Introduzione alle esercitazioni di laboratorio

Progettazione digitale assistita da calcolatore

Ambiente di lavoro: Altera Max + Plus II ®

Stefano Mattoccia
email: smattoccia@deis.unibo.it

Università di Bologna
DEIS - Dipartimento di Elettronica Informatica e Sistemistica

Bologna, 21 Luglio 2004

Introduzione

Max + Plus II è un ambiente software per la progettazione e la simulazione di circuiti logici.
La più importante caratteristica dell'ambiente è quella di consentire al progettista di gestire la complessità attraverso l'approccio gerarchico alla progettazione.
La metodologia di progetto prevista si basa sulla suddivisione del problema, e quindi del progetto, in più livelli di blocchi interconnessi sempre più semplici.
Il sistema di progettazione consente di simulare il comportamento di ogni blocco al fine di verificarne e il corretto funzionamento.
Il procedimento è interattivo e si può effettuare su una comune piattaforma PC.
La progettazione interattiva e assistita dal calcolatore è molto più rapida e affidabile della progettazione verificata direttamente sulla realizzazione hardware.

L'ambiente di lavoro Max + Plus II consente di trasferire il progetto su circuiti integrati programmabili detti Field Programmable Gate Array (FPGA).
Il trasferimento del progetto al chip viene effettuato utilizzando una scheda di programmazione che si collega al PC attraverso la porta parallela.
Si consulti in proposito il sito www.altera.com e/o i manuali di Max + Plus II per maggiori dettagli sui dispositivi hardware utilizzabili con Max + Plus II e le modalità della loro programmazione.

Il costruttore Altera consente agli *studenti* di utilizzare il software di progettazione, previa richiesta di autorizzazione individuale da inoltrare direttamente alla casa madre.

Per essere autorizzati a utilizzare MAX+Plus II **è necessario** richiedere al produttore del software (Altera) una **chiave software personale e riservata agli studenti.** La procedura di installazione e autorizzazione è descritta in seguito.

Download e installazione di MAX+PLUS® II

- CdRom
- Sito Web di Altera: www.altera.com

La documentazione è inclusa nel pacchetto.

Come ottenere la Chiave Software per abilitare la licenza studenti

Per ottenere la chiave software per l'abilitazione di Max + Plus II (10.xx) sul proprio PC in *versione studenti* è necessario eseguire le seguenti operazioni:

1) Da una shell del DOS del PC sul quale si desidera installare Max + Plus II digitare

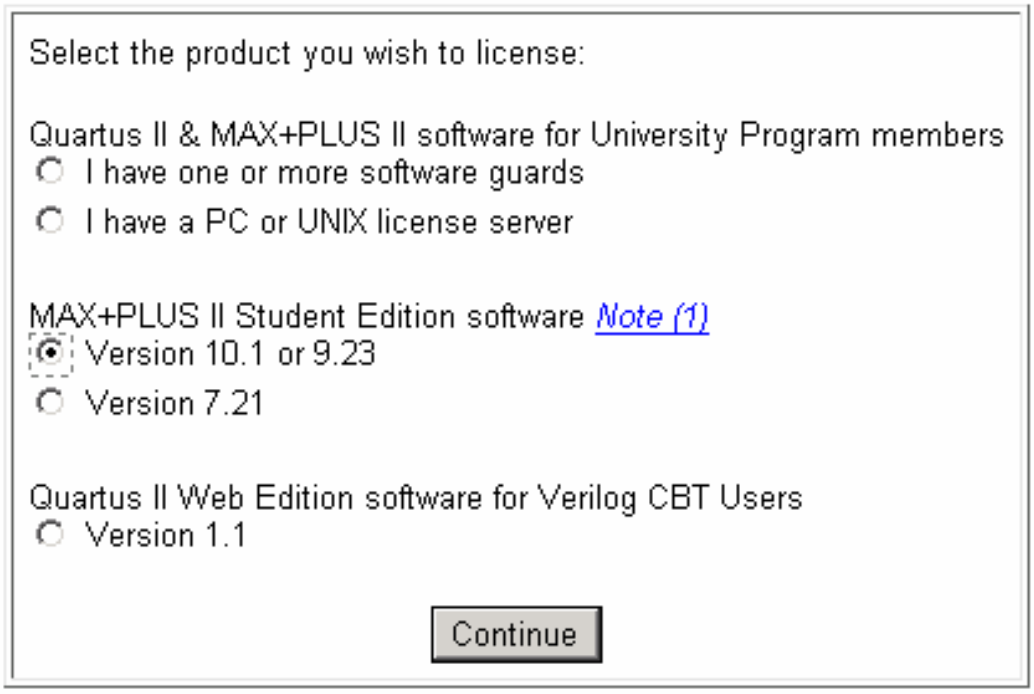
   dir /p

Questo comando consente di ottenere il numero di serie dell'hard-disk del proprio PC: (Numero di serie del volume: XXXX-XXXX). Generalmente il numero di serie è una stringa di 8 cifre.

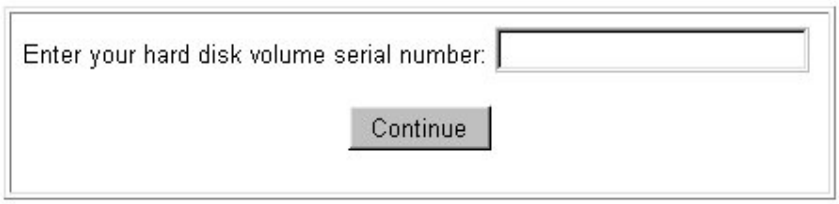
2) Collegarsi al sito di Altera al seguente indirizzo:
2

http://www.altera.com/support/licensing/lic-university.html

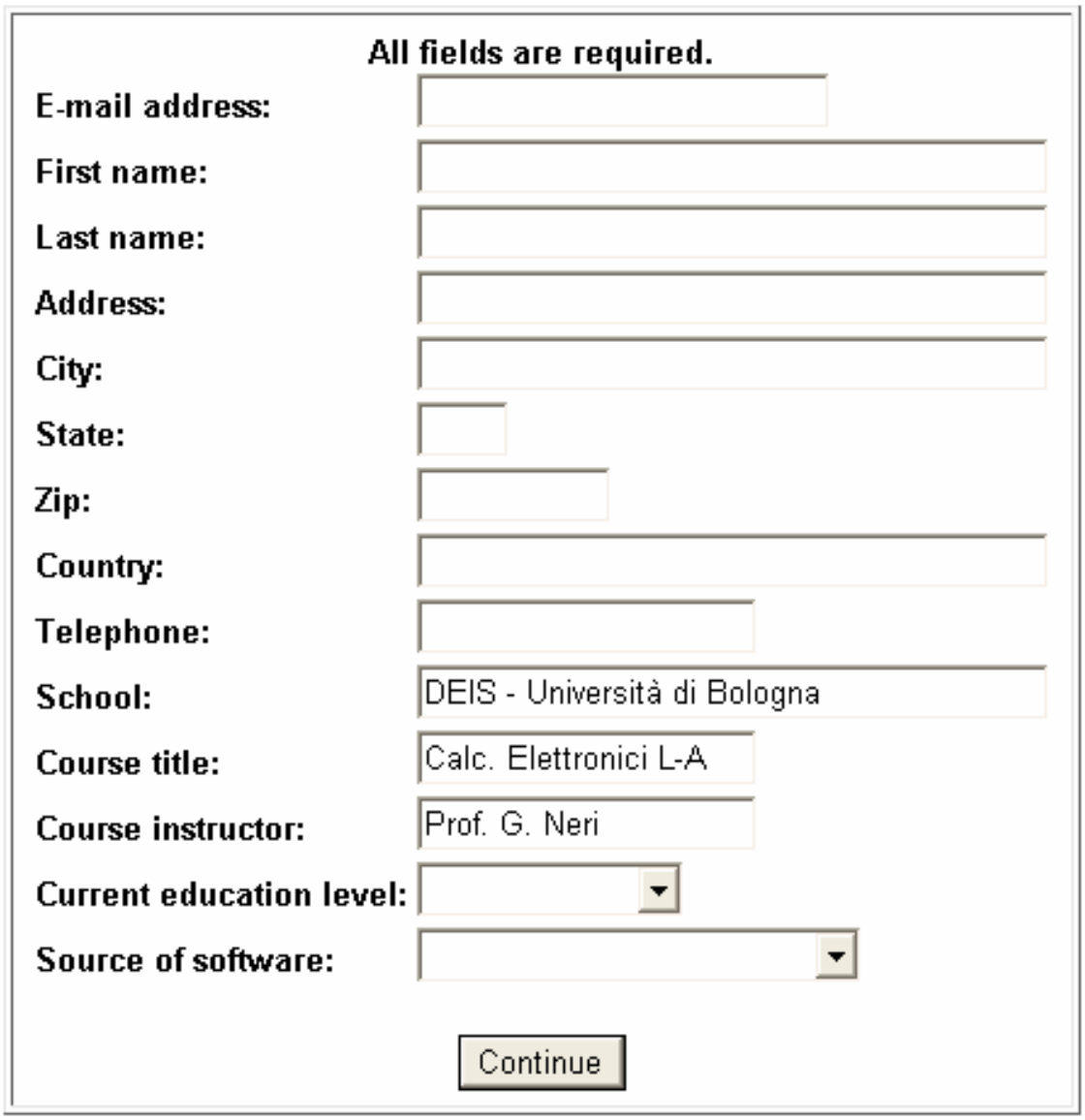
Selezionare la versione di MAX+PLUS ® II Student Edition 10.1 or 9.23 come indicato nella figura seguente



