Progetto di reti Logiche

Funzione seno in virgola fissa

Daniele Cioffi e Chiara Stilla

Introduzione

Il modulo calcola il seno di un angolo compreso tra 0° e 359°, tramite interpolazione lineare di valori noti presenti in una lookup table.

L'angolo in ingresso (θ) è un intero senza segno espresso su 9 bit. Se θ non rientra nei limiti di specifica si alza un segnale di errore.

Nella lookup table (LUT) sono presenti i seni di tutti gli angoli multipli di 8 compresi tra 0° e 88° e il seno di 89° e 90°. Tutti i seni sono espressi in virgola fissa su 10 bit di cui: 2 di parte intera e 8 di parte decimale.

Per calcolare sin(θ), il modulo trasforma l'angolo in ingresso in un angolo compreso tra 0° e 90° (θ'), e procede all'interpolazione.

La trasformazione dell'angolo avviene tramite identità trigonometriche:

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

L'interpolazione è invece calcolata nel seguente modo:

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

di cui:

- y = valore del seno interpolato
- y_0, y_1 = seno degli angoli multipli di 8 vicini a θ'
- x_0 = angolo multiplo di 8 precedente a θ'

Specifica

La rete è formata da:

- Un registro in ingresso
- Una rete puramente combinatoria che svolge il calcolo del seno dell'angolo in ingresso
- Due registri in uscita, rispettivamente per il risultato e per il segnale di errore

La rete combinatoria è formata da tre macro moduli che svolgono i passaggi del calcolo.

Il primo modulo si occupa di:

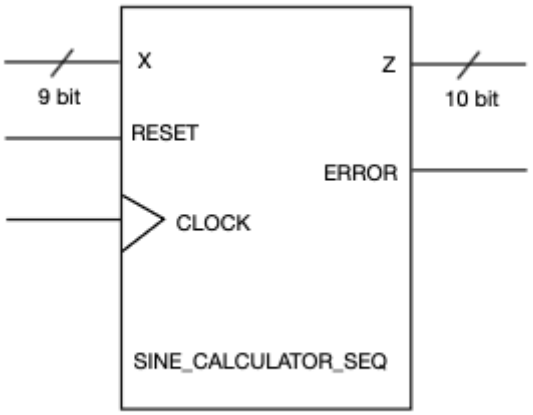
- Trasformare l'angolo in ingresso in un angolo compreso tra 0° e 90° tramite le identità trigonometriche
- Alzare il segnale di segno nel caso in cui il seno da calcolare sia negativo
- Alzare il segnale di errore nel caso in cui l'angolo in ingresso sia > 359°

Il secondo modulo si occupa di calcolare il seno dell'angolo trasformato dal modulo precedente utilizzando la formula dell'interpolazione vista nell'introduzione e consultando le lookup table, ottenendo così un risultato su 10 bit in virgola fissa. Questo risultato però non tiene conto del segno calcolato dal modulo precedente.

Il terzo modulo si occupa infine di rappresentare nel modo corretto, tramite il segno calcolato dal primo modulo, il risultato finale.

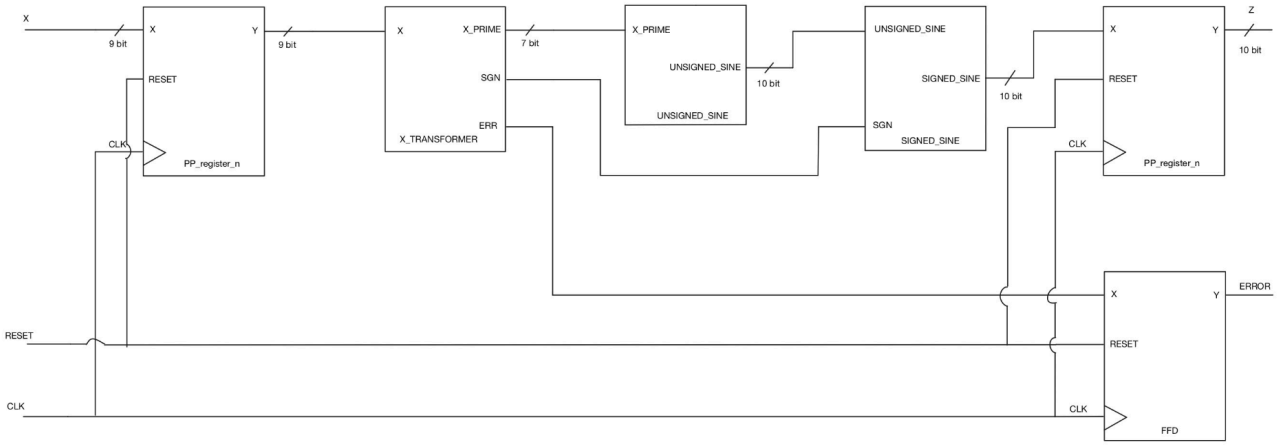
Interfaccia del sistema

L'interfaccia del top-module è la seguente:



