di GIORGIO OBER (GIOBE2000)

l'angolo di Mr A.KEER

(parte sedicesima)

PROGETTARE con le PORTE LOGICHE

Unità Aritmetico Logica

La rassegna delle macchine combinatorie destinate al supporto delle operazioni aritmetiche non può prescindere dalle ALU, sofisticate macchine plurifunzionali, parte integrante persino dei moderni microprocessori.

Nelle puntate precedenti abbiamo trattato con la massima attenzione tutti i mattoni principali della logica combinatoria, a cominciare da quelli fondamentali fino a considerare le complesse strutture logiche chiamate a realizzare la somma e la sottrazione; sembra naturale quindi terminare la lunga trattazione con questo importante componente, in grado di assicurare sia le operazioni logiche sia quelle aritmetiche.

Prima di entrare dei dettagli dei componenti disponibili sul mercato desidero sottolineare che la ALU (Aritmetic Logic Unit) è una inalienabile parte integrante anche di ogni CPU (Central Processing Unit), termine con cui si definiscono sinteticamente i microprocessori, sia quelli di tipo x86 (contenuti in molti Personal Computer) che quelli destinati ai sistemi embedded (cioè quelli progettati espressamente per una determinata applicazione).

Per aumentare la loro potenza di elaborazione, le moderne CPU contengono una seconda unità di calcolo, detta FPU (Floating Point Unit), specializzata nel trattamento dei numeri razionali frazionari ed irrazionali, indispensabili nel calcolo scientifico (logaritmi, funzioni trigonometriche, divisioni, radici, ecc..), richiesto dagli applicativi software di ingegneria (come i vari CAD, Computer Aided Design) ma anche dai più noti videogiochi 3D e nel trattamento dello streaming video, sempre più assetato di risorse.

L'unità FPU (un tempo situata su un chip dedicato, detto Coprocessore matematico o x87) è citata solo per completezza e può ritenersi la strutttura in grado di completare i servizi elementari assicurati dalla ALU, rivolti al processo dei numeri naturali positivi e negativi e alle operazioni della logica booleana.

ALU (TTL): 74LS181,74LS381,74LS382

La serie **TTL** prevede 3 ALU in grado di operare in parallelo su 2 numeri a 4 bit, espressi in binario o in BCD: quello più noto è il **74LS181**, detto **4-Bit Arithmetic Logic Unit**; questo dispositivo è stato progettato per includere (senza l'utilizzo di alcuna circuiteria esterna) ogni dettaglio circuitale necessario a realizzare 16 operazioni aritmetiche e 16 diverse funzioni booleane. La *Figura 1* mostra il suo *pin-out*.

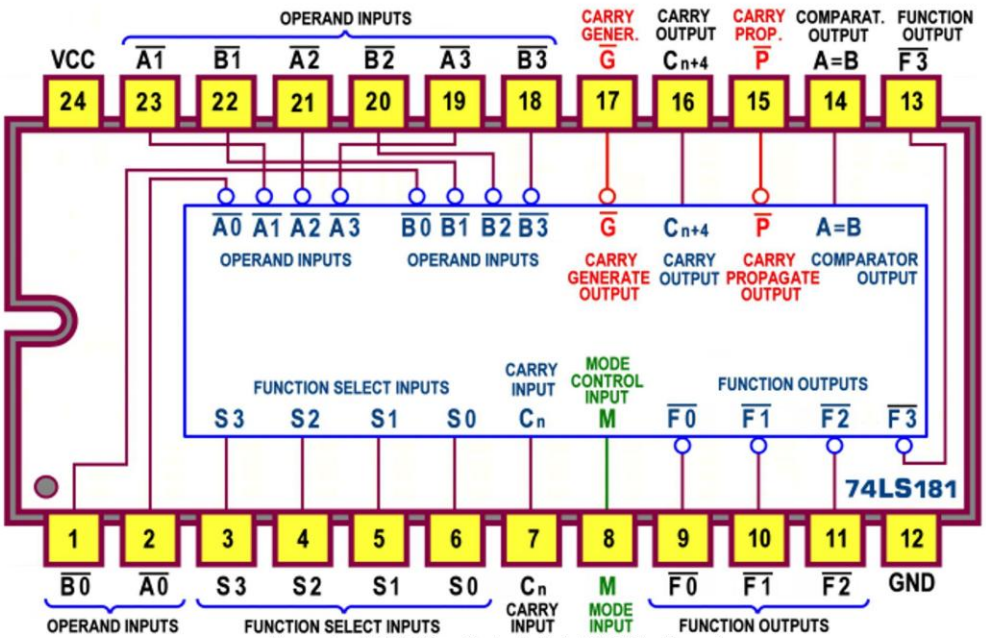

Figura 1 - 4-Bit Arithmetic Logic Unit 74LS181: Pin-out

Già la dimensione del dispositivo (ospitato in un contenitore a 24 piedini, 24-Lead Plastic Dual-In-Line Package) tradisce una sua certa complessità funzionale; la *Figura 2* riorganizza i numerosi segnali coinvolti nel corrispondente schema funzionale.

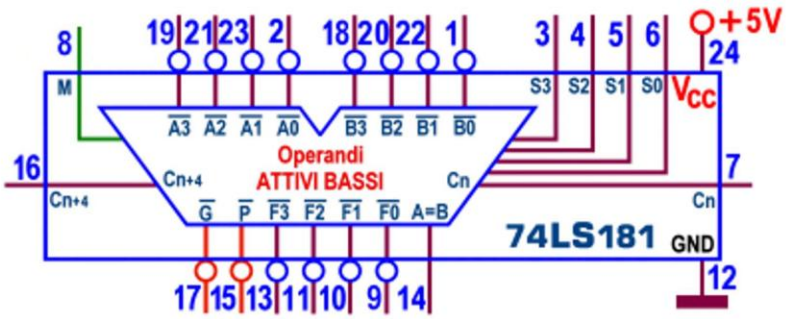

Figura 2 - 4-Bit Arithmetic Logic Unit 74LS181: Schema funzionale (in logica negativa)

Possiamo notare che sia le linee d'ingresso dei 2 operandi (Operand Inputs, $A_3A_2A_1A_0$ e $B_3B_2B_1B_0$) che quelle d'uscita (Function Outputs, $F_3F_2F_1F_0$) sono attive basse; poichè il **74LS181** può essere usato anche con operandi e uscite attivi alti, sarà necessario riprendere più avanti le seguenti considerazioni per mettere in evidenza le inevitabili differenze.