Procedendo è necessario inserire il numero di serie del proprio hard-disk, rilevato seguendo le istruzioni del punto 1)



Procedendo ulteriormente viene richiesto di compilare un *form* con i propri dati personali. La figura seguente mostra un esempio parzialmente compilato con le informazioni comuni a tutti. Inserite i vostri dati nei campi lasciati vuoti e cliccate su **continue**.



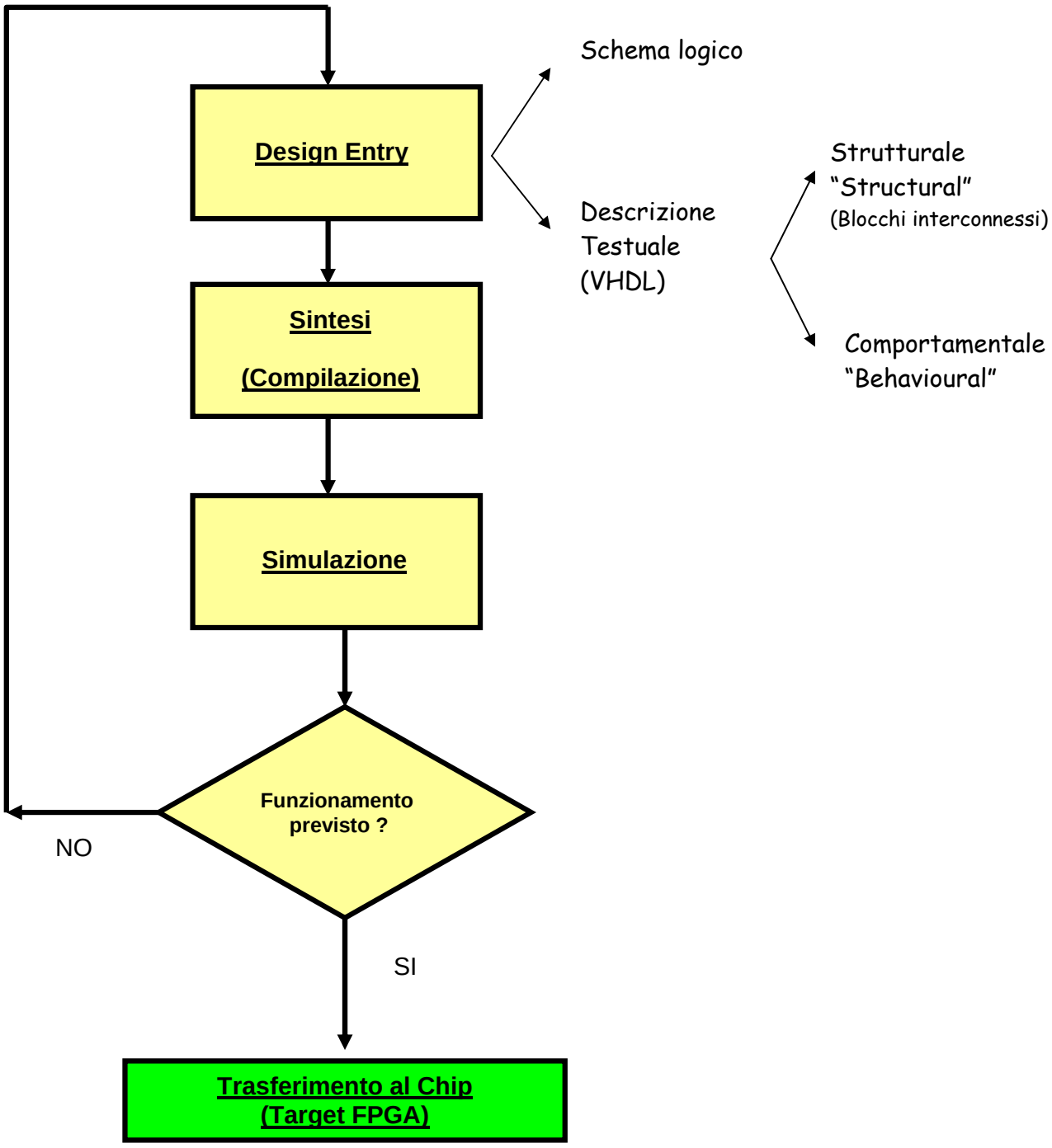
NOTA: E' importante inserire correttamente il proprio indirizzo di *e-mail* al fine di poter ricevere la licenza di attivazione del software.

3) Attendere il codice di attivazione **'license.dat'** della propria *licenza studente* registrata. Questo file verrà spedito, generalmente dopo pochi minuti, all'indirizzo di *e-mail* indicato nel *form* compilato al passo 2).

4) Installare Altera Max + Plus II eseguendo il programma di *setup* fornito insieme al pacchetto software.

5) Eseguire Altera Max + Plus II. Dal menu ***Options*** selezionare ***License Setup….*** Indicare nella casella di testo '*License File or Server Name'* il percorso del file 'license.dat' ricevuto via *e-mail*.

Metodologia di progetto



Nel corso, Max + Plus II verrà prevalentemente utilizzato per consentire una maggior comprensione degli argomenti trattati attraverso l'utilizzo della simulazione.

Vedremo ora con un esempio come progettare in modo gerarchico, scomponendo cioè il problema in sottoproblemi sempre più semplici, un sommatore a 4 bit. Successivamente verrà mostrato come inserire lo schema logico del progetto all'interno di Max + Plus II e come sia possibile simularne il funzionamento.

Progettazione gerarchica di un sommatore a 4 Bit

Problema: *progettare in modo gerarchico una rete che esegua la somma di due numeri codificati con codice binario da 4 bit.*

Soluzione:

