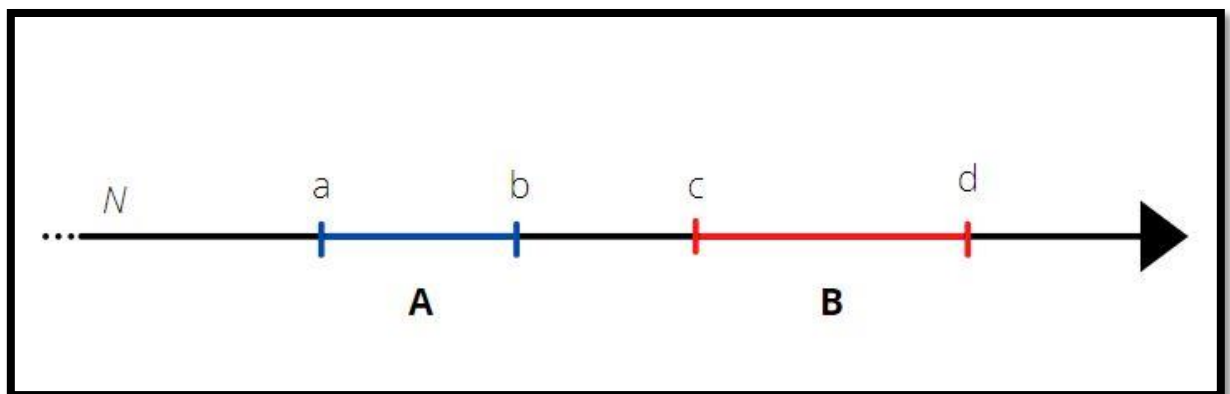


# Media di due intervalli

Progetto d'esame di Reti Logiche A.A. 2020/2021

(sessione invernale)



Studente: Alessio Muzi  
matricola n°: 299329



## SPECIFICA

### Scopo del progetto

Il progetto prevede la realizzazione di un componente attraverso il simulatore circuitale TKGate (versione 2.0).

Il circuito, prese in input 2 coppie di numeri interi a 4 bit, esegue inizialmente un confronto in ogni coppia stabilendo qual è l'estremo superiore e quale quello inferiore dell'intervallo, per poi eseguire la media aritmetica dei valori dei due intervalli.

### Specifica funzionale

Il progetto è strutturato per input di numeri interi a 4 bit.

Inoltre, per evitare intervalli nulli, ogni coppia deve essere composta da numeri diversi.

Utilizzando le regole della notazione posizionale per numeri interi senza segno:

$$2^4 - 1 = 16 - 1 = 15$$

potremmo quindi codificare valori interi da 0 a 15 in input.

In TKGate, per inserire i numeri nel circuito vengono utilizzate porte di switch, le quali accettano valori in codifica esadecimale (dove le lettere "a", "b", "c", "d", "e", "f" assumono rispettivamente i valori da 10 a 15).

Il modulo potrà eseguire 4 operazioni: confronto, sottrazione, addizione e divisione per due.

Il confronto sfrutta il fatto che i numeri sono rappresentati in base 2, eseguendo quindi al più 4 confronti, uno per ogni cifra, partendo dal bit più significativo. Nel caso siano essi entrambi 0 o entrambi 1, si prosegue con il confronto della cifra nella posizione n-1 e così via, fino ad arrivare alla prima cifra diversa. Per ogni confronto abbiamo quindi 3 casi:



$$\begin{array}{lll} E_0 = \overline{AB} + \overline{\overline{AB}} & = 1 & \text{se } A = B \\ C_0 = A\overline{B} & = 1 & \text{se } A > B \\ D_0 = \overline{A}B & = 1 & \text{se } A < B \end{array}$$

Poiché due numeri sono uguali se e solo se tutti i bit sono rispettivamente uguali e i numeri delle coppie accettati sono necessariamente diversi, non dobbiamo preoccuparci del caso in cui non ci sono differenze anche nel bit meno significativo dei due numeri. Inoltre, per non creare confusione, verrà considerato come “confronto”  $A > B$ , dove  $A$  è il primo numero e  $B$  il secondo, il quale darà come output 1 se  $A$  è maggiore di  $B$  o 0 se  $A$  è minore.

La sottrazione tra gli estremi degli intervalli viene eseguita tramite la somma del complemento a 2 del sottraendo, mentre l’addizione dei valori dei due intervalli viene eseguita tramite ripple-carry adder a 4 bit.

