La Calcolatrice Atahualpa

Premessa

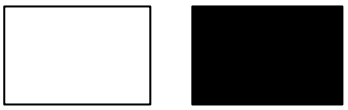
Il piccolo abaco inca a base 40, decriptato dal Prof. Nicolino De Pasquale, permette di effettuare calcoli con una precisione sorprendente anche per i giorni nostri. Partendo da ovvie e dovute cautele che rendono il paragone improponibile, possiamo fare alcune considerazioni sulla precisione di calcolo confrontando semplicemente l'abaco con la struttura base dei moderni microprocessori. Considerando l'esemplare andino a 15 righe, utilizzato per calcoli astronomici e raffigurato in alcuni vasi (De Pasquale), possiamo notare che l'approssimazione reale vale 40^{-15} pari a circa 10^{-24}, ossia l'unità in confronto ad un impronunciabile numero di 25 cifre decimali (1.073.741.824.000.000.000.000.000, poco più di un milione di miliardi di miliardi). Immaginando questo numero come la capacità strutturale di un registro in una architettura binaria di calcolo, si ottiene l'equivalenza nella dimensione di 80 bit. Il diffuso attuale processore Pentium (Intel) ha un'architettura di 32 bit e dobbiamo aspettare la successiva generazione Itanium (Intel, previsto per il 2004) che sarà di "soli" 64 bit (2^{-64} pari a $5{,}4 \cdot 10^{-20}$), ma con precisione circa 58.000 volte inferiore rispetto all'abaco più evoluto degli Incas. La considerazione, si torna a ribadire, non è assoluta perché ovviamente non è confrontabile un singolo registro con l'intera architettura di un processore ed inoltre è noto che le nostre strutture di calcolo prevedono precisioni estese ottenibili con l'unione di più registri base. Resta però il fatto che un simile abaco permetteva agli Incas, come storicamente documentato, di trattare numeri enormi con una precisione tale da prevedere con esattezza gli eclissi. Straordinarie anche le necessarie conoscenze astronomiche in un epoca in cui la nostra civiltà stava ancora confrontandosi sulla visione eliocentrica del sistema solare. Con tali premesse é particolarmente interessante esplorare un sistema di calcolo così profondamente diverso da tutti gli altri sinora conosciuti e il primo gradino della ricerca porta alla costruzione di una elementare calcolatrice basata esclusivamente su algoritmi inca, quindi totalmente indipendente dai sistemi decimale o binario. Il nome "Atahualpa" deriva dall'ultimo re che ha regnato sul Popolo Inca.

Figura 1

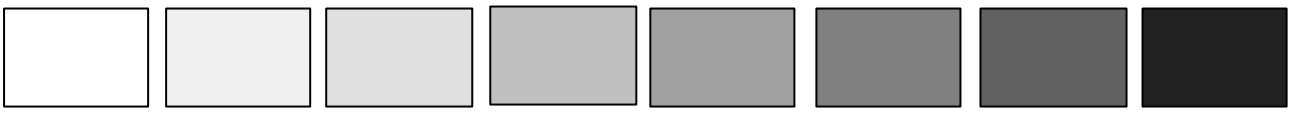
La base 40

Il codice inca è a base 40 e quindi si possono avere 40 combinazioni differenti. Si potrebbe pensare alla costruzione di una macchina multilivello, ma questa scelta si rileva non conveniente. I nostri attuali automi più diffusi sono basati sul sistema binario che da un lato, essendo il più semplice, ha minor potere di numerazione, ma dall'altro permette la costruzione di apparati caratterizzati da bassa probabilità di errore, basso costo e grande velocità. Infatti la base binaria ha solo 2 livelli (zero ed uno) e nel loro riconoscimento una macchina si trova nelle condizioni ottimali dovendo distinguere solo tra assenza e presenza di segnale per assegnare un valore ad ogni bit. Facendo un paragone ottico è come se si dovessero semplicemente distinguere fogli bianchi da fogli neri per determinare se ci si trovi di fronte ad uno zero oppure ad un uno. Passando invece ad un sistema di

numerazione con più livelli occorre aggiungere tra il bianco e il nero più sfumature di grigio e il riconoscimento automatico diviene subito più critico, occorre utilizzare dispositivi più costosi perché, per valutare la differenza tra un livello di grigio e l'altro, diventa elevata la probabilità di commettere errori tanto più numerosi sono i livelli. La figura 2 rende l'idea con un esempio di codifica a 8 livelli.



Codifica binaria (2 livelli)



Esempio di codifica in 8 livelli

Figura 2

Il paragone è ottico, ma l'analogia con il caso elettrico non cambia le conclusioni. Il riconoscimento tra 2 soli livelli minimizza costo e probabilità di errore; non è un caso infatti che il sistema minimo a 2 livelli si sia affermato con così grande e capillare diffusione. Per quanto detto, considerando ovviamente le attuali tecnologie disponibili, sicuramente la costruzione di un sistema di codifica come quello inca con 40 livelli non è proponibile perché avrebbe un costo abnorme (dovendo anche essere costruito interamente dal principio) ed una immunità ai disturbi critica, quindi elevata probabilità di errore. Comunque per la costruzione della calcolatrice possono ugualmente essere adottati componenti binari per l'acquisizione dei dati in ingresso (tastiera) e la visualizzazione in uscita (display), nonché per la memorizzazione e l'elaborazione dei dati. Basta per questo interpretare ogni seme dell'abaco come un segnale binario che ne attesti l'assenza o la presenza. Questa scelta semplifica notevolmente il lavoro potendo utilizzare comuni apparati reperibili in commercio senza però cambiare la natura del progetto perché nelle elaborazioni non sarà utilizzata né l'aritmetica binaria né quella decimale, ma, come detto, saranno individuati e adottati specifici algoritmi basati sull'aritmetica inca. Occorre anche sottolineare che gli Incas adottavano la simbologia dell'abaco anche per la memorizzazione dei numeri attraverso il sistema delle cordicelle annodate (quipus, nella miniatura di figura 1 è raffigurato tra le mani del Curaca) non possedendo una associazione simbolica con le 40 combinazioni della base, ossia è direttamente l'abaco a rappresentare i numeri mediante assenza o presenza di semi per cui la scelta effettuata rispetta pienamente il sistema andino. Anzi, una certa parentela strutturale con il sistema binario si può evidenziare cercando scorciatoie pratiche semplificate di conversione manuale da codice inca a decimale. Infatti il peso di ogni riga inca ennesima è pari alla base elevato alla N e 40^N non è una potenza facilmente determinabile a mente, ma se scriviamo

$$40^N = 2^{2N} \cdot 10^N$$

possiamo calcolare i pesi con notevole semplicità (ad esempio il peso della 5 riga equivale a $2^{10} \cdot 10^5$ ossia 1.024 seguito da 5 zeri). Questa considerazione mostra che la base 40 è riconducibile sia alla base 2 e sia alla base 10; da questa proprietà potrebbero derivare particolari algoritmi ottimizzati