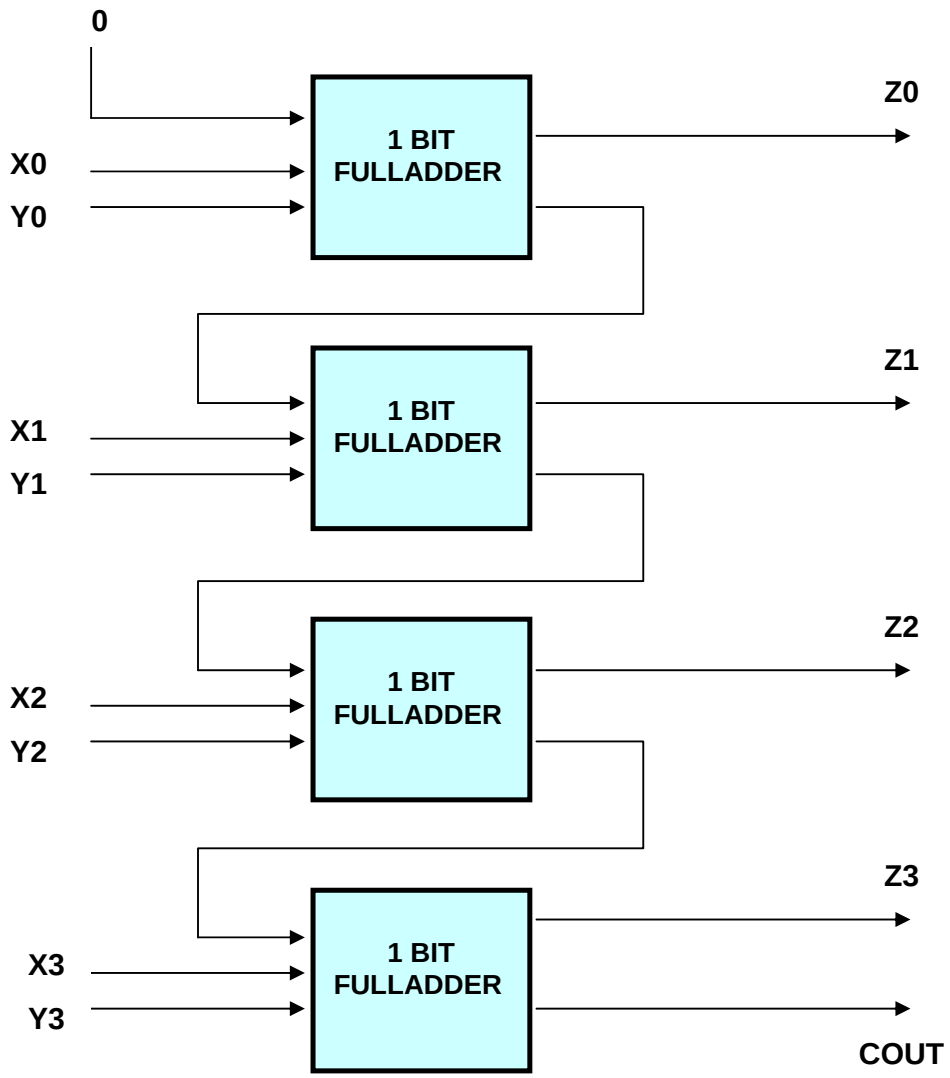
Il primo passo è quello di descrivere lo schema in termini di segnali di ingresso e uscita ("*descrizione ai morsetti*"). Tale livello di descrizione corrisponde al livello gerarchico 0 (o '*entity*').

Level 0 - 'Entity'

A partire dal livello gerarchico 0 si scompone il problema in sottoproblemi sempre più semplici come mostrato negli schemi logico seguenti, corrispondenti ai diversi livelli gerarchici 1,2,..

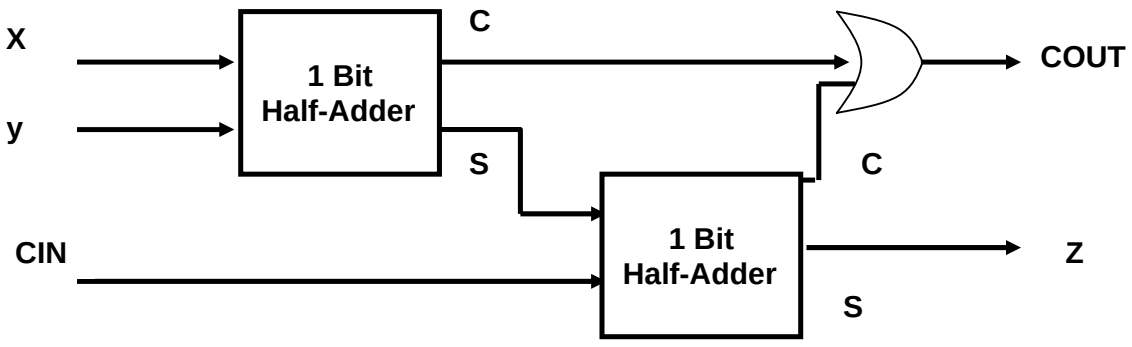
Level 1 – 4 Bit Adder

Lo schema logico di livello 0 può' essere ulteriormento scomposto in blocchi più semplici, come mostrato nello schema logico seguente, utilizzando 4 Full-Adder a 1 Bit.



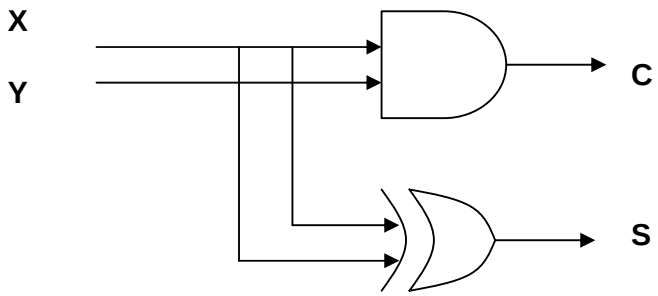
Level 2 – 1 Bit Full Adder

I Full-Adder ad 1 Bit sono descrivibili mediante il seguente schema logico utilizzando degli Half-Adder a 1 Bit.



Level 3 – Half Adder

Infine, gli Half-Adder utilizzati per realizzare i Full-Adder a 1 Bit possono essere descritti dal seguente schema logico. Questo è livello di descrizione più basso del nostro progetto.



Design Entry

Dopo avere progettato il sommatore a 4 Bit con l'approccio *top-down* descritto è possibile introdurre gli schemi logici corrispondenti ai vari livelli gerarchici all'interno di MAX + Plus II per poterne simulare il funzionamento ed eventualmente mapparlo all'interno di un dispositivo fisico. La fase di inserimento del progetto all'interno di MAX + Plus II avviene con un approccio *Bottom-Up* partendo cioè dai livelli gerarchici più bassi fino ad arrivare al livello gerarchico 0.
Una volta che l'intero progetto sarà inserito all'interno del simulatore sarà possibile esaminarlo e simularlo sfruttandone la descrizione di livello 0 ('entity') corrispondente al massimo livello di astrazione. Se necessario si potrà scendere di livello ed analizzare il comportamento dei vari blocchi che compongono l'interno progetto fino a scendere ai livelli di descrizione più bassi (e dettagliati).

Ogni blocco che compone un progetto può' essere descritto all'interno di MAX + Plus II seguendo due approcci differenti **1)** e **2)**:

1) **Schema Logico**
   Si inserisce lo schema logico del circuito utilizzando simboli grafici (porte logiche, flip-flop, etc) e si connettono graficamente i segnali secondo la logica del progetto utilizzando il mouse.

2) **Descrizione Testuale**
   Questo approccio sfrutta un linguaggio di programmazione ad alto livello denominato VHDL (**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage). E' possibile mediante questo linguaggio specificare il funzionamento di un blocco logico attraverso una descrizione testuale del suo funzionamento.

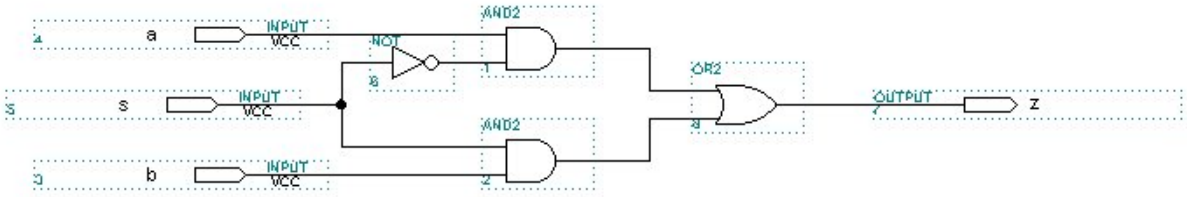
Nel corso verrà utilizzato come metodo di Design Entry quello basato sull'inserimento dello schema logico. Il linguaggio VHDL verrà studiato in corsi successivi. Per chi fosse interessato sono disponibili *tutorials online* presso i seguenti siti web:

http://www.vhdl-online.de/~vhdl/

http://rassp.scra.org/vhdl/courses/courses.html

Prima di continuare con l'esercizio sul sommatore vediamo brevemente con un esempio le differenze tra i due approcci progettando un multiplexer a 2 vie.

Multiplexer a 2 Vie descritto mediante schema logico



Schema logico di un Multiplexer a 2 Vie

Multiplexer a 2 Vie descritto mediante descrizione testuale

```
-- Un semplice Multiplexer a 2 Vie

LIBRARY ieee;
USE ieee.std_logic_1164.all;
-

ENTITY Mux2Vie IS

        PORT (
                        a               : IN  STD_LOGIC;        -- ingresso a
                        b               : IN  STD_LOGIC;        -- ingresso b
                        s               : IN  STD_LOGIC;        -- ingresso di Selezione

                        z               : OUT STD_LOGIC         -- Output
        );

END Mux2Vie;


ARCHITECTURE miarchitettura OF Mux2Vie IS
BEGIN

-- Il funzionamento del circuito è il seguente:
--      se 's' è zero  allora z=a
--      altrimenti (se 's' è uno) z=b

--  La descrizione in VHDL è la seguente

                z <= a WHEN ( s = '0')
                        ELSE b;

END miarchitettura;
```

Simulazione con MAX II Plus

Partendo dal progetto del '*sommatore a 4 bit*' descritto precedentemente l'inserimento dello schema logico all'interno di MAX + Plus II avverrà in modo gerarchico, partendo dal livello più basso, seguendo le fasi descritte in seguito.