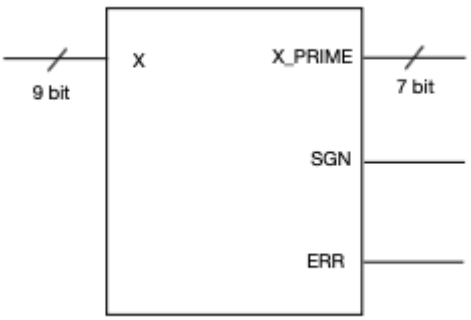
- X: angolo in ingresso espresso su 9 bit (unsigned)
- CLOCK: segnale di clock asincrono (1 bit)
- RESET: segnale di reset asincrono (1 bit)
- Z: seno interpolato di X espresso su 10 bit in virgola fissa (signed) di cui 2 di parte intera e 8 di parte decimale
- ERROR: segnale di errore che vale 1 nel caso in cui l'angolo X sia maggiore di 359° (1 bit)

Architettura del sistema



Modulo 1: X_TRANSFORMER

Interfaccia



- X: angolo in ingresso espresso su 9 bit (unsigned)
- X_PRIME: angolo trasformato compreso tra 0° e 90° espresso su 7 bit (unsigned)
- SGN: bit di segno del seno
- ERR: bit di errore

Architettura

Il primo livello di logica è formato da due gruppi di CLA:

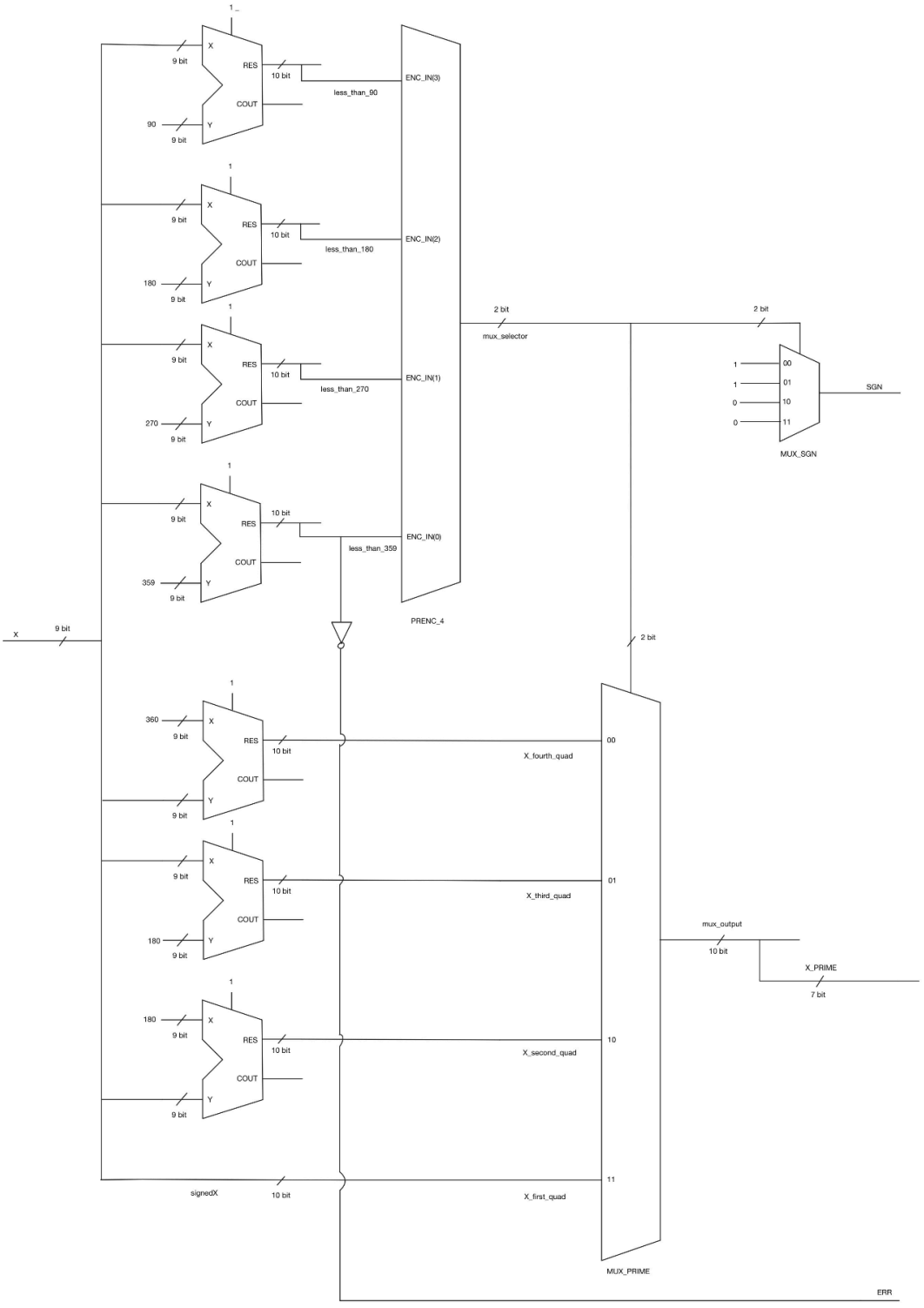
- Il primo gruppo serve per capire il quadrante di appartenenza dell'angolo X. Questi CLA sottraggono rispettivamente 90°, 180°, 270°, 359° all'angolo in ingresso.
- Il secondo gruppo applica le identità trigonometriche per calcolare X_PRIME per tutti i casi possibili di X