Variabili di ingresso e di uscita

Nello schema a blocchi generale della calcolatrice devono essere specificate le variabili di ingresso e le variabili d'uscita. Risultano in ingresso tutti i segnali provenienti dalla tastiera e in uscita tutte le segnalazioni realizzabili con matrici a cristalli liquidi o a diodi luminosi di tipo Led. Occorre ora scegliere il numero di righe da inserire nel display e, potendo utilizzare apparati hardware binari, conviene scegliere 8 in quanto potenza di 2. Si hanno in totale 40^8 combinazioni pari a 6,55 migliaia di miliardi, più che sufficienti per gli scopi del prototipo. L'assegnazione della riga con peso 40^0 è comunque arbitraria e può essere scelta a seconda delle necessità per la massima flessibilità della calcolatrice. In questo modo si possono ottenere anche numeri minori di 1, fermo restando le stesse combinazioni totali. Per l'immissione dati non conviene adottare un numero di tasti pari al numero di punti del display. Infatti può essere adottata una riga di immissione unica che, tramite due tasti di scorrimento (in alto e in basso), permetta di raggiungere tutte le righe per definire lo stato di ogni seme. Questa scelta semplifica di molto il dispositivo di ingresso riducendo a 11 i tasti numerici (invece di 88) e in aggiunta richiede solo in uscita una serie di 8 Led a fianco di ogni riga per segnalare quale sia quella attiva per l'immissione dati.
Si hanno allora 11 tasti numerici, 9 tasti funzione per gli ingressi, 88 led per la visualizzazione delle 8 righe, 8 led per la segnalazione della riga attiva e 1 led per la segnalazione della condizione di overflow ed 1 per l'indicazione del segno. In totale si contano 20 ingressi e 98 uscite che però possono essere notevolmente ridotti adottando la tecnica del multiplexer gestendo le informazioni in forma matriciale.

La figura 3 mostra la schematizzazione elementare del sistema.

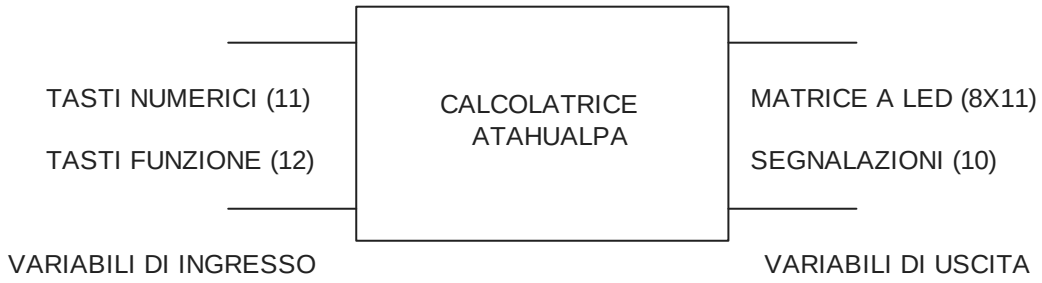


Figura 3

Per meglio seguire i calcoli conviene utilizzare una variante della modalità operativa denominata "notazione inversa polacca" che prevede di effettuare prima le immissioni e poi definire l'operatore. Lo scopo è quello di visualizzare operandi e risultato anche dopo l'esecuzione delle operazioni. Oltre ai tasti delle 4 operazioni fondamentali e alla cancellazione sarà quindi presente un tasto di scambio tra gli operandi.

In conclusione si ottiene il layout dei dispositivi di ingresso e di uscita della calcolatrice rappresentato nelle figure seguenti.

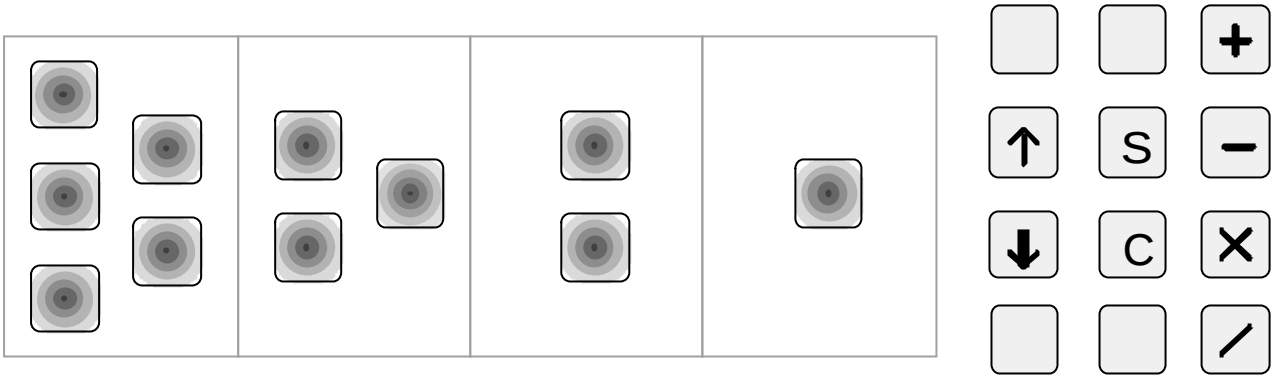


Figura 4

Tastiera.

La figura rappresenta i dispositivi di ingresso: tasti numerici e tasti funzione. Per semplicità i tasti numerici si riferiscono ad una sola riga, ma mediante 2 tasti di direzione si passa da una riga all'altra per definire ogni seme dell'abaco. Una opportuna segnalazione luminosa sul pannello di uscita indica a quale riga si riferisce la digitazione bistabile (premendo un tasto si accende il led corrispondente, premendolo di nuovo lo stesso si spegne).