Half-Adder a 1 bit (Level 3)

Design Entry dello schema logico.
Compilazione del progetto.
Simulazione del funzionamento.
Creazione del simbolo di libreria corrispondente all'Half Adder a 1 bit.

Full Adder a 1 bit (Level 2)

Design Entry dello schema logico sfruttando il simbolo di libreria dell'Half-Adder a 1 bit.
Compilazione del progetto.
Simulazione del funzionamento.
Creazione del simbolo di libreria corrispondente al Full-Adder a 1 bit.

Full-Adder a 4 bit (Level 1)

Design Entry dello schema logico sfruttando il simbolo di libreria del Full-Adder a 1 bit.
Compilazione del progetto.
Simulazione del funzionamento.
Creazione del simbolo di libreria corrispondente al Full-Adder a 4 bit.

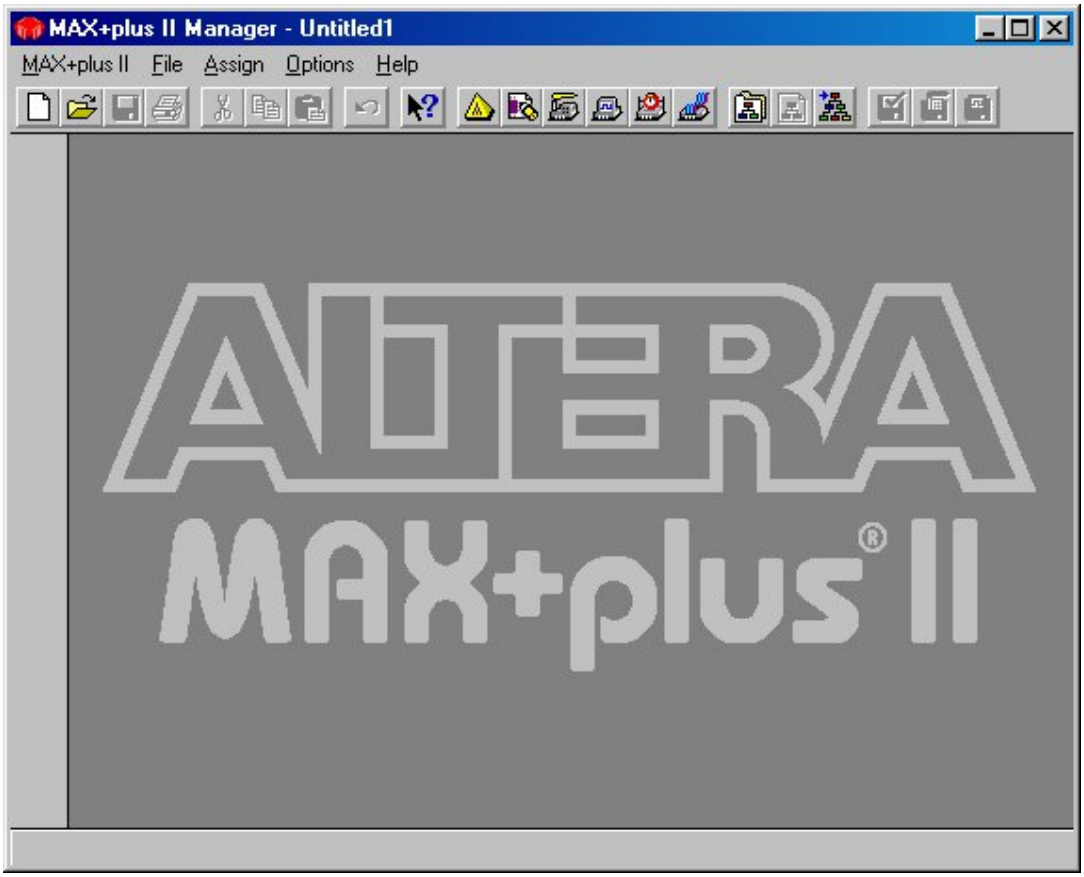
Sommatore a 4 bit (Level 0)

Design Entry dello schema logico sfruttando il simbolo di libreria del Full-Adder a 4 bit.
Compilazione del progetto.
Simulazione del funzionamento.

Half-Adder a 1 bit (Level 3)

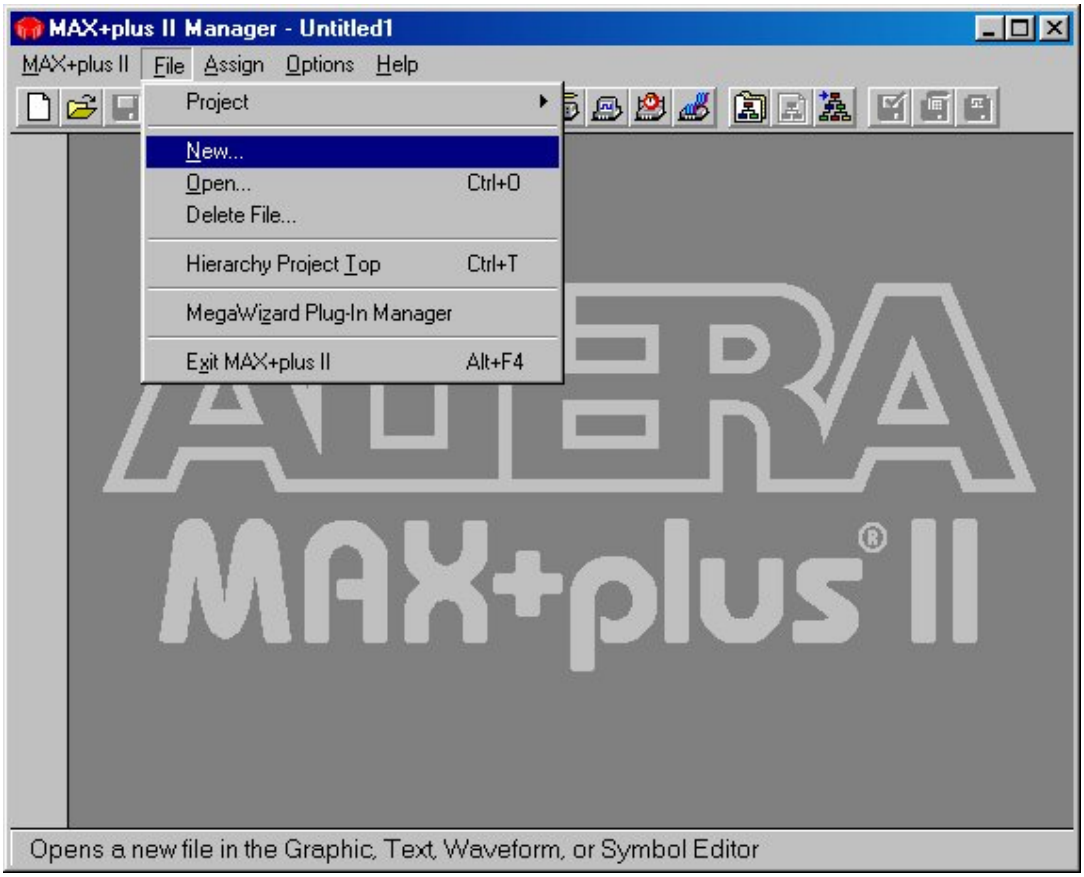
Vedremo ora in dettaglio come progettare attraverso il ***Graphic Editor*** lo schema logico del circuito Half-Adder a 1 bit. Successivamente verrà mostrato come compilare il progetto e simularlo attraverso l'utilizzo del ***Waveform Editor***. Infine, si vedrà come generare un **simbolo di libreria** per il circuito Half-Adder progettato.

Design Entry



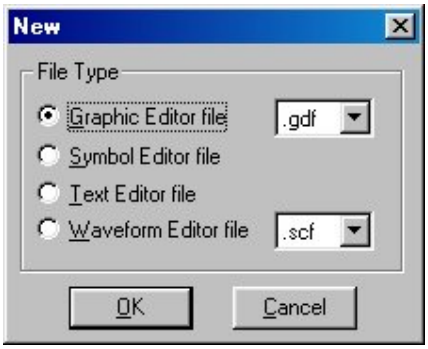
- Figura 1 – Ambiente di lavoro

Per creare un nuovo progetto, selezionare **New** dal Menu **File,** come mostrato in Figura 2.



