiplo di 8 precedente, entrambi interi su 7 bit,

i valori del seno relativi ai multipli precedente e successivo codificati in virgola fissa su 10 bit e il flag di segno.

L'uscita è il valore con segno, su 10 bit in virgola fissa, del seno dell'angolo fornito inizialmente.

calcola la seguente funzione per interpolare i valori del seno:

$$y = y_0 + \frac{(x - x_0)(y_1 - y_0)}{8}$$

il valore viene poi complementato e selezionato se il flag di segno = 1 (negativo).

Il valore $x - x_0$ viene troncato internamente su 3 bit (è sempre < 8) permettendo di utilizzare un moltiplicatore più compatto.

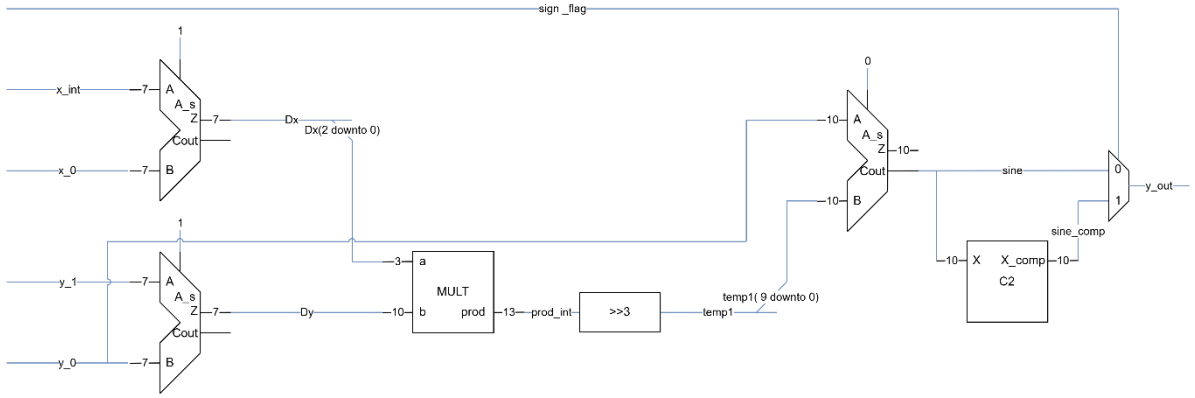
Si noti inoltre che il risultato viene calcolato trattando i numeri come semplici stringhe binarie e che perciò dopo aver effettuato la moltiplicazione e la divisione per 8 il numero di 13 bit viene troncato su 10 bit scartando i 3 bit più significativi (che sono sempre 0).

In caso di ingressi uguali[2] il componente si comporta correttamente, calcolando $(y_1 - y_0) = 0$ e riporta automaticamente in uscita y_0.

[2] Questo caso si verifica per input pari a 88°,89° e 90°, caso in cui y_0 è da considerare corretto perché codificato nella LUT.

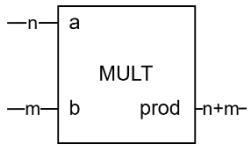
In caso si verificasse a fronte un input troppo grande non presente nella LUT sarà già stato sollevato il flag di errore e perciò qualsiasi dato in uscita sarà senza senso e perciò da ignorare.

Struttura dell’interpolatore:



Multiplier

Per effettuare la moltiplicazione tra due numeri viene utilizzato un moltiplicatore generico che accetta due numeri di n e m bit in binario e ne restituisce il prodotto su n + m bit.

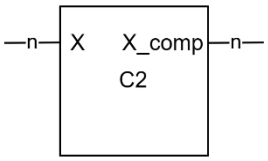


nome	dimensione	direzione
a	n bit	Input
b	m bit	Input
prod	n + m bit	Output

L'architettura interna è la classica a matrice che sfrutta i moduli MAC per calcolare i prodotti di ogni coppia di bit ma è stato implementato per poter moltiplicare numeri binari con lunghezze diverse.