Display

Il disegno sulla destra rappresenta le segnalazioni d'uscita con 8 righe rappresentate in totale mediante diodi led disposti esattamente come previsto dall'abaco inca. Al fianco sinistro di ogni riga un Led dedicato (più chiaro) segnala a quale riga si riferiscono i tasti di immissione. Questa scelta consente di limitare ad una sola riga la tastiera di ingresso riducendo le dimensioni della calcolatrice, ma non la praticità di utilizzo. Infine un led nella parte superiore del display indica la situazione di errore nel caso di un risultato più grande della capacità massima della matrice di uscita. Il display a 8 righe, partendo dalla potenza zero, permette di ottenere in totale 40^8 combinazioni pari a 6,55 migliaia di miliardi, mentre con le 5 righe rappresentate nella miniatura del Curaca, si ottengono 40^5 combinazioni pari a 102,4 milioni. Ma, come detto, la riga con potenza 40^0 è rilocabile a seconda delle esigenze in modo dinamico e quindi il massimo numero rappresentabile dipende da questa scelta.

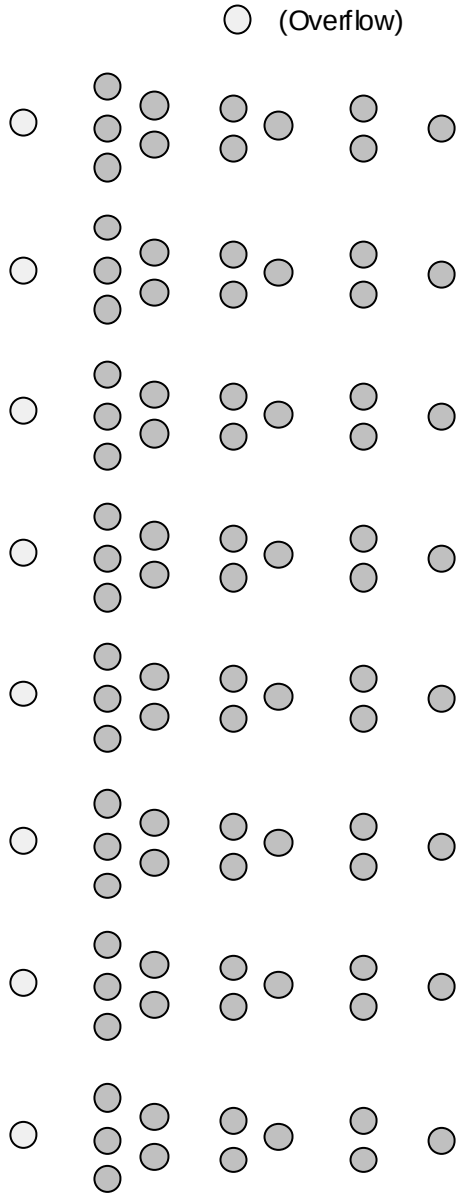


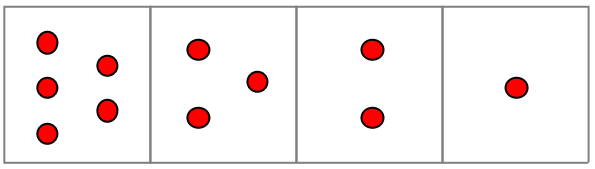
Figura 5

Scelta della tecnologia

La scelta primaria da effettuare per la costruzione di un sistema sta nel tipo di tecnologia da utilizzare. In elettronica le possibili alternative sono davvero molteplici e anche profondamente diverse tra loro. In questo caso si poteva procedere con la sintesi di una unità aritmetica dedicata, con la sintesi di una rete sequenziale, con i gate array, con una rete neurale, oppure ancora con una macchina programmabile a microprocessore o microcontrollore. Le prime vie sono state scartate perché le realizzazioni hardware non sono in genere flessibili e poco si prestano alla sperimentazione per l'ottimizzazione degli algoritmi pur avvalendosi della simulazione. Le reti neurali, essendo in grado di apprendere e mettere quindi a frutto gli insegnamenti acquisiti, sono più adatte alla soluzione di problemi nei quali non si riesce a prevedere tutti i casi che possono verificarsi, ma l'aritmetica inca è basata su regole non imprevedibili. Dato anche il grande numero di variabili in gioco si è scelta la strada a microprocessore per poter contare sulla maggiore espandibilità e flessibilità del sistema in sede prototipale. Una volta stabilita la configurazione ottimale il tutto può essere trasferito su un sistema basato su un adeguato microcontrollore in modo da minimizzare l'hardware.

Lo studio preliminare del codice inca

E' estremamente interessante esplorare un mondo numerico tanto particolare cercando di ricreare e sintetizzare le procedure elementari e gli algoritmi che dovevano costituire la base del calcolo numerico. Il sistema di numerazione delle arcaiche popolazioni andine a base 40 rappresenta in una riga, attraverso una serie di semi di mais, 40 differenti valori compreso lo zero. In realtà le combinazioni totali sono maggiori, ma si considerano nel codice solo i valori effettivi senza tener conto delle ridondanze (più combinazioni differenti portano allo stesso valore). Ci troviamo anche di fronte all'uso di sottobasi nell'abaco. Ogni unità può essere presa al massimo una volta, ogni coppia al massimo 2, analogamente ogni terna 3 e ogni cinquina 5. In figura è riportato il massimo codice ottenibile pari a 39. Si può osservare che i 40 simboli della base sono ottenuti utilizzando 3 "sotto-basi". L'unità e le coppie operano in base 2 essendo i pesi pari a 2^0 e 2^1, le terne in base 3 con peso 3^1 e le cinquine in base 5 con peso 5^1. Come si vede si tratta di un codice misto nel quale il concetto di posizione non segue i canoni standard. Inoltre ciascuna sotto cifra può essere presa al massimo per un numero pari alla base ad eccezione della prima. Infatti in senso convenzionale un sistema che opera in base B possiede B simboli (da un minimo di 0 a un massimo di B-1), mentre il sistema Inca ne prevede uno in più potendo rappresentare dalla mancanza di semi fino al massimo possibile pari a 5 (con riferimento alle cinquine), per un totale di 6 combinazioni. La scrittura di un numero N può essere schematizzata nel modo seguente:



Rappresentazione del numero 39

Figura 6

$$N = Ci \cdot 5 + Te \cdot 3 + Co \cdot 2^1 + Un \cdot 2^0$$

Dove con Ci si sono indicate le cinquine, con Te le terne, con Co le coppie e con Un le unità Volendo forzare la definizione di posizione si potrebbe scrivere:

$$N = Ci \cdot \sqrt[3]{5}^{3} + Te \cdot \sqrt[2]{3}^{2} + Co \cdot 2^1 + Un \cdot 2^0$$