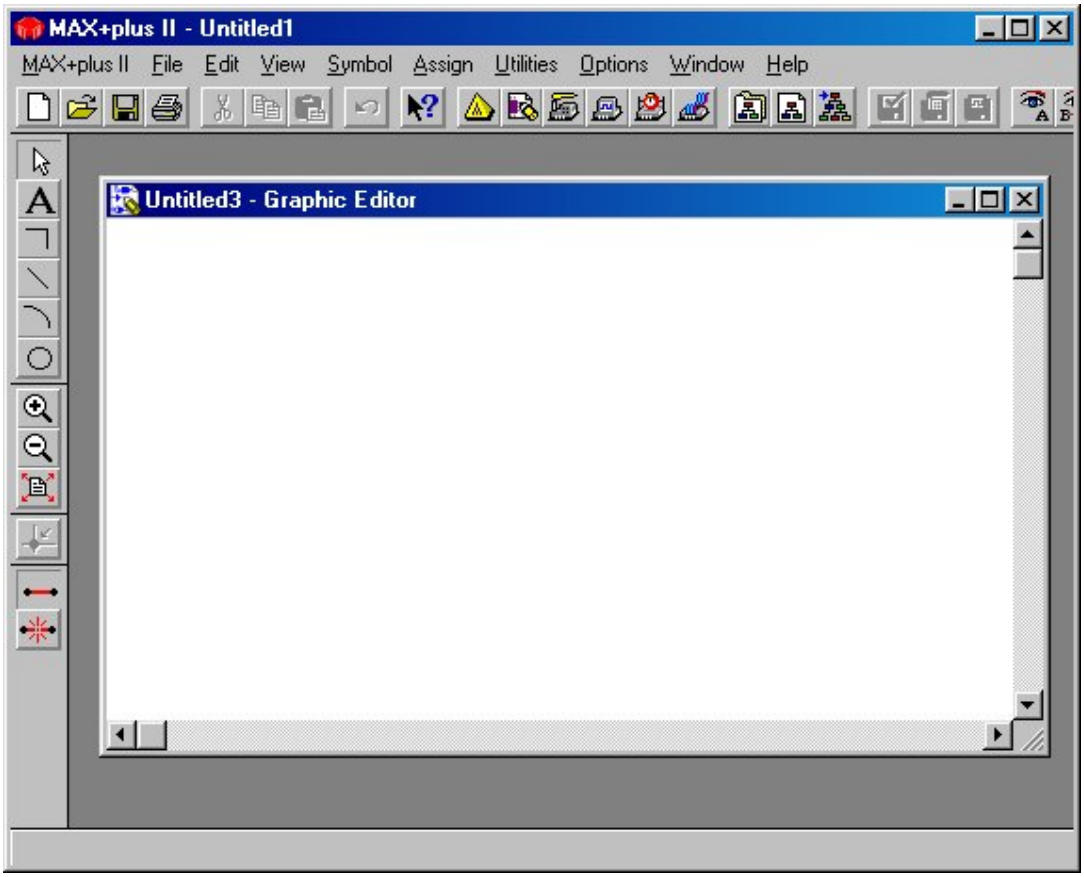
➢ Figura 2 – Nuovo progetto

Scegliendo **New** si ottiene la seguente finestra di dialogo:



➢ Figura 3 – Creazione di un nuovo file

Selezionando Graphic Editor File si entra nella modalità di Design Entry, nella quale sarà possibile inserire graficamente i simboli necessari per la progettazione. La finestra a sfondo bianco che appare rappresenta l'area di lavoro (Figura 4).



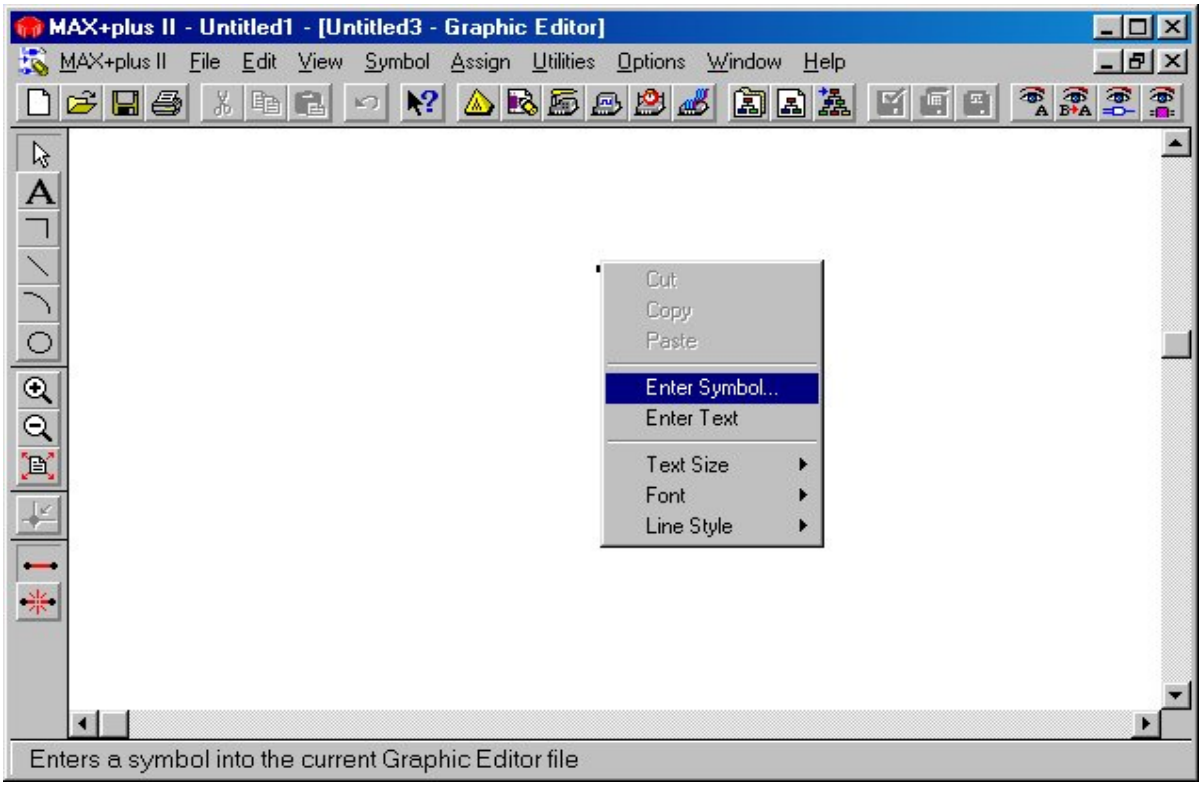
➢ Figura 4 – Area di lavoro

Vediamo come inserire graficamente l'AND e lo XOR necessari per progettare *un half-adder*.
Con il **TASTO DESTRO del MOUSE** si *clicca* entro l'area di lavoro. Appare la finestra di dialogo di Figura 5. Questa *dialog-box* consente di selezionare, da alcune librerie di simboli predefinite, la porta logica che ci interessa se questa è definita all'interno delle librerie stesse. Se è noto il nome del simbolo che si vuole inserire è possibile digitarlo nella casella *Symbol Name* (es per inserire un AND a 2 ingressi digitare "*and2*").
Nel caso in cui il nome del simbolo non sia noto è possibile cercarlo entro le "*symbol libraries*" . Nella configurazione corrente le symbol libraries definite in Max + Plus II sono:

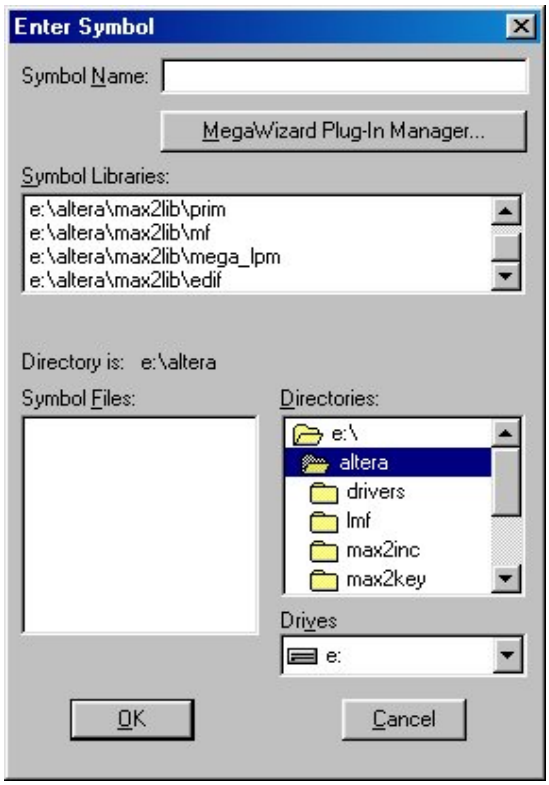
….\prim → Componenti 'primitivi'
….\mf → Componenti 'complessi'
….\mega_lpm → Componenti 'molto complessi'
….\edif → Componenti 'complessi'

Se invece il simbolo logico è stato creato dal progettista, allora deve ovviamente essere cercato nel direttorio in cui è stato salvato. Si raccomanda allo studente di non salvare mai un componente da lui creato in uno dei quattro direttori suindicati.