In ogni caso, le 5 linee di controllo garantiscono sugli operandi 2^5=32 diverse operazioni; in particolare, se l'ingresso di modo (M, Mode Control Input) è alto, impone le 16 di tipo logico (**AND**, **NAND**, **OR**, **NOR**, **OREX**, **NOREX**, più altre 10), mentre con M=0 renderà disponibili le 16 di tipo aritmetico (addizione, sottrazione, confronto, raddoppio [scorrimento di un bit a sinistra], più altre 12).

Le rimanenti 4 linee (Function Select Inputs, $S_3S_2S_1S_0$) stabiliscono il valore (logico o aritmetico) assunto dalle uscite, in accordo con la Tabella di Funzione, proposta in *Figura 3*.

LINEE DATI IN INGRESSO E USCITE ATTIVE BASSE

SELECTION				M=1 LOGIC FUNCTIONS	M=0 ARITHMETIC OPERATIONS	
S3	S2	S1	S0		C_n=0 (senza carry)	C_n=1 (con carry)
0	0	0	0	$\overline{A}$	A meno 1	A
0	0	0	1	$\overline{AB}$	AB meno 1	AB
0	0	1	0	$\overline{A}$+B	A$\overline{B}$ meno 1	A$\overline{B}$
0	0	1	1	1 logico	meno 1 (Complemento a 2)	Zero
0	1	0	0	$\overline{A+B}$	A più (A+$\overline{B}$)	A più (A+$\overline{B}$) più1
0	1	0	1	$\overline{B}$	AB più (A+$\overline{B}$)	AB più (A+$\overline{B}$) più1
0	1	1	0	$\overline{A \oplus B}$	A meno B meno1	A meno B
0	1	1	1	A+$\overline{B}$	(A+$\overline{B}$)	(A+$\overline{B}$) più1
1	0	0	0	$\overline{A}$B	A più (A+B)	A più (A+B) più1
1	0	0	1	A⊕B	A più B	A più B più1
1	0	1	0	B	A$\overline{B}$ più (A+B)	A$\overline{B}$ più (A+B) più1
1	0	1	1	A+B	(A+B)	(A+B) più1
1	1	0	0	0 logico	A più A [#]	A più A più1
1	1	0	1	A$\overline{B}$	AB più A	AB più A più1
1	1	1	0	AB	A$\overline{B}$ più A	A$\overline{B}$ più A più1
1	1	1	1	A	A	A più1

Figura 3 - 4-Bit Arithmetic Logic Unit 74LS181: Tabella (in logica negativa)

L'analisi paziente della voluminosa tabella permette di scoprire molti importanti dettagli; innanzi tutto è stato necessario evitare la confusione tra l'operazione di somma logica (OR) da quella aritmetica, affidando alla prima il classico segno "+" e alla seconda la parola "più"; pur non causando ambiguità alla sottrazione (esclusivamente aritmetica) è coerentemente affidata la parola "meno" in vece del classico segno "-".

Una tra le funzioni (indicata in Tabella con [#]) merita un commento: la somma aritmetica di A con se stessa equivale a raddoppiare (moltiplicare per 2) il valore di A; per esempio, sommando $A=(0110)_2=(6)_{10}$ con se stesso otteniamo $F=A+A=A*2=(1100)_2=(12)_{10}$.

Osservando il risultato binario ci rendiamo conto che esso è uguale al valore di A spostato di un bit verso sinistra (shift left) con ingresso di un bit a 0 da destra; si tratta di una tecnica molto usata nella programmazione assembly dei processori, per garantire la moltiplicazione di un numero per una potenza di 2.

Tutti i risultati aritmetici generati da questo dispositivo sono in *Complemento a 2*; per esempio per eseguire la somma "A più B" si pone 1001 sugli ingressi di selezione $S_3S_2S_1S_0$ (di solito senza la presenza del *Riporto* esterno C_n) mentre per la sottrazione "A meno B" il codice di selezione sarà 0110, attivando anche l'ingresso C_n, in questo caso necessario per trasformare il *Complemento a 1* del *Sottraendo*, generato internamente.

Va detto anche che numerose funzioni sono di uso poco probabile e non saranno praticamente mai utilizzate; la cosa non deve stupire, data l'articolata struttura interna, chiamata ad assicurare le operazioni più classiche e a combinare i loro circuiti per ottenerne altre.

Importante sottolineare, invece, che il componente contiene le particolari reti combinatorie (basate sulla tecnica di "*previsione del Riporto*", ***Carry Look Ahead***) in grado di generare in anticipo il valore del *Riporto* (o il *Prestito*) a 4 bit nelle funzioni aritmetiche; per rendere

possibile la creazione di sommatori (o sottrattori) in grado di gestire parole multiple di 4 bit, collegando più ALU in cascata, sono previste la linea d'ingresso C_n (Carry Input) e le 3 linee d'uscita C_{n+4} (Carry Output), G (Carry Generate Output) e P (Carry Propagate Output).

Le prime 2 assicurano la tipica gestione "*a propagazione del Riporto*" (***Ripple Carry***) in cui l'uscita C_{n+4} della ALU associata alla parte meno significativa dei dati da operare è collegata all'ingresso C_n della ALU successiva; la *Figura 4* mostra il circuito di una ALU a 8 bit costruita con questa tecnica.