La divisione per due viene eseguita eliminando il bit meno significativo e eseguendo uno shift di tutte le cifre verso destra.

L’output finale, della grandezza di 6 bit, sarà visualizzato in forma decimale attraverso un LED decimale a 7 segmenti ed un led in caso di resto decimale 0,5. L’intero progetto è stato realizzato con l’utilizzo di porte logiche implementate in tecnologia C-MOS.

Si stima l’esecuzione dell’intero processo in 4 cicli di clock.

## Tecnologia C-MOS

La Tecnologia C-MOS (acronimo di *complementary metal-oxide semiconductor*) è un tipo di tecnologia utilizzata nell’ambito della elettronica digitale per la progettazione di circuiti integrati.

Il transistor è il componente base di ogni porta logica, ed è formato da 3 terminali: Source (S), Drain (D) e Gate (G).

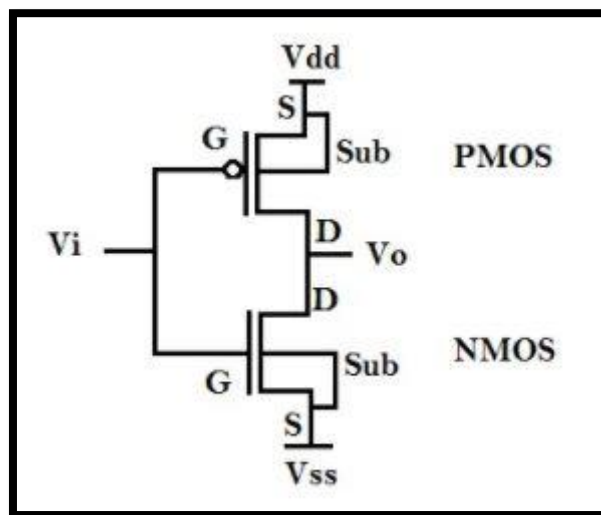
Questi 3 componenti rendono funzionante il Transistor nella stessa



maniera di un interruttore ( la corrente raggiunge da S a D passando per G solamente se G permette questo collegamento).

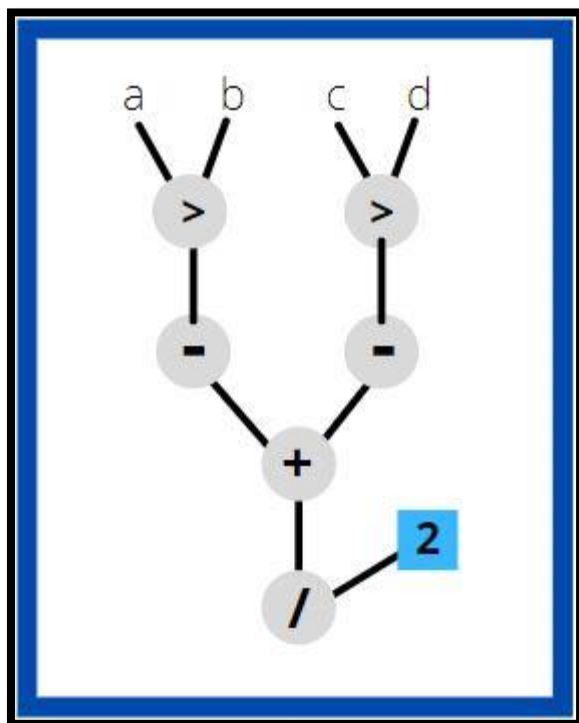
La Tecnologia C-MOS ha le basi in due componenti, il primo si chiama N-MOS, esso crea un collegamento tra gli estremi S e D solamente se G assume valore uguale a 1 detto pull-down.

La controparte duale si chiama invece P-MOS, essa crea un collegamento tra i due estremi solamente se G assume un valore uguale a 0 detto pull-up.