Il secondo livello di logica è formato da:

- Priority Encoder: prende in ingresso i MSB del primo gruppo di CLA e genera il segnale di selezione per i due MUX.
- Porta not: nega il MSB calcolato dal CLA che svolge X-359°; il risultato è il flag di errore

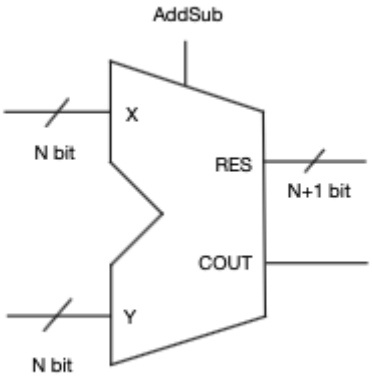
Il terzo livello di logica è formato da 2 MUX:

- MUX_PRIME: in base al segnale di selezione calcolato dal priority encoder, sceglie il corretto X_PRIME da portare sull'uscita.
  Segnale di selezione:
  - 00 => X appartiene al quarto quadrante
  - 01 => X appartiene al terzo quadrante
  - 10 => X appartiene al secondo quadrante
  - 11 => X appartiene al primo quadrante
- MUX_SGN: sceglie il corretto bit di segno con lo stesso criterio di selezione di MUX_PRIME


1
X
9 bit
RES
10 bit
COUT
90
Y
9 bit
less_than_90
ENC_IN(3)
1
X
9 bit
RES
10 bit
COUT
180
Y
9 bit
less_than_180
ENC_IN(2)
1
X
9 bit
RES
10 bit
COUT
270
Y
9 bit
less_than_270
ENC_IN(1)
1
X
9 bit
RES
10 bit
COUT
359
Y
9 bit
less_than_359
ENC_IN(0)
PRENC_4
2 bit
mux_selector
2 bit
1
1
0
0
00
01
10
11
SGN
MUX_SGN
X
9 bit
2 bit
1
360
X
9 bit
RES
10 bit
COUT
Y
9 bit
X_fourth_quad
00
1
X
9 bit
RES
10 bit
COUT
180
Y
9 bit
X_third_quad
01
1
180
X
9 bit
RES
10 bit
COUT
Y
9 bit
X_second_quad
10
1
X
9 bit
RES
10 bit
COUT
Y
9 bit
signedX
10 bit
X_first_quad
11
mux_output
10 bit
X_PRIME
7 bit
MUX_PRIME
ERR