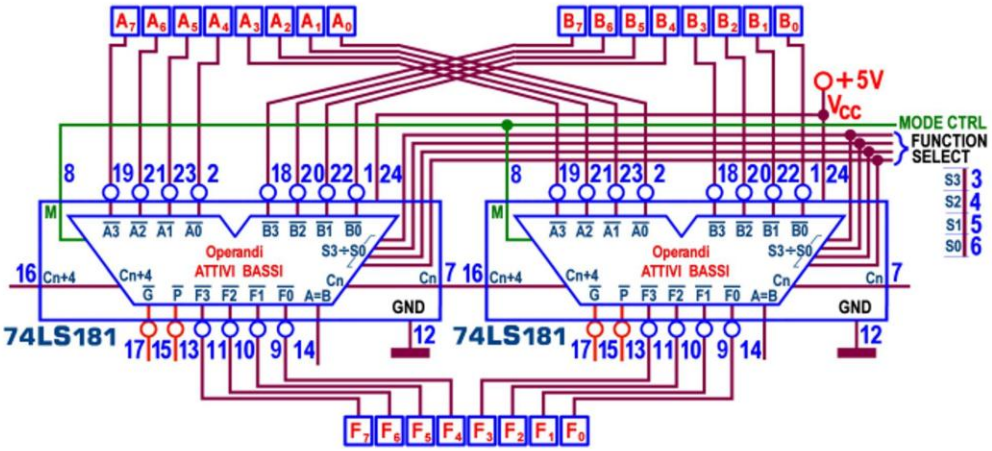

Figura 4 - 4-Bit Arithmetic Logic Unit 74LS181: Ripple Carry ALU a 8 bit

La puntigliosa trattazione delle puntate precedenti ha messo in luce i problemi di scarsa velocità intrinseci dei sommatori/sottrattori di tipo ***Ripple Carry***; questo fatto ha reso indispensabile la presenza di uno specifico gestore esterno (il **74LS182**, che riprenderemo più avanti) in grado di garantire il ***Carry Look Ahead*** anche tra gli stadi ALU in cascata (senza dover ricorrere al *Riporto* a ondulazione), supportata appunto dalle linee d'uscita G e P (group-carry lookahead).

Il segnale di Controllo di Modo è in grado di influenzare la disponibilità e il coinvolgimento dei *Riporti* interni; quando M=0 (ALU come generatore di funzioni aritmetiche) essi saranno abilitati, mentre con M=1 ciascuna uscita dipende solo dai dati d'ingresso, A e B, il valore di C_n è indifferente (e viene ignorato) e nessun *Riporto* sarà propagato tra i moduli interni della ALU.

Il componente si presta anche a confrontare tra loro i valori dei 2 operandi: l'uscita A+B (Comparator Output) passa a livello alto quando essi sono uguali, ma per poter disporre di questa informazione il dispositivo deve essere in modalità aritmetica (M=1) e programmato (con *Riporto* esterno C_n attivo) per la sottrazione ($S_3S_2S_1S_0$=0110).

Con l'aiuto dell'uscita di *Riporto* C_{n+4} (sempre in modalità "sottrazione") è naturale poter disporre anche delle informazioni sulla grandezza relativa A>B e A<B degli operandi.

L'uscita A+B è di tipo *open collector*, per un facilitare il collegamento in "wire-AND" con le stesse linee delle altre ALU in cascata, se si desidera un confronto su dati di più grandi di 4 bit.

La *Figura 5* mostra lo *schema pratico* con ingressi di dato e uscite in logica negativa mentre il *simbolo logic*o predisposto dallo *standard IEEE* è visibile in *Figura 6*. In quest'ultima appare evidente un modulo (Common Control Block) destinato al controllo comune dei blocchi interni; lo standard indica con la lettera M la parola di controllo formata dai 4 ingressi di selezione e dal segnale di modo M, in corrispondenza dei quali viene mostrato il peso (esponente della potenza di 2) del rispettivo bit della parola stessa.

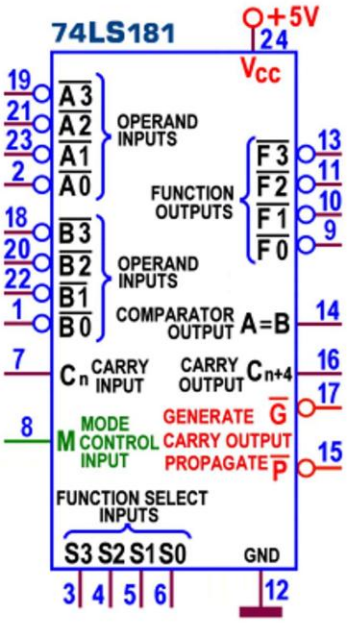

Figura 5 - 4-Bit Arithmetic Logic Unit 74LS181: Schema pratico (in logica negativa)

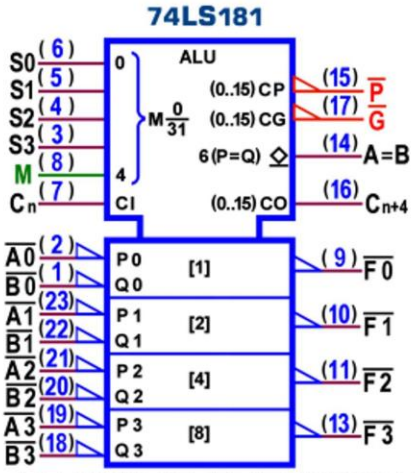

Figura 6 - 4-Bit Arithmetic Logic Unit 74LS181: Simbolo logico ANSI/IEEE Std. 91-1984 (in logica negativa)

Ad essa viene affiancato il range 0-31 (espresso come frazione 0/31) per esplicitare la disponibilità di 32 diversi comandi; lo standard evidenzia inoltre che le 3 uscite di *Riporto*, G (CG, Carry Generate), P (CP, Carry Propagate) e C_{n+4} (CO, Carry Output), sono attivate nelle modalità 0-15 e che l'uscita A=B è attiva in modalità 6 (0110, sottrazione aritmetica).

La parte bassa del simbolo mostra i 4 singoli blocchi della ALU, etichettati con i pesi dei bit che essi processano.

Poiché il **74LS181** può essere usato anche con operandi e uscite attivi alti i datasheet prevedono una seconda Tabella di Funzione, dalla quale risulta (vedi *Figura 7*) che le funzioni logiche (ottenute con M=1 in corrispondenza dello stesso codice di selezione) sono esattamente "*duali*" di quelle presentate dalla Tabella di *Figura 3*; ricordo che il principio di

dualità prevede che da ogni espressione logica vera può esserne ottenuta un'altra, altrettanto vera e detta *duale*, semplicemente scambiando ogni operatore AND con un operatore OR e ogni 1 con uno 0 (e viceversa).