## IMPOSTAZIONE DEL PROGETTO A LIVELLO RT

### Data Flow Graph



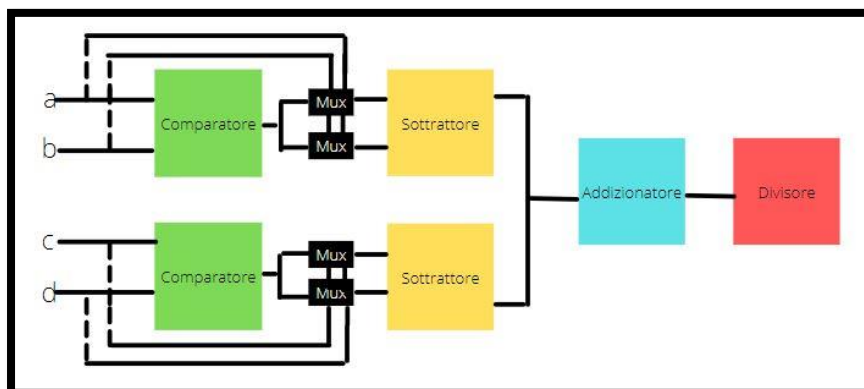
$$T_{clk} > \max \{ T_p(\text{Comp}), T_p(\text{Sott}), T_p(\text{Add}), T_p(\text{Div}) \}$$

$$\text{Latenza: } 2 * T_p(\text{Comp})$$

$$\text{Throughput: } 1/T_p(\text{Comp})$$

### Risorse

Complessivamente, il circuito richiede due comparatori a 4 bit, due sottrattori a 4 bit, un addizionatore a 5 bit ed un divisore a 6 bit. Inoltre serviranno 4 multiplexer a due ingressi e due porte NOT per il corretto instradamento dei numeri nei sottrattori.





## PROGETTO DELLE RISORSE A LIVELLO GATE

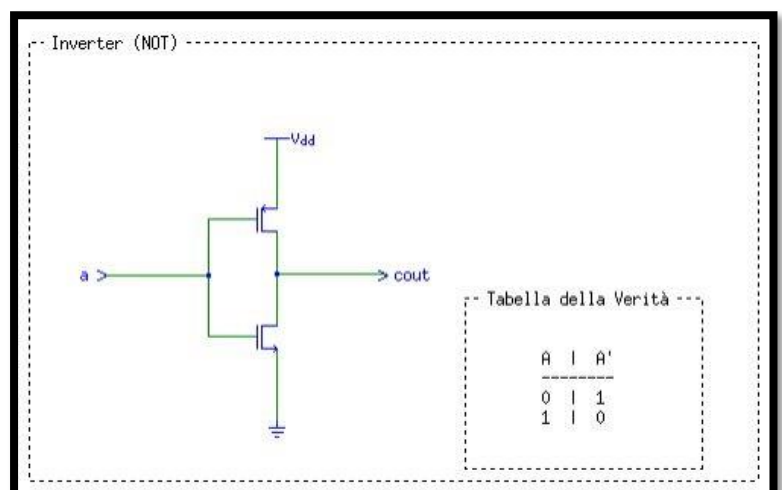
### Porte logiche elementari

Tramite la tecnologia C-MOS, è possibile riprodurre tutte le porte logiche ma, per motivi di stima di complessità circuitale e ordine progettuale, è possibile esprimere ogni funzione logica, grazie alle proprietà dell'algebra booleana, usando un numero ristretto di porte. La libreria, definita "funzionalmente completa", usata per questo progetto sarà composta da porte {NOT} , {NAND} e {NOR}.

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

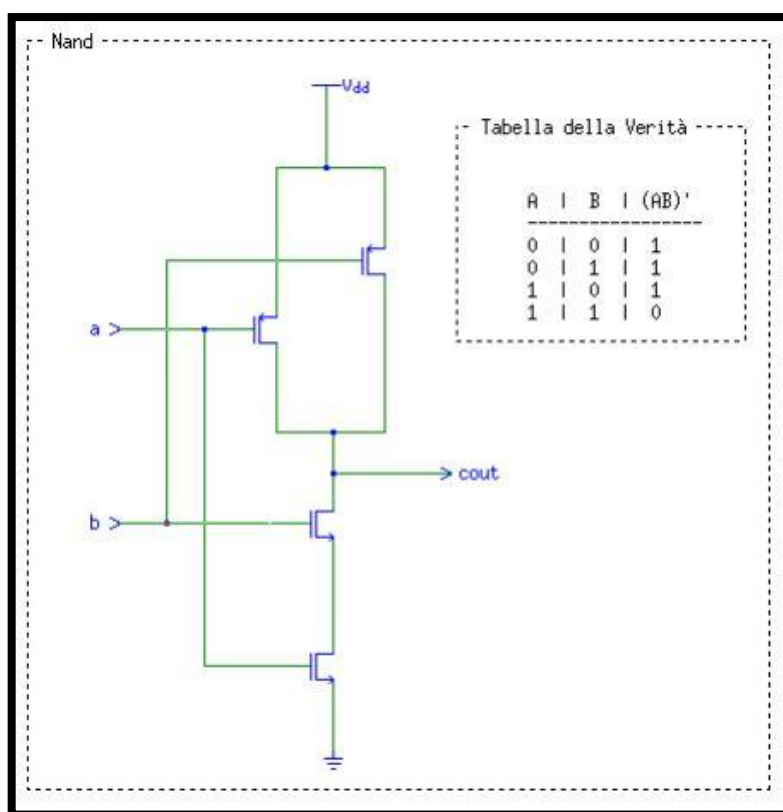
### NOT (Inverter)