Il sistema Inca è quindi un sistema che utilizza sottosistemi a base multipla e risulta molto lontano dalle intuitive numerazioni basate ad esempio sul numero di dita della specie uomo.
Addentrandosi nella struttura di un tale universo matematico è di fondamentale importanza la sintesi dell'elemento primario della attività numerica: il sommatore in grado di effettuare la somma tra due cifre con riporto. Tale circuito è noto nella aritmetica binaria con il termine inglese full adder e, per quanto detto in precedenza, c'è da aspettarsi per il mondo Inca un addizionatore profondamente differente dal caso binario. In effetti la presenza contemporanea di 3 basi distinte per la formazione della base globale fa intuire che tale circuito non sia né semplice né "lineare" come accade invece per l'equivalente binario.
Infatti per quest'ultimo il sommatore elementare (half adder), capace di sommare tra loro due bit generando in uscita un bit di risultato più un bit di riporto, permette di ottenere il sommatore completo (full adder) considerando in ingresso anche il riporto come appare nella figura seguente:

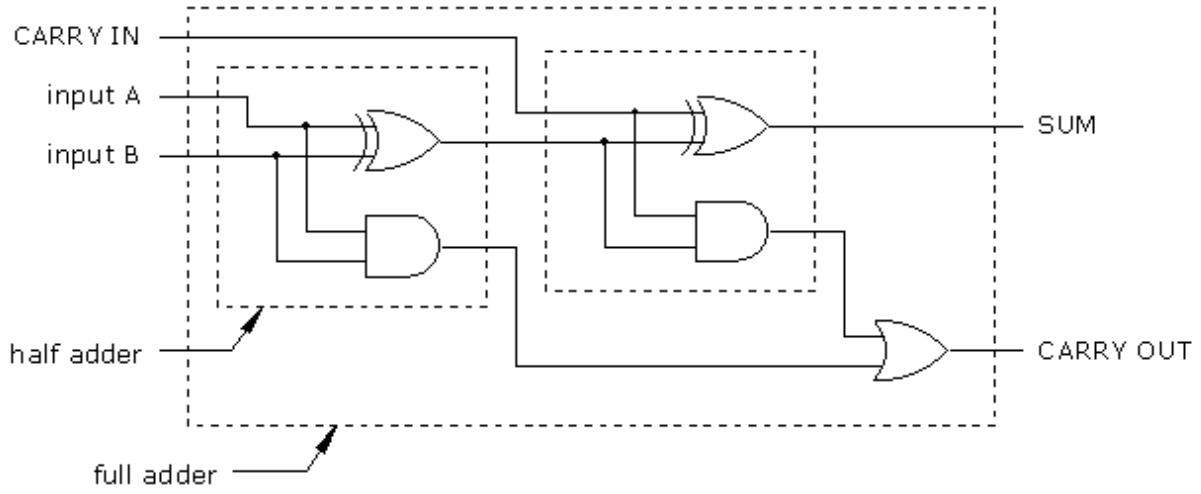


Figura 7

Tutta la struttura del full adder è basata sull'half adder. Data la semplicità del codice binario, più circuiti full adder possono essere disposti semplicemente in cascata per ottenere l'addizionatore binario di due parole di N bit come riportato nella figura seguente riferita al caso di 4 bit:
Nel caso del sistema di numerazione Inca invece l'addizionatore è composto da addizionatori parziali tra loro differenti poiché operano nelle 3 differenti basi sopra citate. Il circuito in cascata non è direttamente applicabile come nel caso binario per il suddetto motivo ed inoltre perché i riporti si propagano sia verso destra che verso sinistra.
A prima vista l'approccio sembrerebbe alquanto complesso così come l'utilizzo pratico dell'abaco a semi di mais, ma se un popolo tanto evoluto ha utilizzato questo sistema di numerazione riuscendo a fare calcoli notevolmente complessi come ad esempio la previsione degli eclissi, si può dedurre che doveva anche possedere regole pratiche ed elementari per utilizzare l'abaco ed eseguire i calcoli. Ma le regole nascono dalle particolarità e, nella strada che conduce alla sintesi del sommatore, conviene partire dallo studio del sistema stesso alla ricerca delle più semplici proprietà algebriche che possono costituire la chiave nella determinazione dell'algoritmo primario. Partiamo allora dalla osservazione dei pesi di ciascuna casella nell'abaco Inca: **5, 3, 2, 1.**

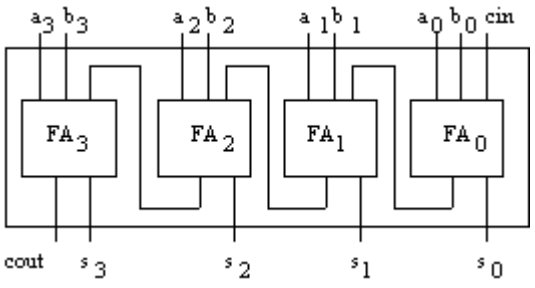


Figura 8

Dati due numeri ovviamente sono sommabili gli elementi che hanno lo stesso peso e si può applicare nell'ambito di ogni casella la semplice somma binaria interpretando come 0 l'assenza di un seme e come 1 la presenza, ma il vero ostacolo della somma sta nella generazione dei riporti che possono andare sia verso destra che verso sinistra con pesi differenti e con la necessità di dover fare diversi passi successivi per riportare il risultato alla forma canonica (vedi miniatura Curaca).

Per formalizzare e semplificare il problema allora conviene studiare le proprietà dei pesi sopra riportati. Risulta subito evidente che la somma un elemento del primo e del secondo peso a partire da destra dà un elemento del terzo (3=2+1) così come la somma di un elemento del secondo e del terzo dà un elemento del quarto (5=3+2). Tale equivalenza è mostrata sull'abaco:

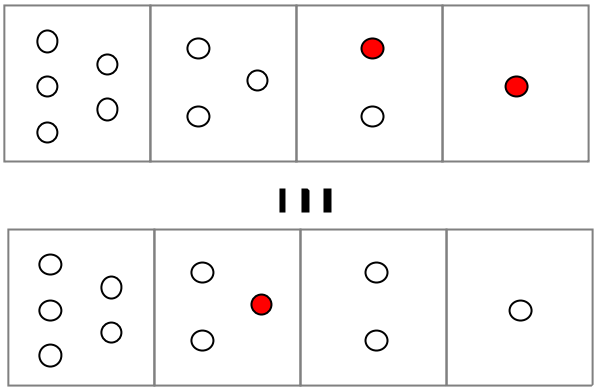


Figura 9

Il caso della cinquina equivalente alla somma della terna e della coppia è perfettamente analogo. Il circuito digitale che realizza tale funzione riconosce la condizione in cui i due ingressi sono entrambi ad 1 generando riporto successivo (C) e annullando contemporaneamente i primi ingressi. Uno schema elettrico del generico traslatore verso sinistra è il seguente:

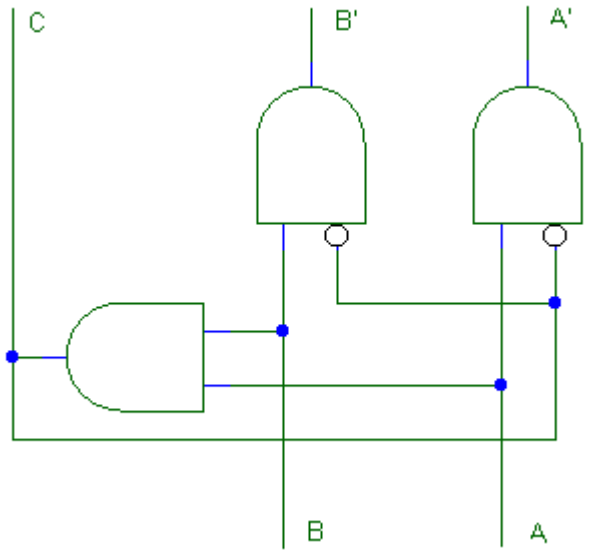


Figura 10

La tabella di verità per l'uscita C (riporto) risulta essere quella di una tipica porta AND:

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

La tabella di verità per l'uscita A' (e ugualmente B') risulta essere quella di una tipica porta AND con un ingresso negato, in modo da annullare l'uscita nel caso l'equivalenza in oggetto sia stata applicata e lasciando i semi inalterati in caso contrario:

C	Not C	A	A'
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

Proseguendo nella sintesi, conviene suddividere il circuito in sottocircuiti per limitare il numero di ingressi in ciascuna rete. Una possibile organizzazione interna è la seguente:

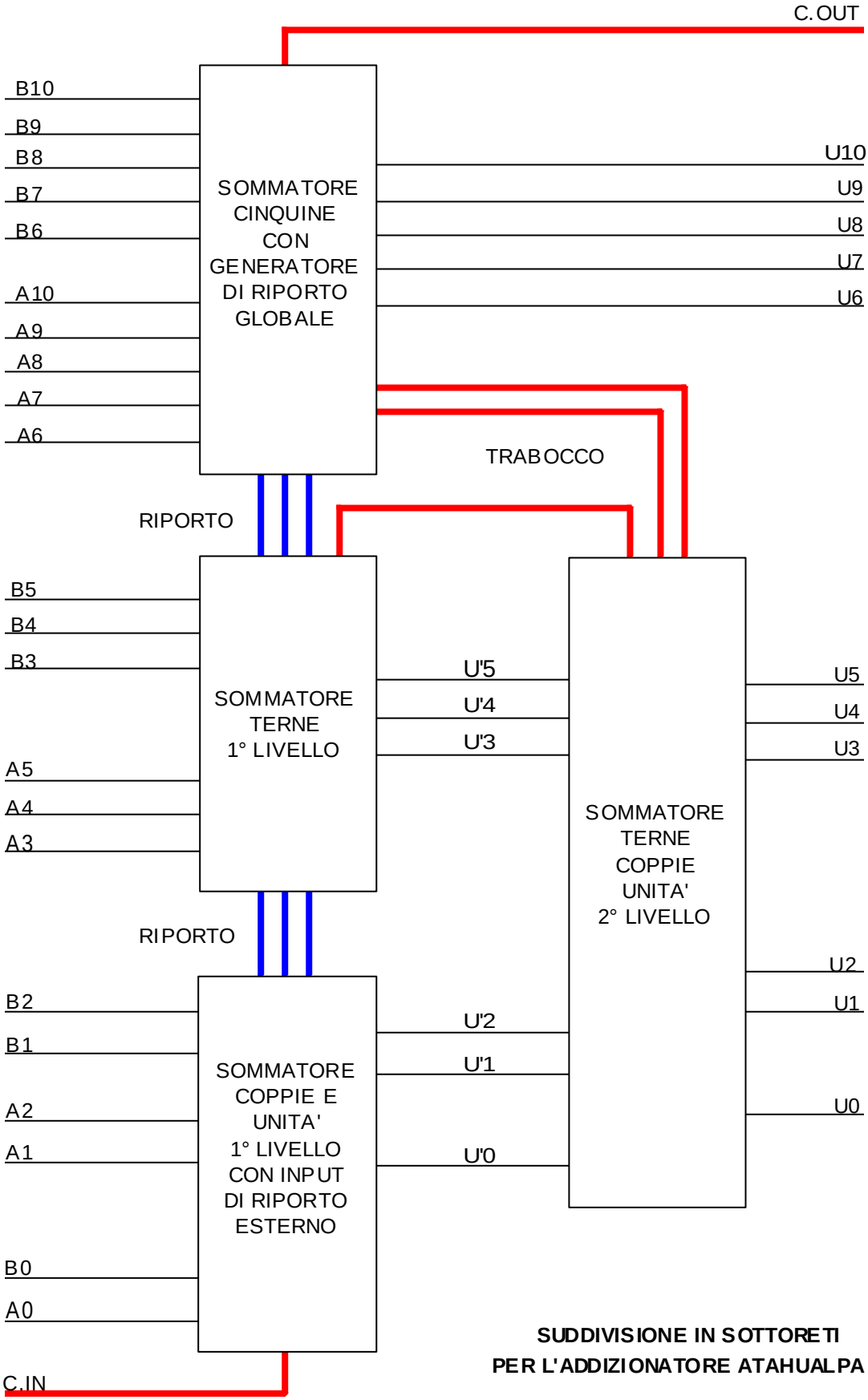

C.OUT
B10
B9
B8
B7
B6
A10
A9
A8
A7
A6
SOMMATORE CINQUINE CON GENERATORE DI RIPORTO GLOBALE
U10
U9
U8
U7
U6
TRABOCCO
RIPORTO
B5
B4
B3
A5
A4
A3
SOMMATORE TERNE 1° LIVELLO
U'5
U'4
U'3
SOMMATORE TERNE COPPIE UNITA' 2° LIVELLO
U5
U4
U3
RIPORTO
B2
B1
A2
A1
B0
A0
SOMMATORE COPPIE E UNITA' 1° LIVELLO CON INPUT DI RIPORTO ESTERNO
U'2
U'1
U'0
U2
U1
U0
C.IN
SUDDIVISIONE IN SOTTORETI
PER L'ADDIZIONATORE ATAHUALPA