Tra le operazioni aritmetiche (ottenute con M=0) alcune sono identiche (quelle con codice 0011, 0110, 1001 e 1100, curiosamente multiplo di 3 decimale), altre (quelle legate alla somma) sono duali per la parte logica coinvolta, e le rimanenti sono le stesse ma in posizione diversa.

LINEE DATI IN INGRESSO E USCITE ATTIVE ALTE

SELECTION				M=1 LOGIC FUNCTIONS	M=0 ARITHMETIC OPERATIONS	
S3	S2	S1	S0		$\overline{C_n}=1$ (senza carry)	$\overline{C_n}=0$ (con carry)
0	0	0	0	$\overline{A}$	A	A più 1
0	0	0	1	$\overline{A+B}$	(A+B)	(A+B) più 1
0	0	1	0	$\overline{A}B$	$(A+\overline{B})$	$(A+\overline{B})$ più 1
0	0	1	1	0 logico	meno 1 (Complemento a 2)	Zero
0	1	0	0	$\overline{AB}$	A più $A\overline{B}$	A più $A\overline{B}$ più 1
0	1	0	1	$\overline{B}$	(A+B) più $A\overline{B}$	(A+B) più $A\overline{B}$ più 1
0	1	1	0	$A\oplus B$	A meno B meno 1	A meno B
0	1	1	1	$A\overline{B}$	$A\overline{B}$ meno 1	$A\overline{B}$
1	0	0	0	$\overline{A}+B$	A più AB	A più AB più 1
1	0	0	1	$\overline{A\oplus B}$	A più B	A più B più 1
1	0	1	0	B	$(A+\overline{B})$ più AB	$(A+\overline{B})$ più AB più 1
1	0	1	1	AB	AB meno 1	AB
1	1	0	0	1 logico	A più A [#]	A più A più 1
1	1	0	1	$A+\overline{B}$	(A+B) più A	(A+B) più A più 1
1	1	1	0	A+B	$(A+\overline{B})$ più A	$(A+\overline{B})$ più A più 1
1	1	1	1	A	A meno 1	A

Figura 7 - 4-Bit Arithmetic Logic Unit 74LS181: Tabella (in logica positiva)

La *Figura 8* e la *Figura 9* mostrano rispettivamente lo *schema funzionale* e il *simbolo logico* secondo lo *standard IEEE* per la gestione del **74LS181** in logica positiva (cioè con operandi e uscite attivi alti); da notare che, in questo caso, i *Riporti* per la gestione ***Ripple Carry*** (C_n e C_{n+4}) sono ora attivi bassi, per mantenere corretta l'operazione aritmetica svolta.

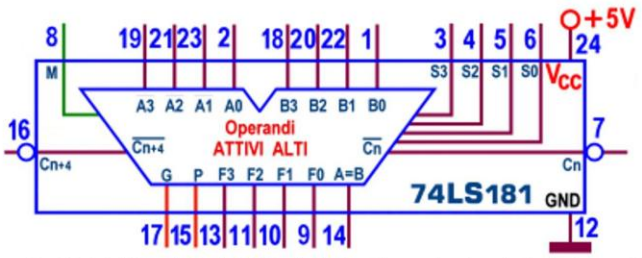

Figura 8 - 4-Bit Arithmetic Logic Unit 74LS181: Schema funzionale (in logica positiva)

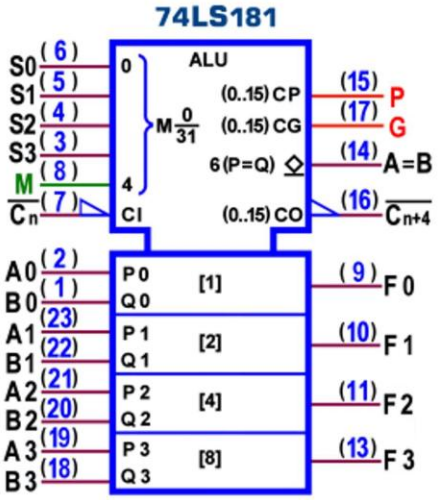

Figura 9 - 4-Bit Arithmetic Logic Unit 74LS181: Simbolo logico ANSI/IEEE Std. 91-1984 (in logica positiva)

La potenza dissipata massima è di **185 mW**; dati i numerosi punti di vista possibili (logici e aritmetici) la valutazione del ritardo di propagazione massimo (misurato con carico di **15**pF) è necessariamente differenziata, ma mediamente pari a **33**ns tra gli ingressi di dato e le uscite, per entrambe le transizioni $\mathbf{t_{PLH}}$ e $\mathbf{t_{PHL}}$); nella prospettiva di collegare più ALU in cascata è interessante il ritardo previsto nel passaggio da C_n a C_{n+4}, pari a **20**ns.

La consultazione delle Tabelle di Funzione dell'integrato **74LS181** ha evidenziato che numerose tra le funzioni offerte sono praticamente inutili; per questa ragione i costruttori hanno reso disponibile una ALU più compatta: il **74LS381**, ancora una **4-Bit Arithmetic Logic Unit**, ma in grado di garantire solo 8 Funzioni, quelle di uso più frequente.