La porta logica NOT (conosciuta anche come inverter) prende in input un singolo bit e come risultato restituisce lo stesso bit in forma negata.



## NAND

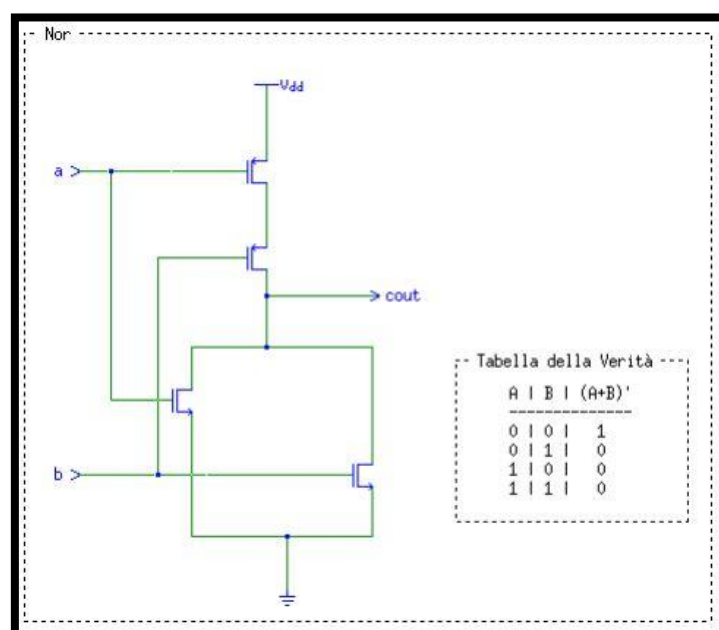
La porta logica NAND riceve in entrata due input e restituisce 1 solo se almeno uno dei due input è 0, mentre nel caso in cui entrambi gli input siano 1, l'output sarà 0. Essa è composta da due P-MOS collegati in parallelo e due N-MOS collegati in serie.



## NOR

La porta logica NOR riceve in input 2 segnali, se riceve almeno un 1 il suo output sarà 0, altrimenti sarà 1.

In modo duale alla porta NAND, la porta NOR è composta da due P-MOS in serie e due N-MOS in parallelo.

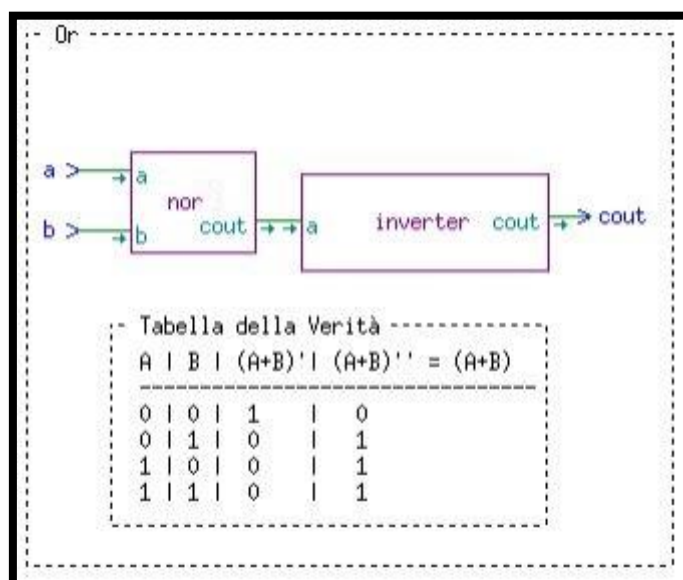


Tramite la combinazione di queste tre porte logiche, è possibile descrivere tutte le altre porte, tra le quali {OR}, {AND} e {XOR}. Inoltre, tramite la combinazione di porte OR ed AND a due ingressi, sono state realizzate porte a 3,4 e 5 ingressi per il comparatore.