NOTA: prima di iniziare un nuovo progetto è bene creare una nuova directory in cui memorizzare i componenti creati dal progettista.

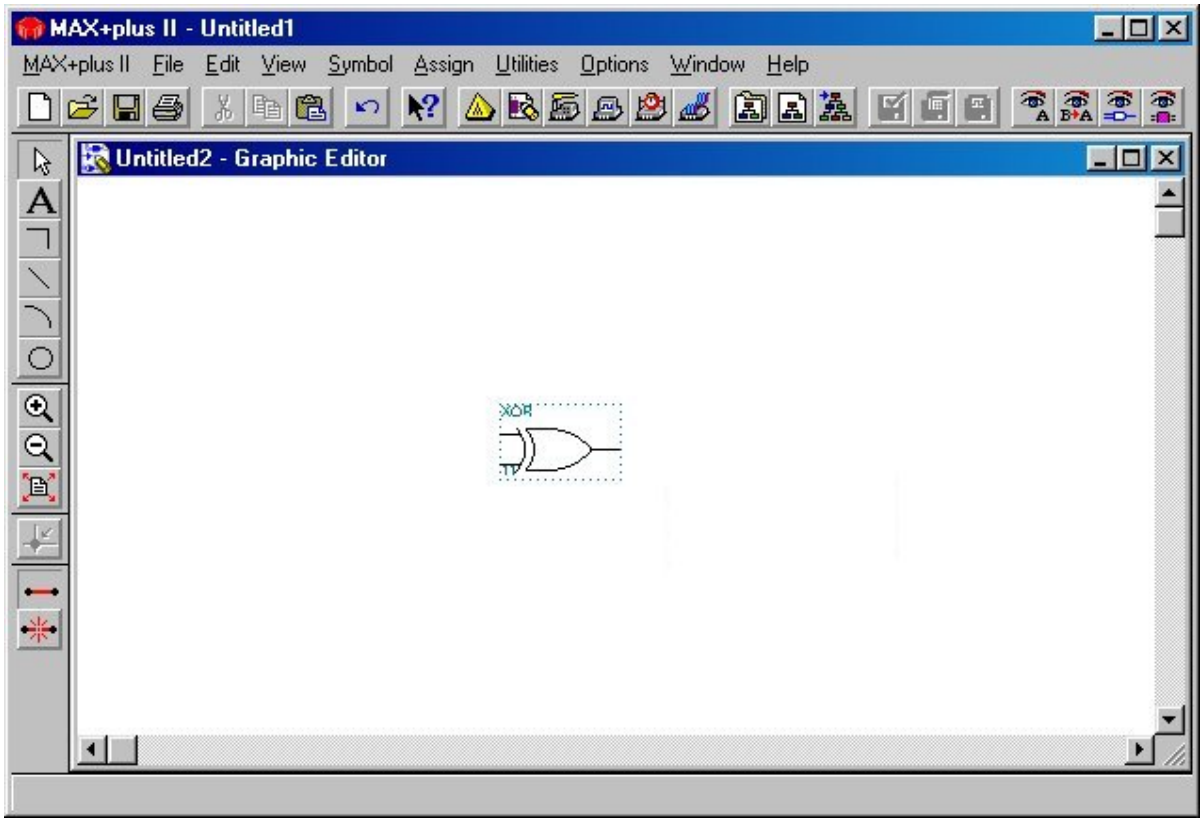


➢ Figura 5 – Inserimento di un simbolo grafico (1)

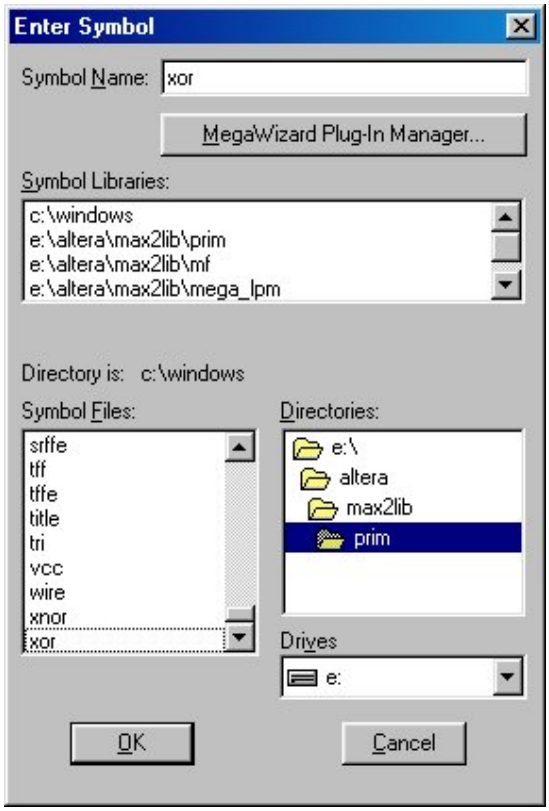


➢ Figura 6 - Inserimento di un simbolo grafico (2)

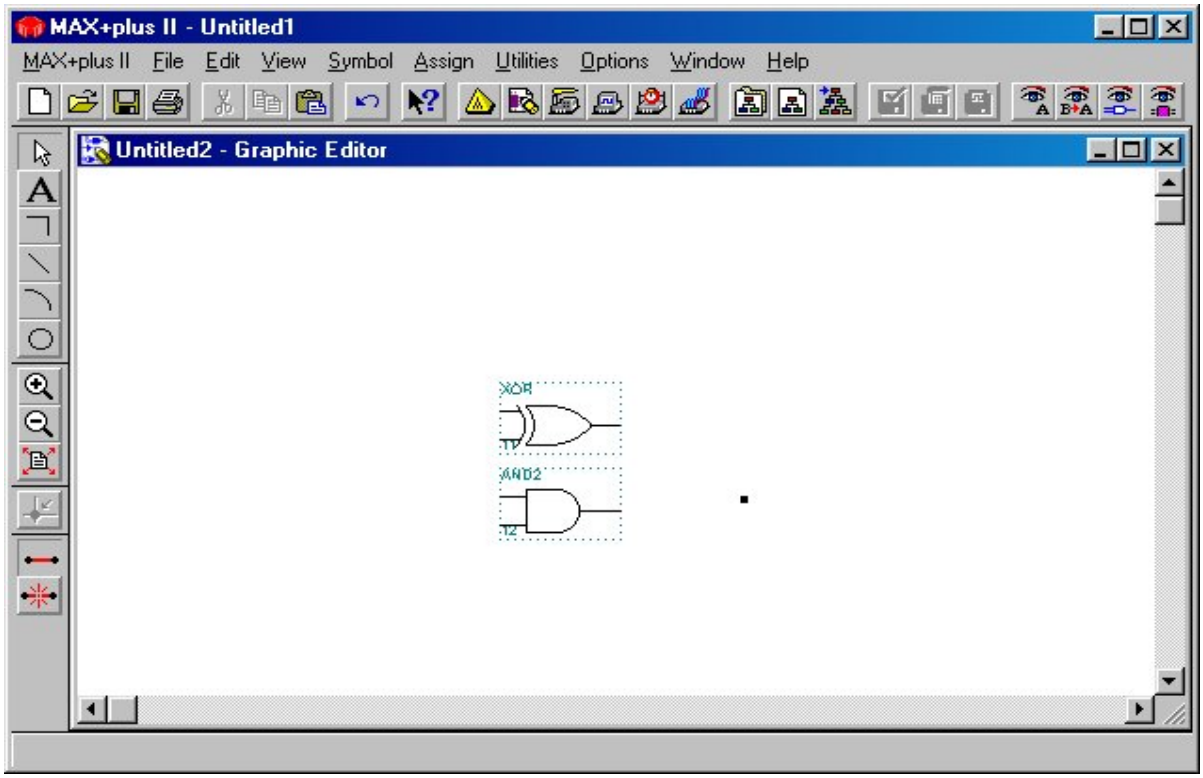
Dalla dialog box entriamo nella directory "…\prim" e selezionamo il simbolo logico dello XOR (Figura 7-8) e dell'AND a due ingressi (and2). Il foglio di lavoro che si ottiene è quello di Figura 9.