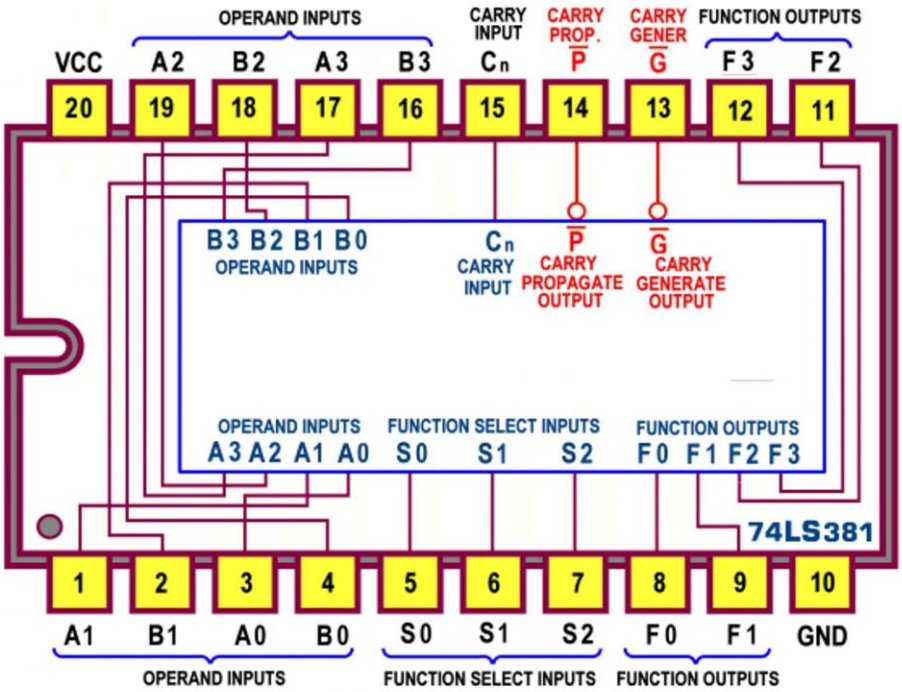

Figura 10 - 4-Bit Arithmetic Logic Unit 74LS381: Pin-out

La *Figura 10* mostra il suo *pin-out* e la *Figura 11* riassume le sue risorse nel classico schema funzionale.

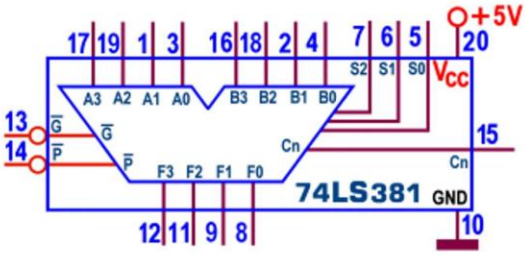

Figura 11 - 4-Bit Arithmetic Logic Unit 74LS381: Schema funzionale

In questo componente la linea di selezione di Modo non è necessaria ed esso dispone ora di sole 3 linee di selezione ($S_2S_1S_0$) in grado di garantire 2^3=8 diverse operazioni: 3 di tipo logico (**AND**, **OR**, **OREX**, senza bisogno di alcun circuito esterno), 3 di tipo aritmetico (una addizione e 2 sottrazioni) e 2 situazioni nelle quali le uscite possono essere forzate a 0 (clear) o a 1 (preset).

La *Figura 12* le riassume tutte, nella Tabella di Funzione fornita dai datasheet.

SELECTION			LOGIC/ARITHMETIC FUNCTIONS
S2	S1	S0	
0	0	0	0 0 0 0
0	0	1	B meno A meno 1 più C_n
0	1	0	A meno B meno 1 più C_n
0	1	1	A più B più C_n
1	0	0	A⊕B
1	0	1	A+B
1	1	0	A·B
1	1	1	1 1 1 1

Figura 12 - 4-Bit Arithmetic Logic Unit 74LS381: Tabella delle Funzioni

Le operazioni aritmetiche sono eseguite con operandi A e B attivi sia alti che bassi, con uscite F soggette alla stessa logica; quando si seleziona la funzione Sottrazione è al solito necessario forzare a 1 (con operandi attivi alti) o a 0 (con operandi attivi bassi) l'ingresso C_n del dispositivo meno significativo.

La presenza delle 2 linee d'uscita G (Carry Generate) e P (Carry Propagate) è il presupposto per assicurare la somma (o la sottrazione) veloce di parole multiple di 4 bit, collegando più ALU in cascata; questi segnali, con l'aiuto dello specifico gestore esterno **74LS182**, garantiscono la tecnica di ***Carry Look Ahead*** in grado di "*prevedere*" in anticipo il valore del *Riporto* (o del *Prestito*).

La *Figura 13* mostra lo *schema pratico* mentre il *simbolo logico IEEE* (*Figura 14*) mette in evidenza gli stessi moduli e le stesse convenzioni proposte in *Figura 9* per il **74LS181**.

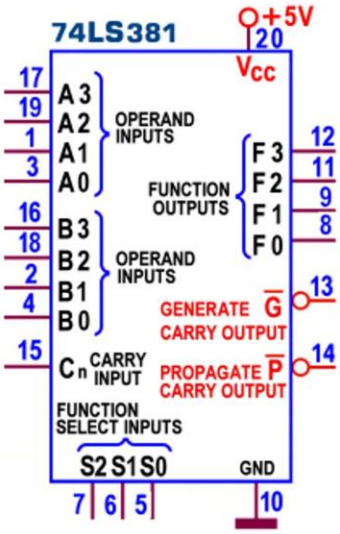

Figura 13 - 4-Bit Arithmetic Logic Unit 74LS381: Schema pratico

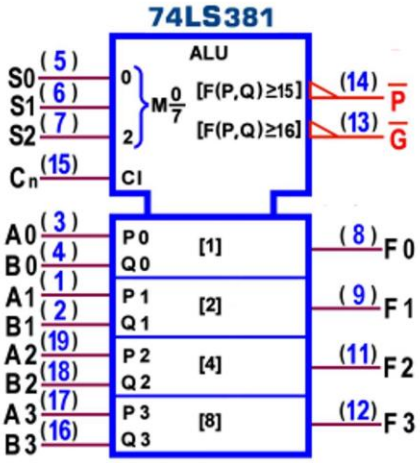

Figura 14 - 4-Bit Arithmetic Logic Unit 74LS381: Simbolo logico ANSI/IEEE Std. 91-1984

L'integrato **74LS382** è una variante lenta del **74LS381**: l'unica differenza è legata al modo con cui essi gestiscono il *Riporto* quando più ALU dello stesso tipo sono collegate in cascata al fine di poter operare (ovviamente in modo aritmetico) su dati di più di 4 bit.