Figura 11

Il primo blocco somma le coppie e le unità, compreso il riporto esterno, producendo in uscita unità, coppie intermedie e al massimo 3 riporti verso le terne. Infatti è pari a 3 il numero massimo di unità che possono essere presenti nella somma di due numeri. Per evitare eccessive interazioni tra i blocchi, si può realizzare un secondo blocco che sommi le terne compreso il riporto di terne proveniente dal primo blocco, producendo in uscita una terna intermedia e al massimo 3 riporti verso le cinquine. Infatti, con tale organizzazione, si travasano al massimo 5 terne in 3 cinquine lasciando 3 terne di uscita più una terna di trabocco interno (infatti è pari a 9 il numero massimo di terne che possono essere presenti). Il sommatore di cinquine, producendo in uscita un riporto ogni 8 cinquine, può generare al massimo 5 cinquine in uscita più 2 riporti intermedi sempre di peso 5. Per la definizione finale delle terne, delle coppie e dell'unità occorre un sommatore di secondo livello che provvede a sistemare la somma totale tenendo conto dei riporti interni delle cinquine e delle terne.

Algoritmi per la somma

Ma la strada che permette di rendere più flessibile il progetto è quella dell'implementazione di algoritmi specifici su una macchina programmabile. L'algoritmo di base è quello della somma che permette di eseguire anche tutte le altre operazioni. Data la ridondanza del codice, la propagazione dei riporti in entrambe le direzioni e l'utilizzo di più sottobasi che compongono la base principale, non ci si può aspettare algoritmi particolarmente semplici.
La struttura di calcolo è stata organizzata in matrici di 8 righe e 4 colonne che rappresentano rispettivamente il numero delle cinquine, delle terne, delle coppie e delle unità presenti in ciascun numero. Questa scelta semplifica di molto sia i calcoli che la rappresentazione dei numeri. Una terza dimensione determina il livello delle matrici per distinguere 2 o più operandi ed il risultato seguendo la citata modalità della notazione inversa polacca. Nella figura seguente si riassume la struttura matriciale:

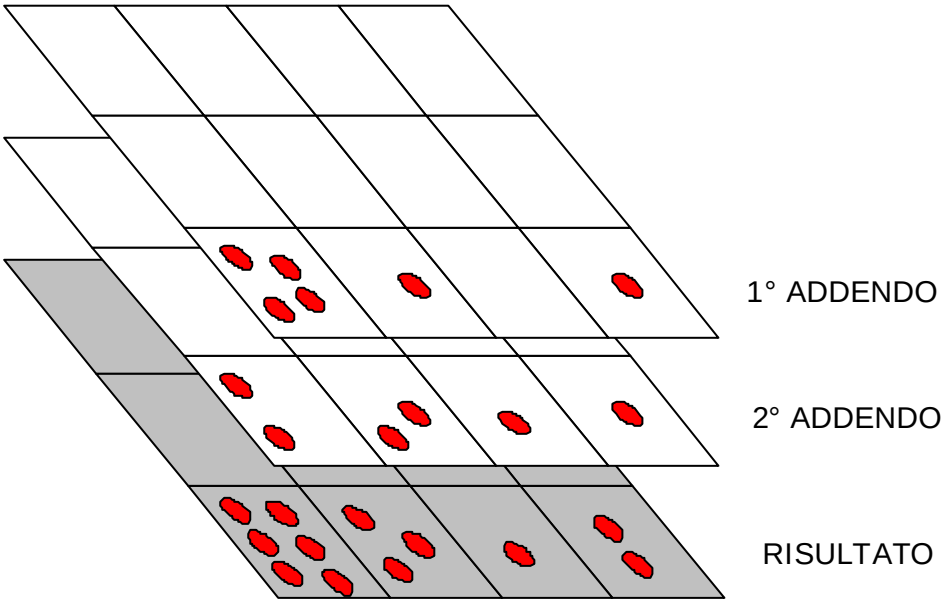


Figura 12

Matrice operativa (esempio con 3 sole righe)

La via certamente più semplice per effettuare la somma di due numeri con l'abaco è quella di "travasare" (ossia sommare) tutti i semi delle caselle corrispondenti di ciascun numero sul contenitore del risultato e procedere quindi alla normalizzazione seguendo le regole fondamentali. Quasi certamente gli Incas seguivano per la somma questo metodo particolarmente semplice ed efficace, che richiede solo un minimo di abilità numerica nella sistemazione dei semi secondo la forma canonica.
Nell'esempio gli addendi hanno sfondo bianco mentre il risultato grigio e, per semplicità, si limitano a 3 le righe per ogni numero. Sempre nella figura esemplificativa il primo addendo ha 4 cinquine, 1 terna e una unità del peso 40^0 (4x5+1x3+1=24), il secondo ha 2 cinquine, 2 terne, 1 coppia e una unità sempre del peso 40^0 (2x5+2x3+1x2+1=19). L'algoritmo per la somma prevede di addizionare direttamente il contenuto di tutte le caselle corrispondenti. In questo modo il risultato viene ad avere 6 cinquine, 3 terne, 1 coppia e 2 unità (6x5+3x3+1x2+2x1=43). Come si vede il risultato é corretto ma la forma di presentazione non rispetta la rappresentazione inca (le cinquine sono più di 5 e le unità più di uno). L'algoritmo primario messo a punto segue proprio questa semplice metodologia sfruttando le matrici per la somma delle caselle corrispondenti trasformando successivamente il risultato in forma canonica.