- Figura 7 - Inserimento di un simbolo grafico (3)



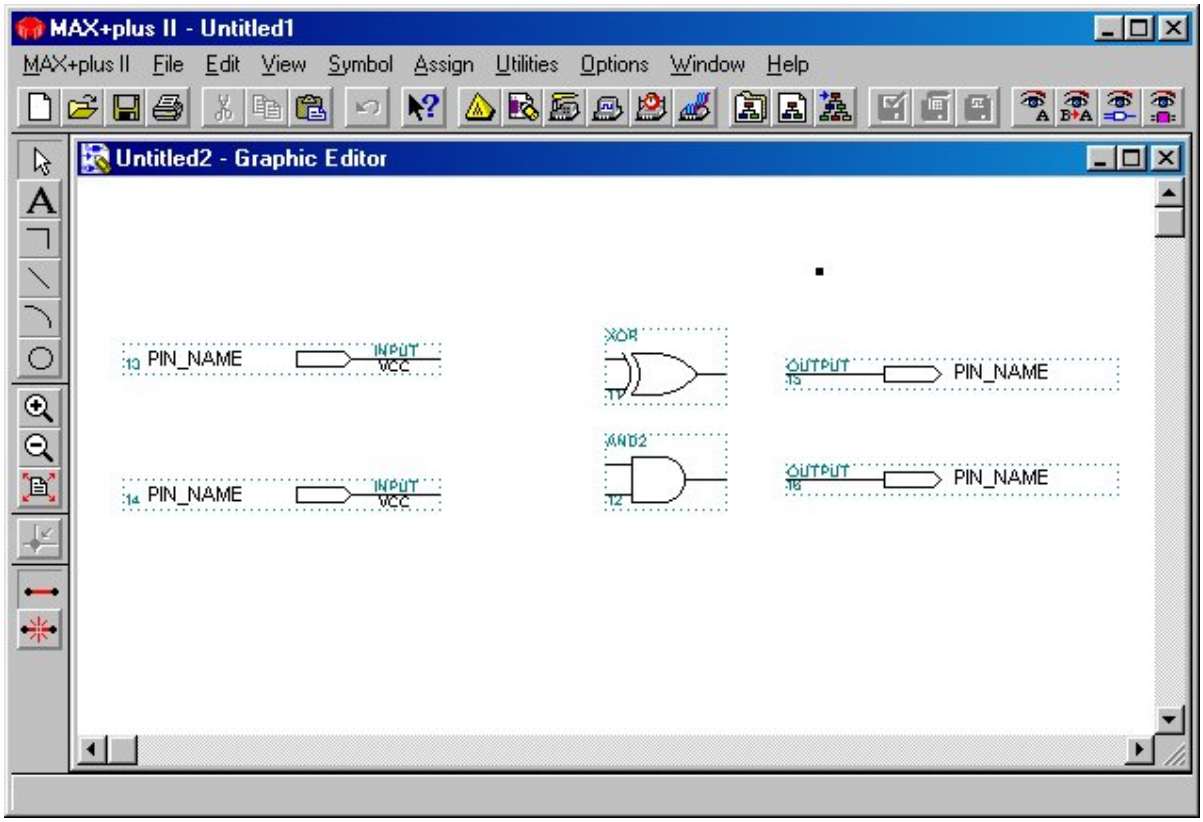
➢ Figura 8 - Inserimento di un simbolo grafico (4)



➢ Figura 9 - Inserimento di un simbolo grafico (5)

Il circuito Half-Adder ha due ingressi **A** e **B** e due uscite **C** e **S.** Gli ingressi e le uscite devono essere specificati in Max + Plus II in modo distinto. Questo avviene inserendo tali simboli attraverso la casella **Symbol Name** selezionando rispettivamente "**input**" e "**output**" per ogni ingresso ed ogni uscita. Alternativamente gli ingressi e le uscite possono essere inseriti attraverso le *symbol libraries* selezionando rispettivamente "**input**" e "**output**".

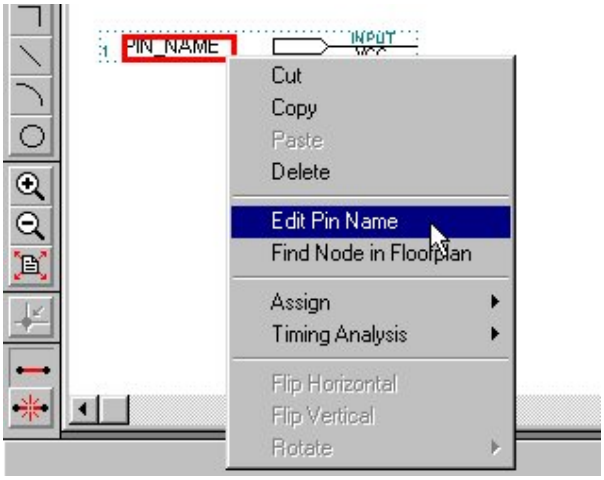
La figura seguente mostra il foglio di lavoro con tutti i componenti necessari per la progettazione dello schema logico corrispondente all'*Half-Adder*.



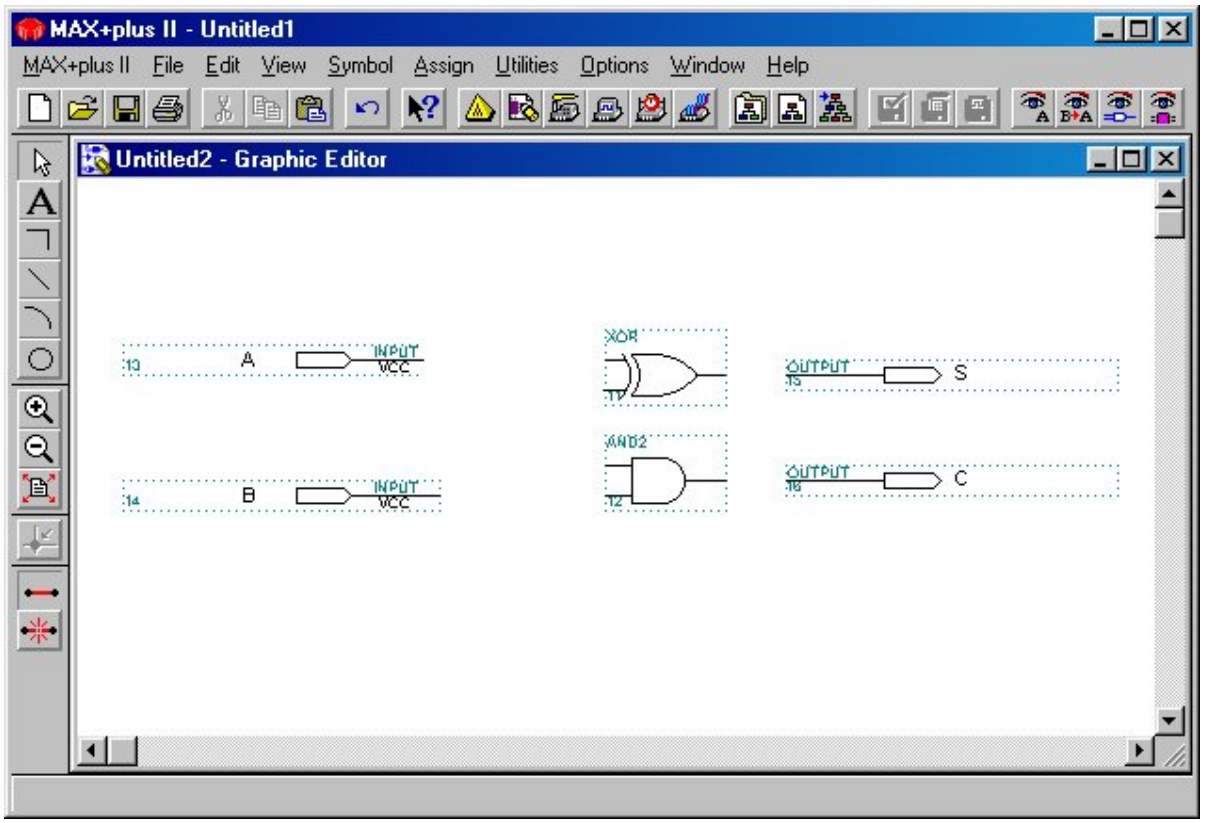
➢ Figura 10 a – Simboli logici necessari per progettare un Half-Adder

Prima di procedere alla connessione dei vari segnali è necessario rinominare ogni segnale di ingresso e uscita con il corretto simbolo logico. Questo si otiene *cliccando* con il **TASTO DESTRO del MOUSE** sul "PIN_NAME" , selezionando la voce **Edit Pin Name** del simbolo di ingresso o uscita, ed inserendo il nome appropriato per ogni segnale (Figura 11).