Al posto delle linee d'uscita "*di gruppo*" G e P (tipiche della veloce tecnica ***Look Ahead*** appena descritta per il **74LS381**) il nuovo modello si appoggia alle linee d'uscita C_{n+4} (Carry Output) e OVR (Overflow) lasciando intendere di ricorrere alla gestione "*a propagazione del Riporto*" (***Ripple Carry***), ancora utile quando non si hanno problemi di velocità. La *Figura 15* mostra il suo *pin-out*.

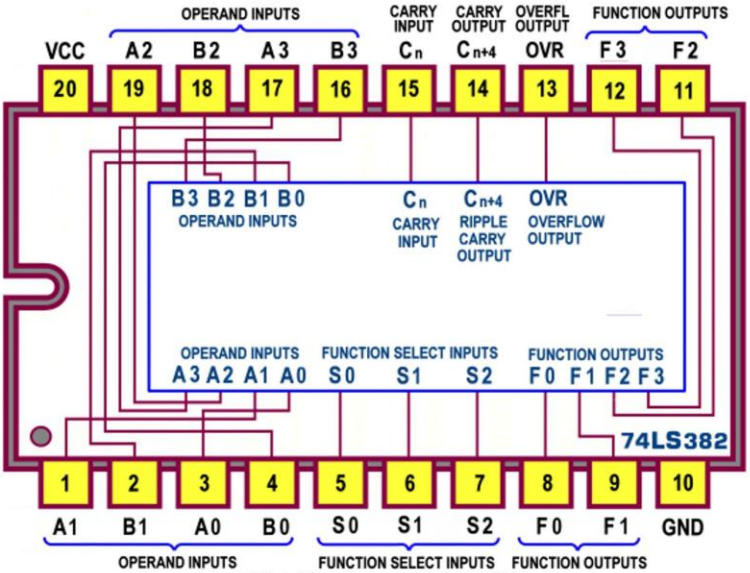

Figura 15 - 4-Bit Arithmetic Logic Unit 74LS382: Pin-out

Lo schema funzionale (*Figura 16*) può essere utilizzato come modulo di una serie di ALU in cascata, la cui linea d'uscita C_{n+4} sarà collegata a quella d'ingresso C_n dello stadio successivo, mentre per la C_n del primo stadio valgono le predisposizioni più volte sottolineate in precedenza; l'uscita OVR segnala la condizione overflow nelle operazioni in *Complemento a 2* ed è logicamente uguale al OREX di C_{n+3} con C_{n+4}; verrà utilizzata solo quella della ALU più significativa.

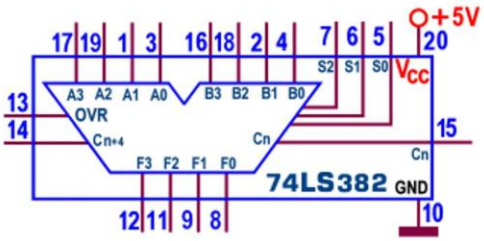

Figura 16 - 4-Bit Arithmetic Logic Unit 74LS382: Schema funzionale

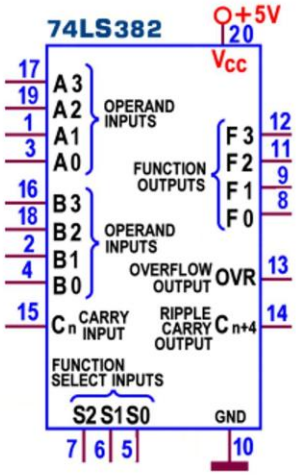

Figura 17 - 4-Bit Arithmetic Logic Unit 74LS382: Schema pratico

In *Figura 17* si può vedere lo schema pratico.

Per entrambi i componenti la potenza dissipata massima è di **325 mW** e i ritardi di propagazione massimi (con carico di **2**kohm/**15**pF) sono mediamente pari a **30**ns (tra ingressi di dato e uscite) e **52**ns (tra ingressi di selezione e uscite); nella gestione di più ALU in cascata il ritardo previsto tra ingressi dato e G o P (**74LS381**) è pari a **33**ns, e tra C_n e C_{n+4} (**74LS382**) è pari a **20**ns.

In chiusura, tra i componenti **TTL** di questa categoria merita citazione il **74LS681**, definito **4-Bit Parallel Binary Accumulator**, costituito da una ALU e di 2 potenti registri.

La parte ALU è sostanzialmente uguale al **74LS181**, cioè è in grado di realizzare 16 operazioni aritmetiche e 16 funzioni booleane e di gestire strutture in cascata sia con veloce tecnica ***Carry Look Ahead*** (con G e P, insieme al **74LS182**) che ***Ripple Carry*** (con C_n e C_{n+4}).

Entrambi i registri interni possono essere letti e/o scritti attraverso 4 linee bidirezionali, sul fronte di salita di un segnale di Clock appositamente previsto; le rispettive uscite forniscono gli operandi della ALU.

Uno dei 2 registri si comporta da semplice deposito di memoria, mentre l'altro dispone di una logica di controllo in grado di programmarlo per essere gestito in 8 modi diversi, per esempio per operare lo scorrimento (shift) dei suoi bit; esso ha anche l'importante compito di accumulatore, occupandosi di ospitare il risultato delle operazioni ALU.

Si tratta dunque di un componente pieno di fascino, in sostanza molto simile ad un "*buona parte di CPU*", tra l'altro espandibile per trattare parole più grandi di 4 bit.

ALU (CMOS): 4581, 40181

La serie **CMOS** mette a disposizione 2 componenti di tipo **4-Bit Arithmetic Logic Unit**, il **4581** e il **40181**, funzionalmente identici e pinout compatibili tra loro e con il **TTL 74LS181**; valgono pertanto tutte le informazioni offerte in precedenza per quest'ultimo integrato, a cominciare dal *pin-out* di *Figura 1*; in essa (e in tutte le altre figure relative a schemi) vanno naturalmente sostituiti i soli riferimenti ai piedini d'alimentazione, ora $\mathbf{V_{DD}}$ (pin 24) e $\mathbf{V_{SS}}$ (pin12).