Complement 2

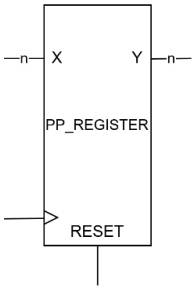
Per complementare l'angolo in caso di segno negativo è stato utilizzato un complementatore generico a n bit che utilizza soltanto porte not e half adder in catena.



nome	dimensione	direzione
X	n bit	Input
X_comp	n bit	Output

PP register

Per memorizzare ingressi e uscite del componente vengono usati dei registri parallelo-parallelo standard a n bit, formati da flip flop D sensibili al fronte di salita e con reset asincrono.

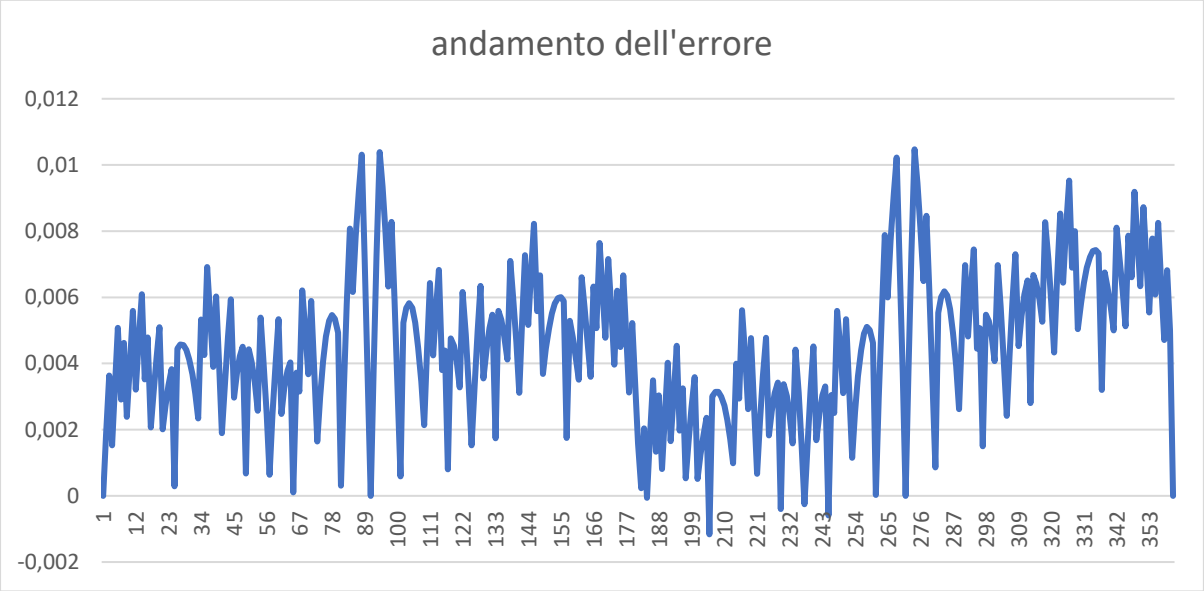


nome	dimensione	direzione
X	n bit	Input
Y	n bit	Output

Verifica

Per replicare il calcolo eseguito è stato realizzato un programma in C che applica lo stesso procedimento del componente , potendo confrontare così i valori attesi teorici e reali.

Il verificatore è stato anche utilizzato per estrapolare i dati relativi all'errore dovuto all'approssimazione eseguita dal componente a ogni angolo richiesto, dati che sono sintetizzati nel seguente grafico:



$errore\ medio$: 0,004546847

$varianza\ campionaria$: 0,0000047618

I picchi di errore in corrispondenza degli angoli associati a 88° (94°, 268° e 274°) sono dovuti all'approssimazione dei dati contenuti nella LUT, che devono essere troncati su 8 bit di parte decimale, dato che per quegli angoli non viene effettuata nessuna interpolazione.

Test-bench

I test-bench realizzato testa il componente su tutti gli angoli anche in ordine casuale.

Viene fornito in input un angolo e al fronte di clock successivo viene generato in output il valore del seno.

L'analisi post-implementation and timing ha permesso di stabilire che la frequenza massima a cui il componente può operare correttamente è di circa 20.833 MHz, che corrispondono ad un periodo di clock di 48 ns.