## OR

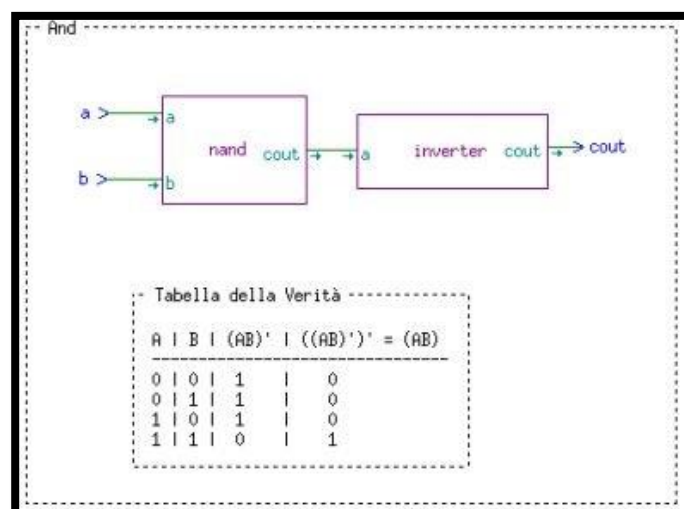
La porta logica OR implementata nel progetto è una combinazione della porta NOR concatenata ad un NOT. La porta OR ha due ingressi ed il suo output è collegato all'input del inverter, ottenendo il risultato negato rispetto al NOR visto in precedenza. Avremo un segnale 0 solamente quando l'input di entrambi gli ingressi è 0, altrimenti avremo 1.



## AND

L'AND utilizzato nel progetto è stato implementato analogamente come l'OR, ovvero con l'utilizzo della sua porta duale NAND e negata con l'inverter (NOT).

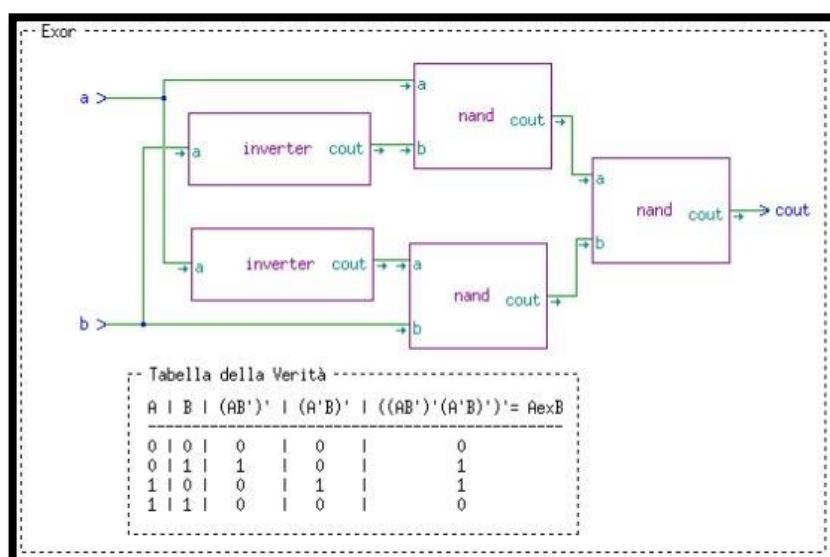
La funzione logica dell' AND (AB) è stata realizzata con la funzione equivalente  $((AB)')$ .



## XOR

La porta XOR è realizzata con 3 porte NAND e 2 inverter concatenati.

La funzione logica dell'XOR è  $(A \oplus B)$  equivalente alla funzione  $((AB)'(A'B))'$ .