Il dettaglio sulle 32 operazioni aritmetico-logiche è sempre sintetizzato dalle *Figure 3* e 7; le caratteristiche elettriche sono quelle tipiche della *famiglia logica* **CMOS**: la tensione di alimentazione $\mathbf{V_{DD}}$ può variare da **3V** a **15V**; i livelli di tensione d'uscita sono tipicamente uguali alla $\mathbf{V_{DD}}$ per la $\mathbf{V_{OH}}$ (a livello alto) e alla $\mathbf{V_{SS}}$ (=0V) per la $\mathbf{V_{OL}}$ (a livello basso); la *potenza dissipata* è trascurabile (qualche **µW**); il *ritardo di propagazione* massimo $\mathbf{t_{PLH}}$ e $\mathbf{t_{PHL}}$ (con carico di **200**kohm/**50**pF) aumenta al crescere del valore dell'alimentazione.

Indicativi sono, per esempio, quello tra ingressi di dato e uscite, che va da **800**ns ($\mathbf{V_{DD}}$=**5V**) a **240**ns ($\mathbf{V_{DD}}$=**15V**), e quello previsto nel passaggio da C_n a C_{n+4}, che va da **400**ns ($\mathbf{V_{DD}}$=**5V**) a **140**ns ($\mathbf{V_{DD}}$=**15V**).

Look-ahead Carry Generator (TTL): 74LS182

Nella descrizione degli operatori aritmetici abbiamo più volte sottolineato il problema legato alla gestione del *Riporto* (*Prestito*) nella somma (sottrazione); quella classica (***Ripple Carry***, intrinsecamente lenta) è stata abbandonata sia negli integrati sommatori (analizzati le puntate precedenti) che nelle ALU (oggetto di questa puntata), dotandoli internamente di speciali reti combinatorie in grado di generare in anticipo il valore del *Riporto* di ogni modulo FA, in funzione del solo valore corrente di tutti gli ingressi di dato.

Nel momento in cui questi dispositivi si sono dovuti mettere in cascata tra loro (per assicurare il servizio di numeri più grandi, multipli di 4 bit) il problema della velocità è diventato di fondamentale importanza, specialmente per le ALU, che richiedono prestazioni elevate.

Pur essendo ancora disponibili le uscite C_{n+4} (ciascuna da collegare all'ingresso C_n dello stadio successivo) l'uso della tecnica "*a propagazione del Riporto*" è reso insostenibile perché, per disporre del risultato, è necessario attendere che ciascun dispositivo produca il suo *Carry*, tipicamente 20ns per ognuno di essi.

La soluzione è quindi quella di "generare in anticipo" anche il valore del *Riporto* finale del gruppo di ALU (in cascata tra loro) coinvolte in operazioni aritmetiche, aggiungendo esternamente una rete combinatoria appositamente studiata, resa disponibile in forma integrata nel **74LS182**, detto per questo **Look-Ahead Carry Generator**; la *Figura 18* mostra il suo *pin-out*.

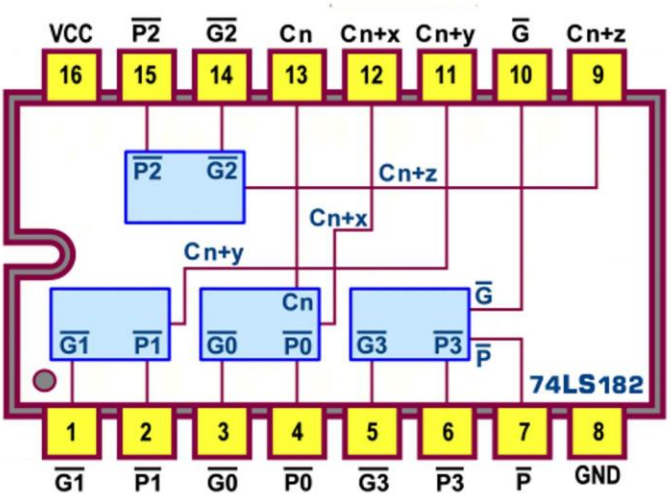

Figura 18 - Look-ahead Carry Generator 74LS182: Pin-out

Lo schema funzionale (*Figura 19*) riorganizza graficamente la sostanza dei fatti: questo dispositivo provvede alla "*previsione*" istantanea del *Riporto* (***Carry Look Ahead***) per ben 4 ALU; quello destinato alla seconda ALU, C_{n+x}, è ottenuto coinvolgendo, insieme a P_0 e G_0 della ALU meno significativa, anche l'eventuale segnale d'ingresso C_n in essa concorrente.

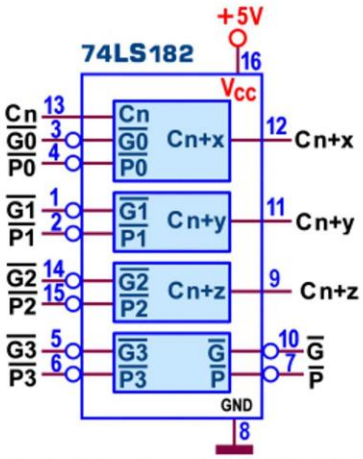

Figura 19 - Look-ahead Carry Generator 74LS182: Schema funzionale

Alle 3 reti rimanenti basteranno i soli *"Riporti di gruppo"* G e P della ALU ad esse relativa per generare l'uscita da collegare alla ALU successiva, C_{n+y} (per la terza) e C_{n+z} (per la quarta); le uscite dell'ultima ALU (quella più significativa) saranno invece ancora del tipo *"di gruppo"* per assicurare ulteriore espansione dei dispositivi in cascata.

In *Figura 20* si può vedere lo schema pratico, richiamato nella *Figura 21* per realizzare una ALU a 16 bit (aritmeticamente) veloce.

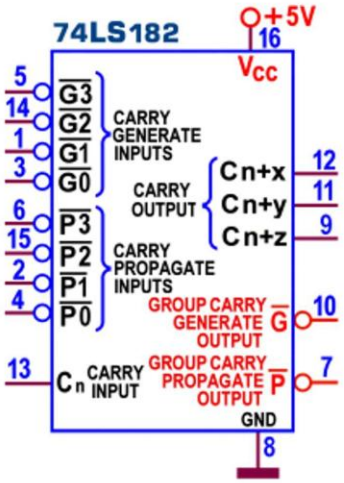

Figura 20 - Look-ahead Carry Generator 74LS182: Schema pratico

Molto istruttiva è la tabella di *Figura 22*, che mette a confronto i tempi necessari per disporre della somma di operandi da 1 a 64 bit, in funzione dei dispositivi coinvolti; si può notare che (per parole fino a 8 bit) la presenza di un generatore **74LS182** insieme a più ALU in cascata permette tempi di ritardo poco più grandi rispetto a quelli di 2 sole ALU gestite in ***Ripple Carry***.