CLA_AddSub

Interfaccia



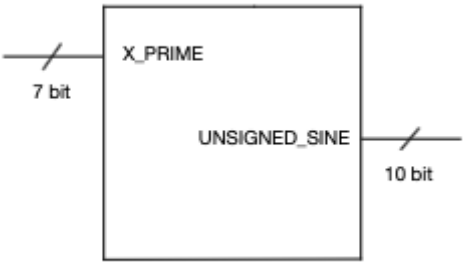
- X: primo operando unsigned su N bit
- Y: secondo operando unsigned su N bit
- AddSub: operazione da svolgere su un bit(0 per la somma e 1 per la sottrazione)
- RES: risultato signed su N+1 bit
- COUT: bit di carry

Il CLA_AddSub prima di svolgere l'operazione aggiunge il bit di segno '0' a X e Y.

La scelta di usare dei sommatori/sottrattori basati su CLA è legata al calcolo del ritardo: dovendo eseguire delle operazioni su operandi con più di 9 bit, il CLA_AddSub garantisce un ritardo T=7 (5 dovuti al CLA + 2 dovuti allo XOR bit a bit di Y con AddSub); altri sommatori (RCA e CSA) sarebbero stati più lenti.

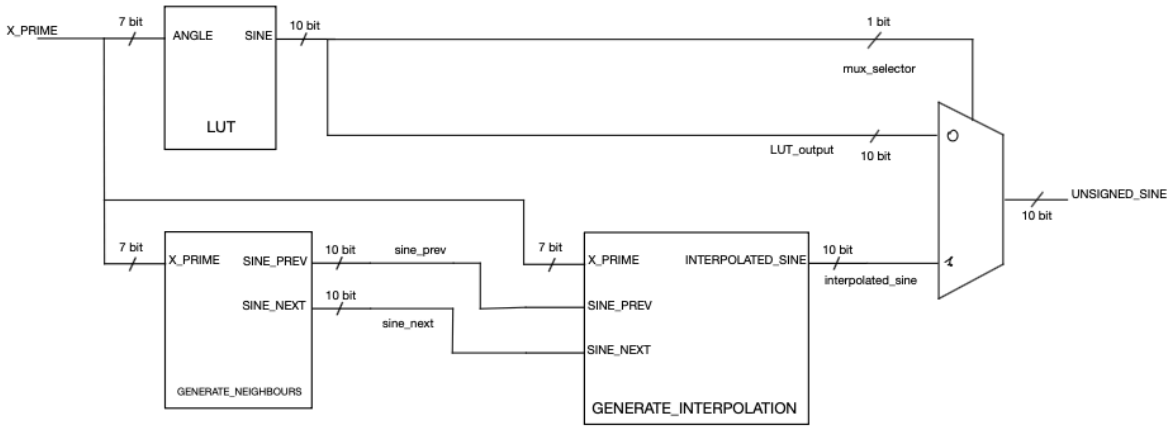
Modulo 2: UNSIGNED_SINE

Interfaccia



- X_PRIME: angolo trasformato (7 bit)
- UNSIGNED_SINE: seno dell'angolo trasformato (sicuramente positivo su 10 bit)

Architettura



L'angolo trasformato entra in due sottoreti.

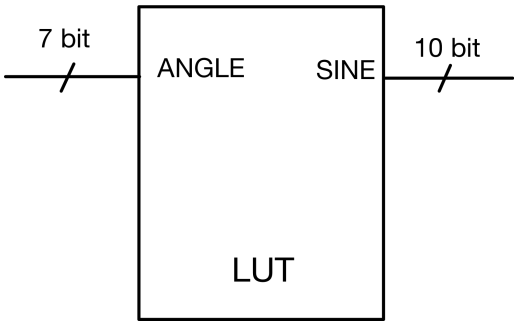
La prima (LUT) restituisce un valore della lookup-table: il seno dell'angolo in ingresso se quest'ultimo è multiplo di 8 e compreso tra 0° e 88° oppure se vale 89° 90°; altrimenti ritorna "1000000000".

La seconda rete effettua l'interpolazione del seno dell'angolo: trova prima il seno degli angoli multipli di 8 più vicini (GENERATE_NEIGHBOURS) e successivamente ne calcola l'interpolazione (GENERATE_INTERPOLATION).