## Macro Aritmetiche Comparatore

Il comparatore è stato realizzato tramite 4 moduli di comparazione ad un bit. Ognuno dei 4 moduli esegue un confronto della cifra corrispondente e assegnerà il valore 1 solo ad una delle 3 uscite, la quale rappresenta quale delle due cifre è maggiore dell'altra o se sono uguali. Combinando i 4 confronti, e ricordando che nel caso in cui due cifre sono uguali, bisogna portarsi dietro un altro valore che rappresenta la parità dei bit in questione, si ottiene questa espressione logica:

$$C = A_3\overline{B_3} + E_3A_2\overline{B_2} + E_3E_2A_1\overline{B_1} + E_3E_2E_1A_0\overline{B_0}$$

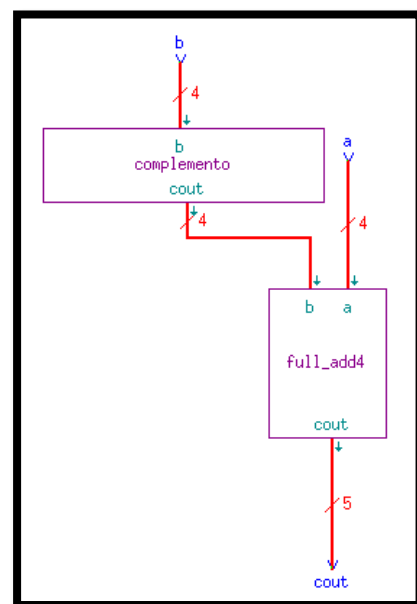
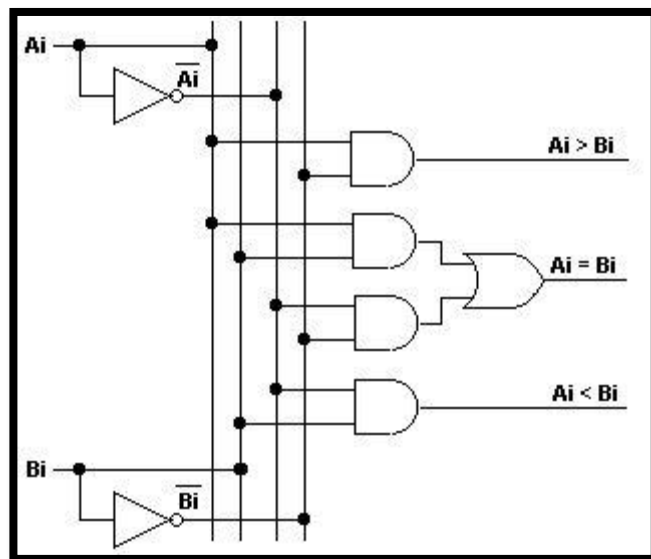
Dove se  $C = 1$ , A è maggiore di B, altrimenti è minore. Questo numero C diventa poi il bit di selezione per i multiplexer impiegati nella sottrazione.

## Sottrattore

Per eseguire la sottrazione, viene eseguito il complemento a 2 del sottraendo, ovvero vengono invertiti tutti gli 0 in 1 e gli 1 in 0, e viene aggiunto 1 al numero ottenuto.

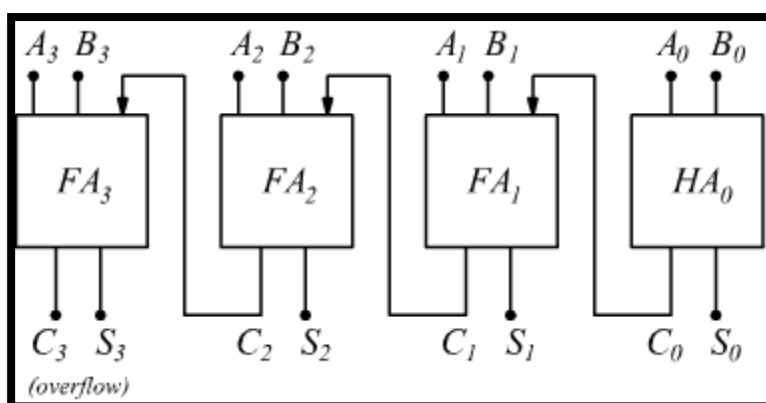
Eseguendo questa procedura, viene essenzialmente sommato al primo numero l'opposto del secondo.