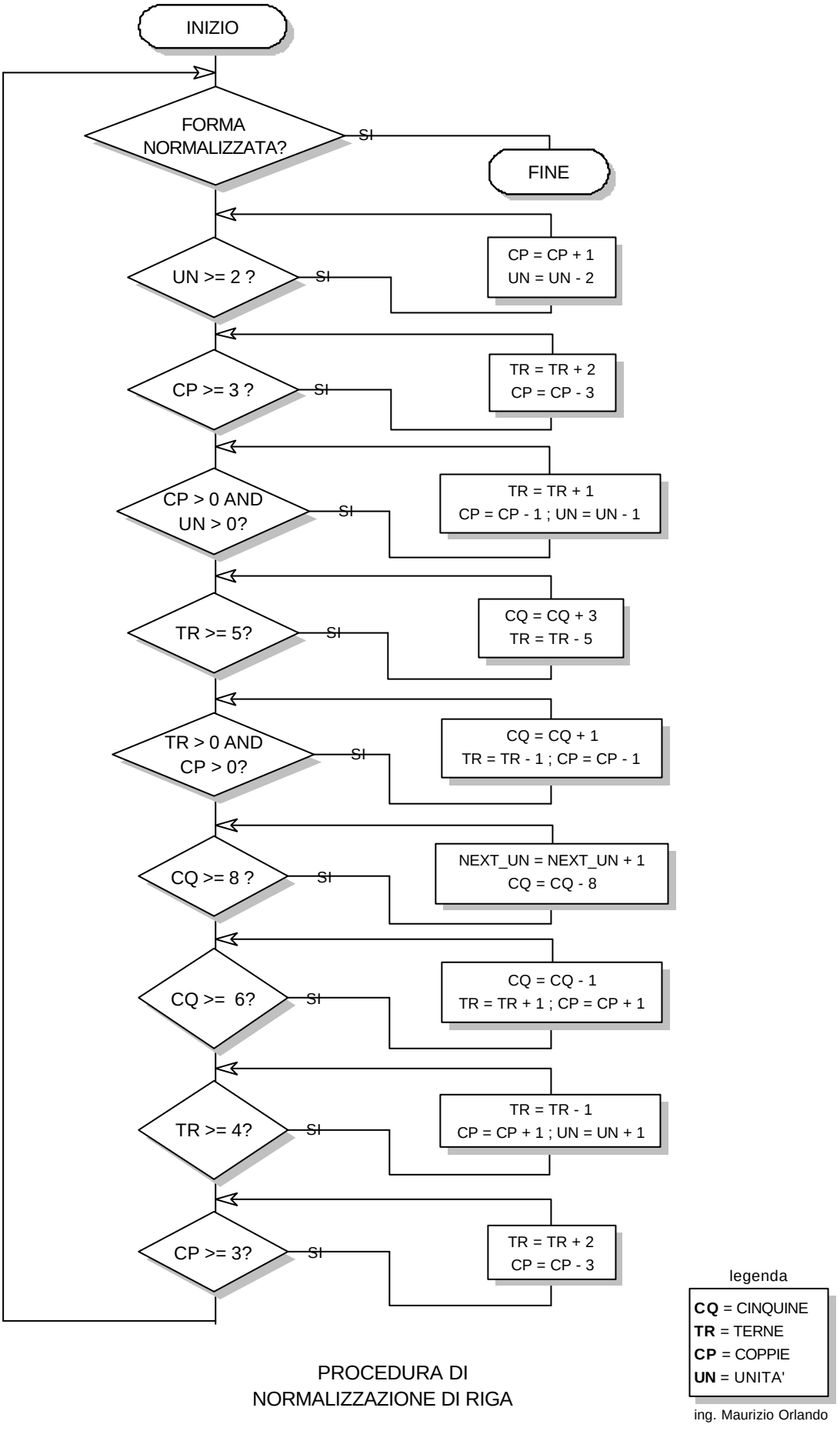
Il seguente listato si riferisce alla somma diretta di tutte le caselle degli addendi dell'intero abaco di 8 righe:

```
FOR I = 0 TO 7
  FOR J = 1 TO 4
    MAT(I,J,2) = MAT(I,J,0) + MAT(I,J,1)
  NEXT J
NEXT I
```

Anche molto semplice è la procedura di verifica della forma normalizzata che prevede l'uso di un flag di segnalazione (in riferimento ad una sola riga della matrice risultato):

```
NORMAL = TRUE
IF MAT(I,1,2) > 5 THEN NORMAL = FALSE
IF MAT(I,2,2) > 3 THEN NORMAL = FALSE
IF MAT(I,3,2) > 2 THEN NORMAL = FALSE
IF MAT(I,4,2) > 1 THEN NORMAL = FALSE
```

La figura seguente mostra l'algoritmo di normalizzazione contenente una procedura di aggiustamento iterativa in quanto potrebbe essere necessario più di un passaggio per raggiungere la forma canonica. Da notare che basta semplicemente alterare anche di poco la sequenza delle operazioni per non avere sempre assicurata la convergenza del metodo. Il controllo di testa della procedura è necessario dal momento che un risultato potrebbe presentarsi direttamente in forma normalizzata senza quindi necessità di aggiustamenti.