I valori di uscita di queste due sottoreti entrano in un multiplexer il cui segnale di selezione è il MSB dell'output della LUT: sapendo che il seno degli angoli presenti nella LUT è un valore sicuramente positivo (MSB = 0) e il valore di default ha MSB = 1, basta questo singolo bit per riportare sull'uscita l'ingresso giusto.

LUT

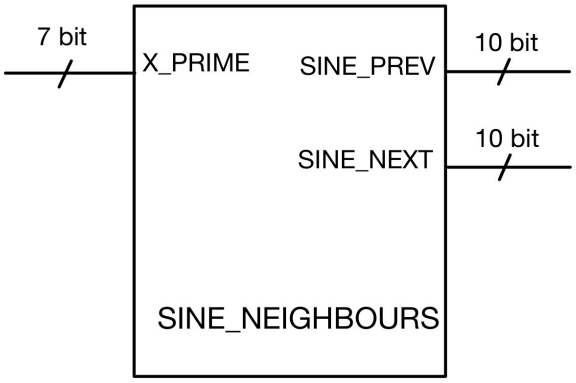
Interfaccia



Angolo (ANGLE)	Valore binario in CA2 (SINE)	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	10.00000000	-2

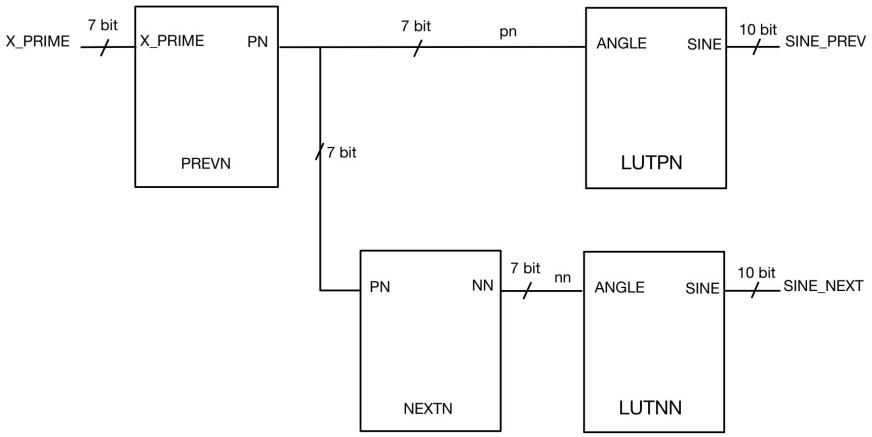
SINE_NEIGHBOURS

Interfaccia



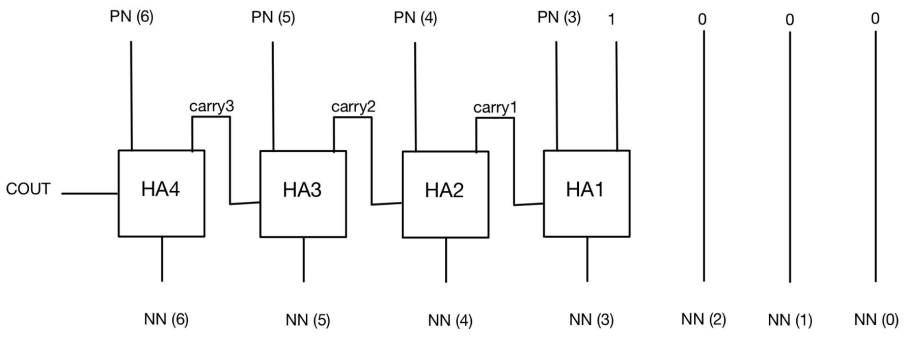
- X_PRIME: angolo trasformato (7 bit)
- SINE_PREV: seno dell'angolo multiplo di 8 precedente a X_PRIME (10 bit)
- SINE_NEXT: seno dell'angolo multiplo di 8 successivo a X_PRIME(10 bit)

Architettura



Il blocco PREVN calcola il valore dell'angolo multiplo di 8 precedente a X_PRIME semplicemente ponendo gli ultimi 3 LSB a 0.