$$A + (-B) = S$$



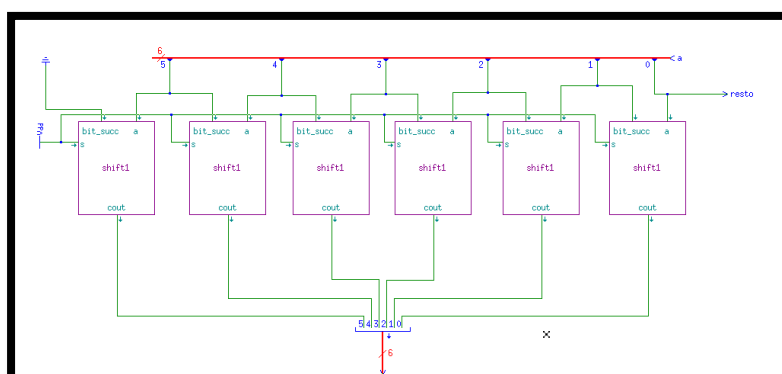
## Addizionatore

Per la realizzazione degli addizionatori, a 4 e 5 bit, sono stati utilizzati Full Adder e Half Adder in tecnologia Ripple Carry. La differenza tra un half adder e un full adder è che l'half adder non tiene conto del riporto, quindi la somma della prima cifra utilizza questo adder, mentre per la somma delle altre cifre è necessario tenere conto di un eventuale riporto, poiché avvengono a cascata. L'output della somma sarà di  $n+1$  bit, quindi in questo caso 5/6 bit.



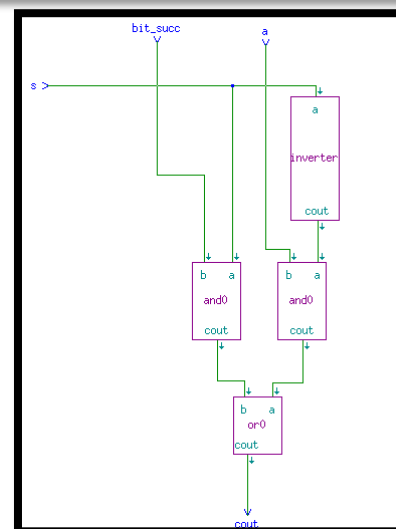
## Divisore

L'ultima operazione rimasta da eseguire è la divisione per due della somma dei due intervalli. Poiché ci troviamo in un sistema base 2, la divisione per due si riduce ad eliminare l'ultima cifra, cioè ad effettuare uno shift di una posizione verso destra per tutte le cifre. Se la cifra eliminata risulta essere un 1, allora in quel caso avremo un resto decimale di 0,5 ; segnalata da un led acceso.



## Multiplexer

Per la corretta selezione di minuendo e sottraendo successiva al confronto, sono stati utilizzati due coppie di multiplexer a due ingressi che lavorano in modo duale. Nel caso il risultato del confronto sia 1, il primo mux restituisce A (il primo numero) e il secondo mux restituisce B (il secondo), mentre se il risultato del confronto è 0, avviene l'opposto.





## Simulazione e analisi del progetto

### Metriche per l'analisi di blocchi funzionali

**Area (A):** Somma del numero di ingressi di tutte le porte logiche (nella tecnologia C-MOS ad ogni ingresso di un gate corrispondono 2 transistor, cioè i due dispositivi elementari di cui sono composte tutte le porte logiche).

**Tempo di propagazione ( $T_p$ ):** intervallo di tempo che intercorre tra l'applicazione di una nuova configurazione agli ingressi di un blocco e la stabilizzazione dei corrispondenti valori dei segnali d'uscita.

**Tempo di contaminazione ( $T_c$ ):** intervallo di tempo che intercorre tra l'applicazione di una configurazione agli ingressi di un blocco e la stabilizzazione dei corrispondenti valori dei segnali d'uscita.

Il tempo di contaminazione totale è il minore tra i tempi di contaminazione tra i pin d'ingresso e i pin d'uscita, mentre il tempo di propagazione è il maggiore tra i tempi di propagazione.  
Entrambi i tempi sono calcolati secondo il modello di ritardo unitario.

**Throughput (Rate):** numero di risultati prodotti nell'unità di tempo.

**Stima prestazionale delle porte logiche:**

Porta Logica	TP	TC	AREA
NOT	1	1	1
NAND	1	1	1
NOR	1	1	1
AND	2	2	2
OR	2	2	2
XOR	3	2	5



## Stima prestazionale delle macro aritmetiche:

### Comparatore:

- TP: Il caso pessimo è il caso in cui tutte le cifre siano uguali tranne l'ultima, la quale percorre il percorso più lungo ( $TP = 3$ ). Inoltre, il segnale attraverserà una porta NOT, 3 AND e 2 OR.