PROCEDURA DI
NORMALIZZAZIONE DI RIGA

Figura 13

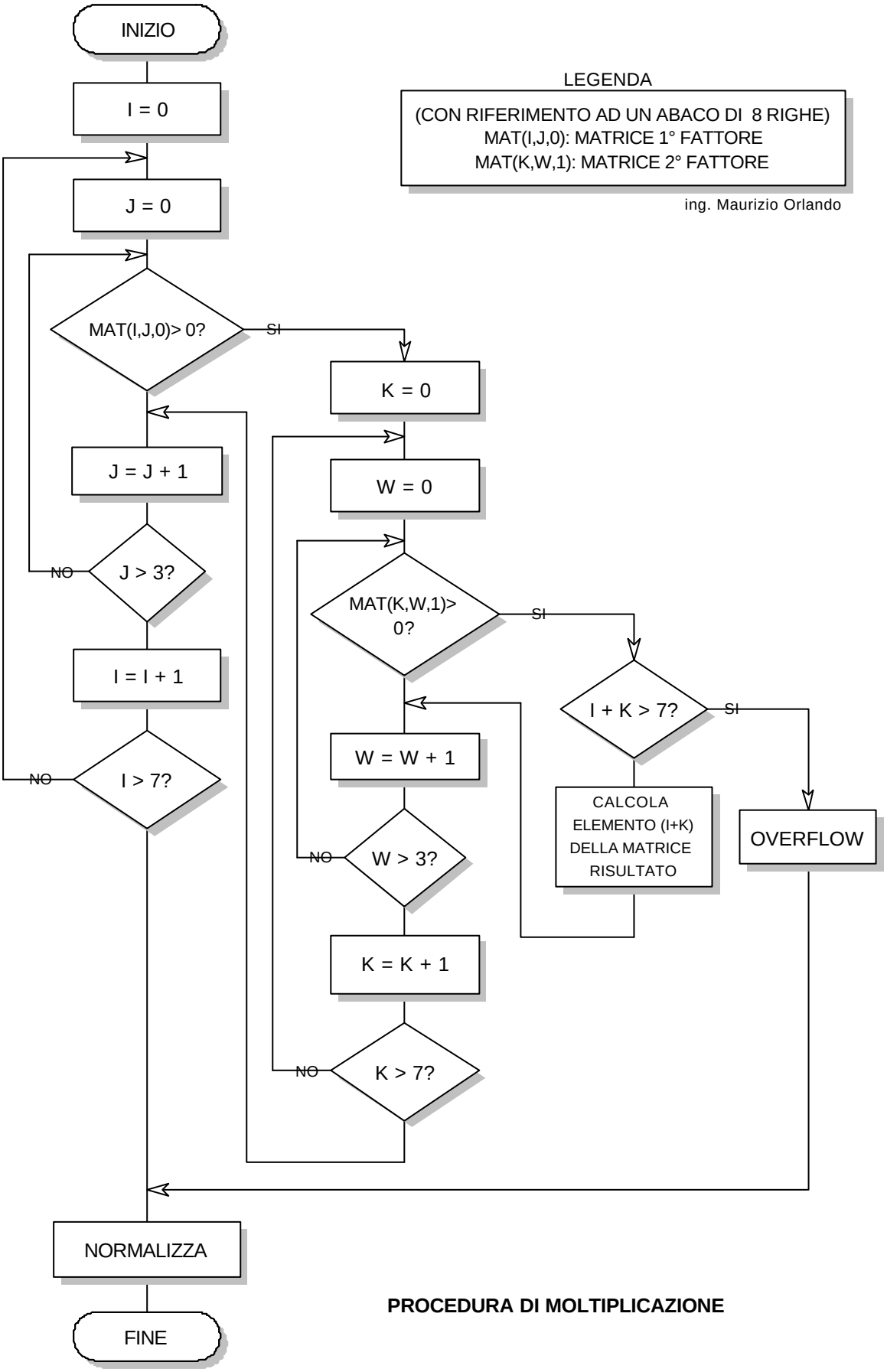

INIZIO
I = 0
J = 0
MAT(I,J,0)> 0?
SI
NO
K = 0
J = J + 1
W = 0
J > 3?
NO
MAT(K,W,1)> 0?
SI
I = I + 1
I + K > 7?
SI
W = W + 1
I > 7?
NO
CALCOLA ELEMENTO (I+K) DELLA MATRICE RISULTATO
OVERFLOW
W > 3?
NO
K = K + 1
K > 7?
NO
NORMALIZZA
FINE
LEGENDA
(CON RIFERIMENTO AD UN ABACO DI 8 RIGHE)
MAT(I,J,0): MATRICE 1° FATTORE
MAT(K,W,1): MATRICE 2° FATTORE
ing. Maurizio Orlando
PROCEDURA DI MOLTIPLICAZIONE

Figura 14

La procedura di normalizzazione contiene tutte le proprietà elementari dell'abaco inca, quelle prima esposte relative al riporto verso sinistra ed altre che completano il riporto sempre a sinistra come: 8 cinquine danno luogo ad un riporto unitario nella riga successiva, 5 terne generano 3 cinquine. Infine sono contemplati tutti i casi che generano riporto verso destra come ad esempio accade se le cinquine risultano 7 e non possono dar luogo ad un riporto superiore.
Per semplicità, una volta implementata la somma, tramite il metodo del complemento alla base, si può passare alla sottrazione, in riferimento ovviamente alla base 40. Da notare che tale complemento presenta le stessa facilità del binario in quanto basta negare ogni seme di tutte le righe interessate e aggiungere una unità. Non semplicissima invece risulta la determinazione del segno perché la comparazione tra numeri nel codice inca è complicata dal fatto che un numero con semi di peso inferiore può essere maggiore di un altro con peso superiore (ad esempio 2 terne sono maggiori di una cinquina). Con somme successive si ottiene la moltiplicazione e con sottrazioni successive si ottiene infine la divisione a completamento delle 4 operazioni. Ma dato che l'abaco riesce a trattare numeri estremamente grandi, il metodo delle somme o sottrazioni successive si rileva troppo lento ed è opportuno ricorrere a specifici algoritmi più efficaci. Ad esempio la figura 14 mostra l'algoritmo per la procedura di moltiplicazione di due fattori basato sul metodo polinomiale:

$$(A \cdot 40^{N} + B \cdot 40^{M} + \ldots) \cdot (C \cdot 40^{N'} + D \cdot 40^{M'} + \ldots) =$$

$$A \cdot C \cdot 40^{N+N'} + A \cdot D \cdot 40^{N+M'} + B \cdot C \cdot 40^{M+N'} + B \cdot D \cdot 40^{M+M'} + \ldots$$

L'algoritmo sintetizzato non prevede il complesso calcolo delle potenze in base 40, ma gestisce la somma dei relativi esponenti come indice per determinare direttamente la posizione dei vari termini del prodotto. Si ottiene così la moltiplicazione di due fattori con un solo passaggio in modo semplice e veloce. Come punto di partenza nello studio della matematica andina è stata realizzata la calcolatrice Atahualpa rispettando il più possibile il contesto storico, ricercando procedure create su misura per il codice inca nell'evidente sforzo di riprodurre le tecniche di calcolo più elementari. Interessanti anche i parallelismi con alcune recenti applicazioni sui numeri di Fibonacci[1] [2] [3] [4] [5] con i quali si trova corrispondenza diretta nell'abaco inca. A tal proposito anche soluzioni neurali possono comunque fornire vantaggiose soluzioni nelle operazioni più complesse come dimostrano alcune ricerche matematiche[6] [7] che sperimentano la possibilità di ottimizzare le strutture di calcolo in ambienti diversi dal binario.
Ma ulteriori ricerche più approfondite sull'abaco hanno evidenziato eccezionali proprietà in grado di semplificare i calcoli in modo incredibile e si rimanda la descrizione dei relativi metodi a successiva pubblicazione.

Maurizio Orlando

m.orlando@quipus.it

[1] K. Egiazarian and J. Astola: *Discrete Orthogonal Transforms Based on Fibonacci-type Recursions.*
[2] R. Stankoviæ, M. Stankoviæ, J. Astola and K. Egiazarian: *Bit-Level and Word-Level Polynomial Expression for Functions in Fibonacci Interconnection Tipologies.*
[3] C. Frougny: *Fibonacci Representations and Finite Automata.*
[4] R. M. Capocelli and P. Cull: *Applications of Fibonacci Numbers*, G.E. Bergum et al. (eds.) 1990, pp. 57-62.
[5] W.-J. Hsu: *Fibonacci cubes – a new interconnetcion topology*, IEEE Transactions on Parallel and Distributed Systems, vol. 4, No. 1, Jan 1993, pp. 3-12.
[6] M. Yacoub and A. Saudi: *Recurrent Neural Networks and Fibonacci Numeration System*, Proceedings of 1993 International Joint Conference on Neural Networks.
[7] C. Frougny: *Systèmes de Numération linéaires et automates finis. Thése de Doctorat d'état*, Université Paris VII, 1989