Il blocco NEXTN somma 8 a pn ottenendo così l'angolo multiplo di 8 successivo a X_PRIME. La somma avviene tramite la seguente rete:

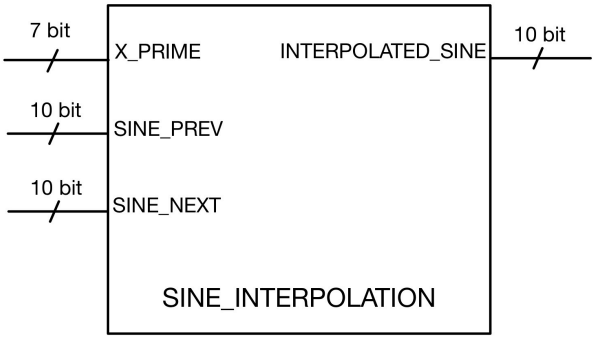


Da un punto di vista di area e ritardo questo modulo sommatore è migliore rispetto al CLA_AddSub.

Le due istanze di LUT restituiscono il seno dei due angoli vicini a X_PRIME (pn e nn)

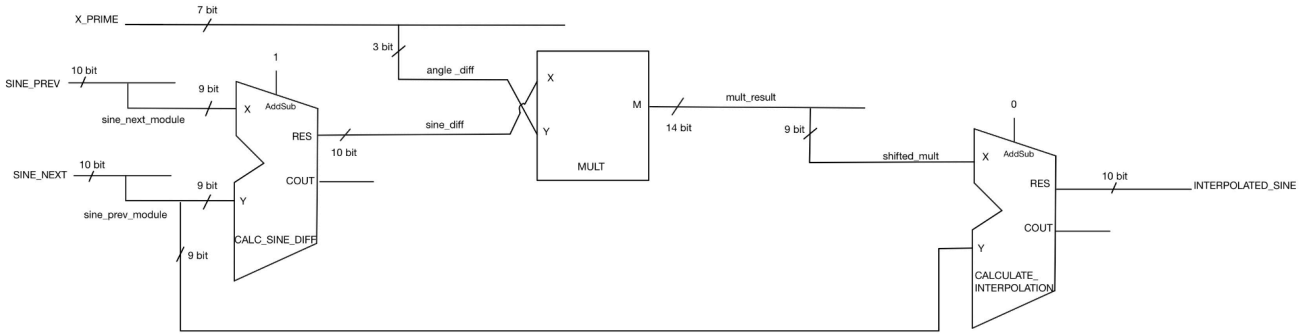
SINE_INTERPOLATION

Interfaccia



- X_PRIME: angolo trasformato (7 bit)
- SINE_PREV: seno dell'angolo multiplo di 8 precedente a X_PRIME (10 bit)
- SINE_NEXT: seno dell'angolo multiplo di 8 successivo a X_PRIME (10 bit)
- INTERPOLATED_SINE: risultato dell'interpolazione (10 bit)

Architettura



Premessa: poichè il CLA_AddSub prende in ingresso valori UNSIGNED viene rimosso il MSB da SINE_PREV e SINE_NEXT.

Questo modulo svolge la formula di interpolazione:

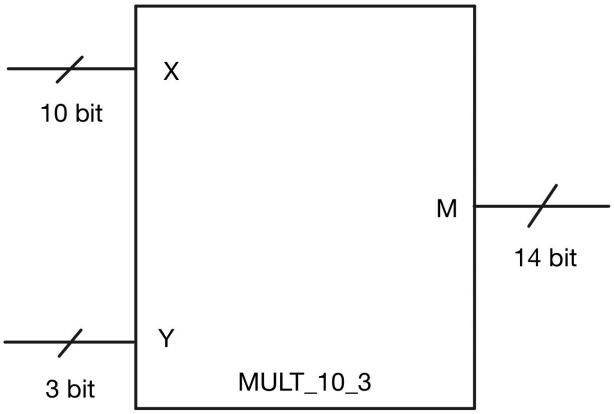
$$INTERPOLATED\ SINE\ =\ SINE\ PREV\ +\ \frac{(X\ PRIME-PN)(SINE\ NEXT-SINE\ PREV)}{8}$$

- SINE_NEXT - SINE_PREV: calcolato dal CLA_AddSubb CALC_SINE_DIFF
- X_PRIME - PN: equivale a prendere gli ultimi 3 bit di X_PRIME, infatti il massimo valore di questa differenza (7) è esprimibile su 3 bit e PN, essendo multiplo di 8, ha gli ultimi 3 bit uguali a 0
- (X_PRIME - PN)*(SINE_NEXT - SINE_PREV): calcolata dal MULT_10_3 (MULT)

- La divisione per 8 è ottenuta shiftando il risultato del MULT di 3 bit a destra
- La somma finale è eseguita dall'ultimo CLA_AddSub (CALCULATE_INTERPOLATION)