Testando il componente è emerso che la maggiore limitazione è dovuta agli angoli piccoli (<4°) e agli angoli che devono essere complementati (più ritardo in termini di livelli di logica).

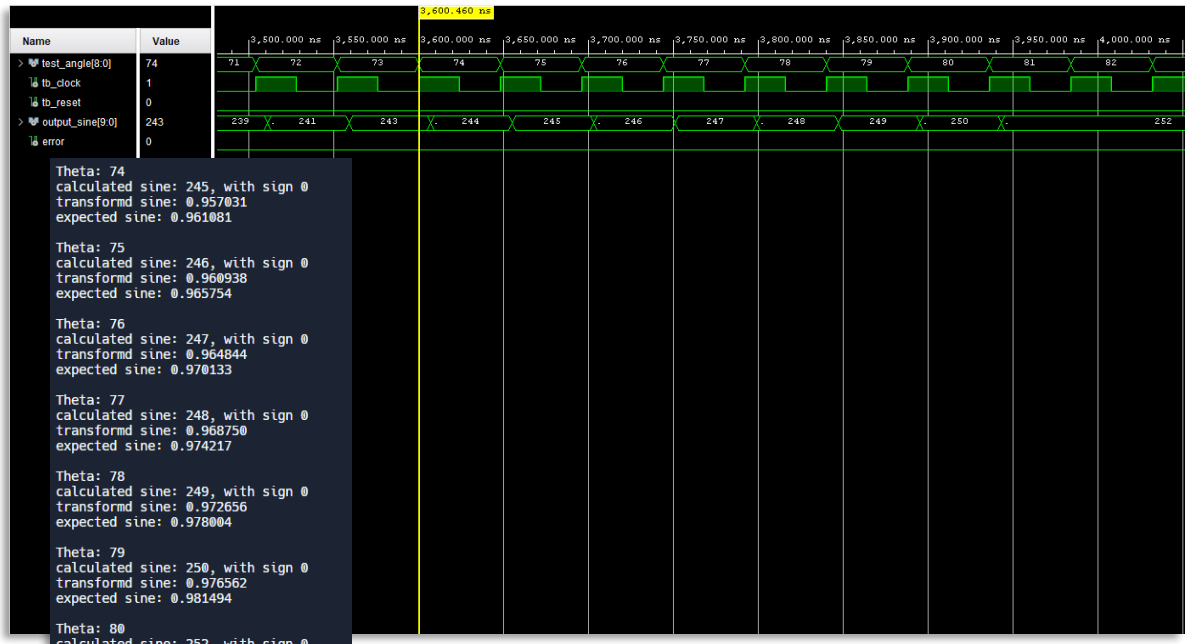
Casi d’uso

Sono stati previsti due casi d’uso per testare il componente:

Nel primo vengono forniti in input tutti gli angoli in ordine, mentre nel secondo, per simulare una applicazione più realistica, vengono inviate richieste di calcolo senza ordine preciso.

In entrambi i casi il componente viene fatto operare alla sua frequenza massima.

confronto tra output del programma verificatore in C e simulazione Vivado:




```
Theta: 74
calculated sine: 245, with sign 0
transformd sine: 0.957031
expected sine: 0.961081

Theta: 75
calculated sine: 246, with sign 0
transformd sine: 0.960938
expected sine: 0.965754

Theta: 76
calculated sine: 247, with sign 0
transformd sine: 0.964844
expected sine: 0.970133

Theta: 77
calculated sine: 248, with sign 0
transformd sine: 0.968750
expected sine: 0.974217

Theta: 78
calculated sine: 249, with sign 0
transformd sine: 0.972656
expected sine: 0.978004

Theta: 79
calculated sine: 250, with sign 0
transformd sine: 0.976562
expected sine: 0.981494

Theta: 80
calculated sine: 252, with sign 0
transformd sine: 0.984375
expected sine: 0.984685

Theta: 81
calculated sine: 252, with sign 0
transformd sine: 0.984375
expected sine: 0.987576

Theta: 82
calculated sine: 252, with sign 0
transformd sine: 0.984375
expected sine: 0.990167
```