Numero di bit delle Parole	Tempo Ritardo Tipico	INTEGRATI UTILIZZATI		METODO usato per il Carry tra le ALU in cascata
		ALU	Look Ahead Generator	
da 1 a 4	20 ns	1	Nessuno	Nessuno
da 5 a 8	20 ns	2	Nessuno	Ripple Carry
da 9 a 16	30 ns	3 o 4	1	Full Look-Ahead
da 17 a 64	50 ns	5 o 16	2 o 5	Full Look-Ahead

Figura 22 - Dispositivi ALU da 1 a 64 bit: Tabella dei Tempi

Quadruplicando lo schema di *Figura 21* (e portando le uscite G e P dei generatori di riporto veloce di ciascun gruppo in un quinto **74LS182**) si realizza una ALU a 64 bit che, pur coinvolgendo ben 16 ALU e 5 Look-Ahead Carry Generator, assicura la somma in soli 50 ns; lo schema utilizza le ALU **74LS181** ma va benissimo anche per la versione ridotta **74LS381**.

La massima potenza dissipata dal **74LS182** è di **80 mW**.

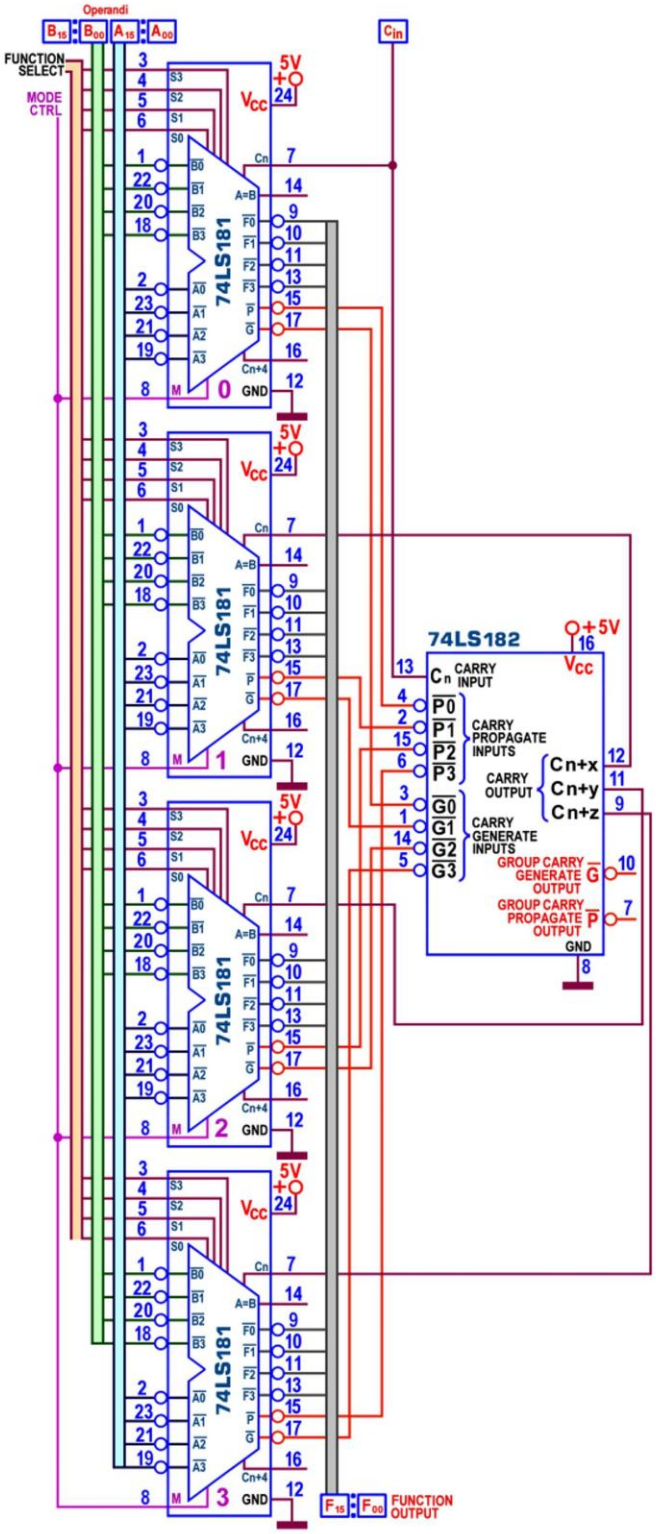

Figura 21 - Look-ahead Carry Generator 74LS182: ALU veloce a 16 bit

Look-ahead Carry Generator (CMOS): 4582, 40182

Anche per i componenti di tipo **Look-Ahead Carry Generator** sono disponibili 2 integrati della serie **CMOS**: il **4582** e il **40182**, funzionalmente identici e pinout compatibili tra loro e con quello **TTL** appena descritto, **74LS182**; tutte le precedenti spiegazioni e immagini (dalla *Figura 18*, *pin-out*, alla *Figura 21*, schema ALU a 16 bit) rimangono valide anche per questi componenti, ricordando che il positivo dell'alimentazione (pin 16) è ora $\mathbf{V_{DD}}$ e la massa (pin 8) è ora marcata $\mathbf{V_{SS}}$. Per altro, la diversa tecnologia assicura una *potenza dissipata* trascurabile ma aumenta notevolmente i *ritardi di propagazione*; per esempio, quello massimo tra ingressi e uscite di ciascun modulo, è valutabile da **400**ns ($\mathbf{V_{DD}}$=**5V**) a **150**ns ($\mathbf{V_{DD}}$=**15V**), con il solito carico di **200**kohm/**50**pF.