MULT_10_3

Interfaccia

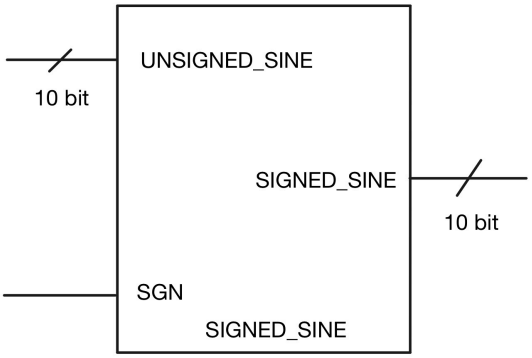


- X: primo operando (10 bit)
- Y: secondo operando (3 bit)
- M: risultato della moltiplicazione (14 bit)

Il moltiplicatore calcola prima i tre prodotti parziali (da 12 bit) e poi li somma tramite un sommatore CarrySave.

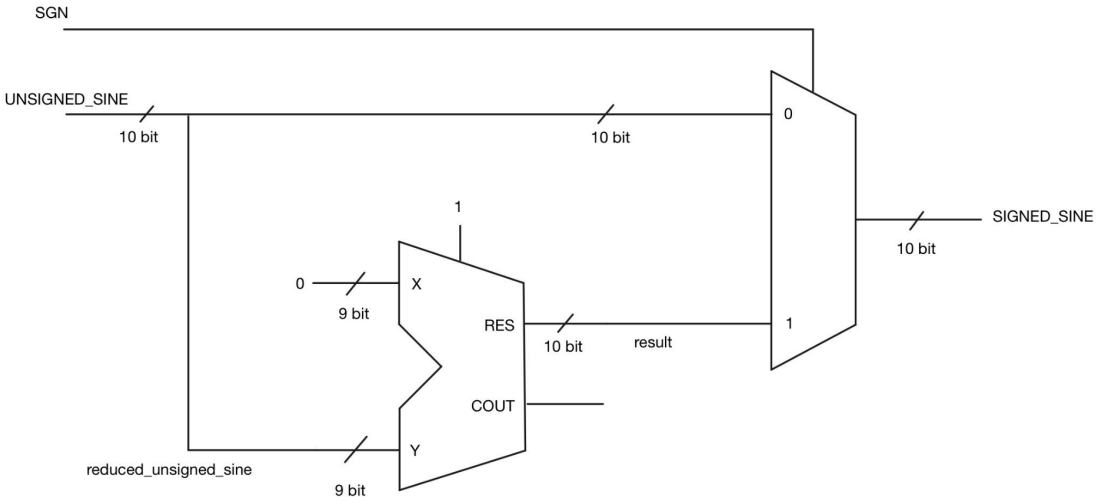
Modulo 3: SIGNED_SINE

Interfaccia



- UNSIGNED_SINE: seno calcolato dal modulo UNSIGNED_SINE (10 bit)
- SGN: bit di segno calcolato dal modulo X_TRANSFORMER
- SIGNED_SINE: seno finale in complemento a 2

Architettura



In base al bit di segno scelgo se negare o meno UNSIGNED_SINE.
La negazione è svolta dal CLA_AddSub.

Verifica

Per ogni componente utilizzato sono stati eseguiti due tipi di test:

- Behavioural: per verificare il corretto comportamento del componente dal punto di vista funzionale
- Post Place&Route (PPR): per verificare il corretto comportamento del componente tenendo conto dei ritardi dovuti agli elementi combinatori del sistema

Test-bench e Casi d'uso

Il test-bench realizzato per il componente top-level testa tutti i possibili angoli esprimibili su 9 bit, ovvero tutti gli angoli tra 0° e 511° permettendoci così di testare anche i casi limite (ovvero gli angoli multipli di 8° e gli angoli 89° e 90°).
I risultati della simulazione behavioural sono stati verificati tramite un programma C.

L'analisi post-implementation timing ha permesso di stabilire che la frequenza a cui il componente può operare correttamente è di circa 200MHz, che corrispondono ad un periodo di clock di 5 ns.

Durante la simulazione PPR viene alzato un segnale di reset per 150ns e successivamente viene fornito in ingresso alla rete un valore ogni ciclo di clock. Il risultato è disponibile dopo poco più di un ciclo di clock.