$$TP = 3 + 1 + 3 * 2 + 2 * 2 = 14. \{O(n)\}$$

- TC: Il percorso più breve è quello eseguito dal segnale che identifica che la cifra più significativa del primo numero è maggiore del secondo, attraversando un ulteriore AND e 2 OR.

$$TC = 3 + 2 + 2 * 2 = 9. \{O(1)\}$$

- Area: Un comparatore a 4 bit è composto da 4 comparatori ad un bit (Area = 12), 4 NOT, 10 AND e 3 OR.

$$Area = 4 * 12 + 4 + 10 * 2 + 3 * 2 = 78. \{O(n)\}$$

- Rate =  $\{O(1/n)\}$

### Sottrattore:

- TP: Il tempo di propagazione deriva dalla somma del tempo di propagazione del complemento a 2 ( $TP = 3$ ) e quello del addizionatore ( $TP = 32$ ).

$$TP = 3 + 32 = 35. \{O(n)\}$$

- TC: Il percorso più breve è quello eseguito dall'ultimo full adder.

$$TC = 2. \{O(1)\}$$



- Area: Il sottrattore è composto da un modulo per il complemento a 2 e da un addizionatore a 4 bit.

$$\text{Area} = 64 + 4 * 5 = 74. \{O(n)\}$$

- Rate =  $\{O(1/n)\}$

### Addizionatore:

Nel progetto sono presenti due tipi di addizionatore, uno a 4 bit, usato nel sottrattore e composto interamente da full adder e ed uno a 5. Il primo calcolo si riferisce al addizionatore a 4 bit, il secondo a quello a 5 bit.

- TP: Il caso pessimo è il caso in cui il riporto si propaga a partire dall'half adder (TP = 3) e in tutti i full adder (TP = 8).

$$TP = 4 * 8 = 32. \mid TP = 3 + 4 * 8 = 35. \{O(n)\}$$

- TC: Il percorso più breve corrisponde al caso in cui entrambe le cifre che entrano nell half adder (o full adder) siano 0 e viene attraversata solo una porta AND.

$$TC = 2. \{O(1)\}$$

- Area: L'area circuitale è la somma dell'area circuitale di un half adder (Area = 7) sommata all'area di 3 full adder (Area = 16).

$$\text{Area} = 4 * 16 = 64. \mid \text{Area} = 7 + 4 * 16 = 55. \{O(n)\}$$

- Rate =  $\{O(1/n)\}$



## Divisore:

- TP: Il tempo di propagazione dell'intero divisore corrisponde al tempo di propagazione della sorgente  $s$  di un singolo bit shifter (TP = 5).

$$TP = 5. \{O(1)\}$$

- TC: In modo analogo, il tempo di contaminazione corrisponde al tempo di contaminazione del bit shifter (TC = 4).

$$TC = 4. \{O(1)\}$$

- Area: Il divisore è composto da 6 bit shifter (Area = 7).

$$Area = 6 * 7 = 42. \{O(n)\}$$

- Rate =  $\{O(1/n)\}$

## Multiplexer:

- TP: Il segnale che attraversa il percorso più lungo è il segnale negato del bit di selezione, che attraversa un NOT, un AND ed un OR.

$$TP = 1 + 2 + 2 = 5.$$

- TC: Il segnale che attraversa il percorso più breve attraversa solamente un AND ed un OR.

$$TC = 2 + 2 = 4.$$

- Area: L'area del multiplexer con ingressi a 4 bit corrisponde a 4 volte l'area di un multiplexer con ingresso ad 1 bit (Area = 7).

$$Area = 4 * 7 = 28.$$



## Totale:

- TP: Il tempo di propagazione del circuito è la somma dei tempi di propagazione di tutti i componenti.

$$TP = 14 + 5 + 35 + 35 + 5 = 94.$$

- TC: Il tempo di contaminazione del circuito è la somma dei tempi di contaminazione di tutti i componenti, escluso quello del comparatore.

$$TC = 2 + 2 + 4 = 8.$$

- Area: Il circuito totale è composto da 2 comparatori, 2 porte NOT, 4 multiplexer, 2 sottrattori, 1 addizionatore e 1 divisore.

$$Area = 2 * 78 + 2 + 4 * 28 + 2 * 74 + 1 * 55 + 1 * 42 = 459.$$

Segue illustrazione del modulo *main* del circuito completo e di una simulazione.