NOTA: Max + Plus II NON è *case sensitive* per cui Y e y sono considerati uguali.

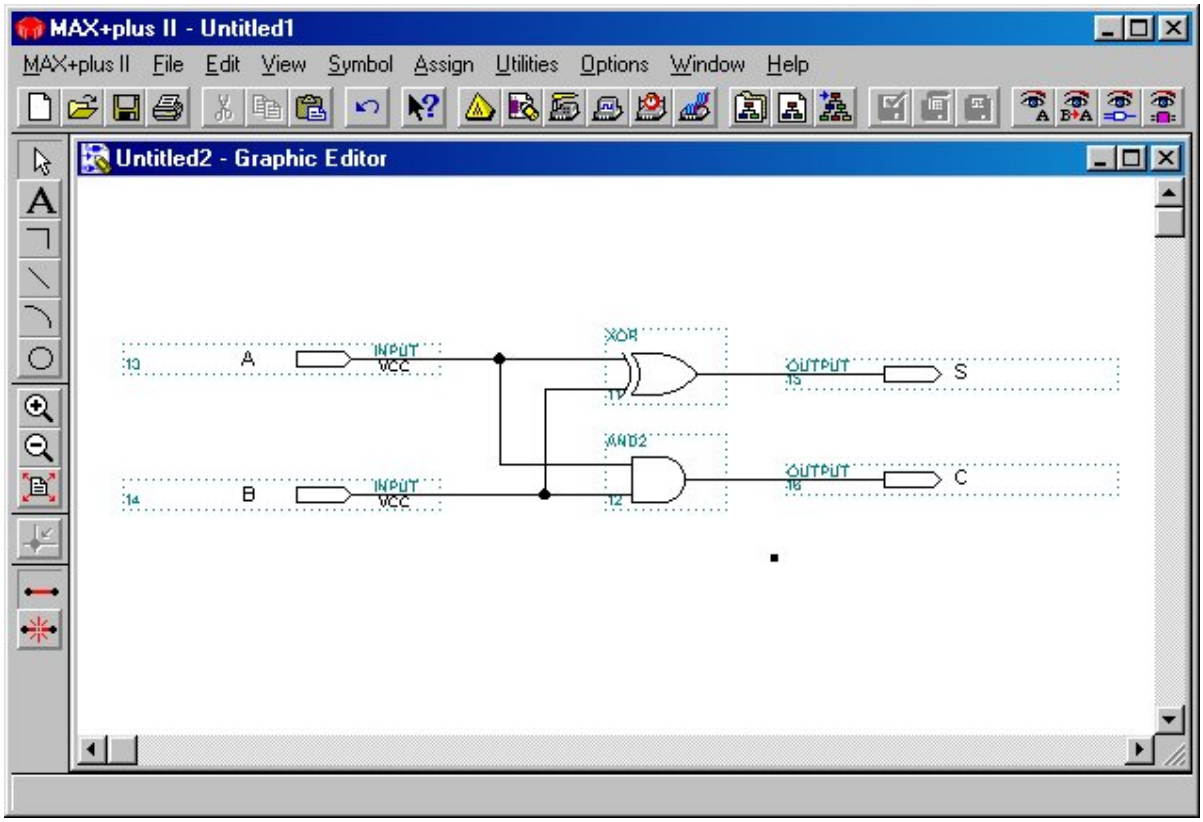


➢ Figura 10-b – Associazione del componente di input al segnale di ingresso



➢ Figura 11 – Risultato delle assegnazioni

A questo punto è possibile procedere alla connessione dei vari simboli logici con i rispettivi segnale. Questo avviene tenendo premuto il **TASTO SINISTRO del MOUSE** e spostandosi tra il punto di partenza e il punto di arrivo del segnale. Il risultato di tale operazione è mostrato in Figura 12.



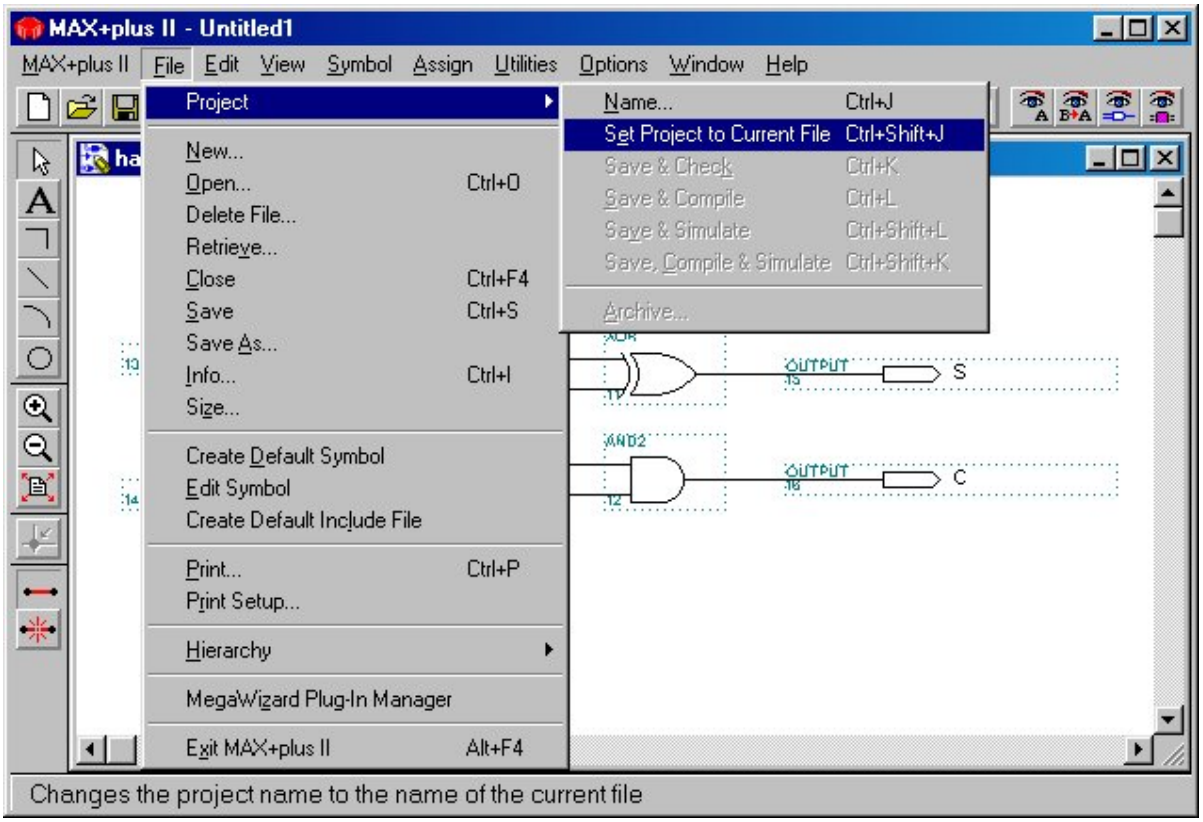
➢ Figura 12 – Half-Adder

A questo punto è necessario salvare il progetto creato attraverso l'utilizzo del *Graphic Editor*. Dal menu "**File"** selezionare "**Save as…."** e salvare il file con il nome desiderato (es "HalfAdder.gdf").

Sintesi (Compilazione)

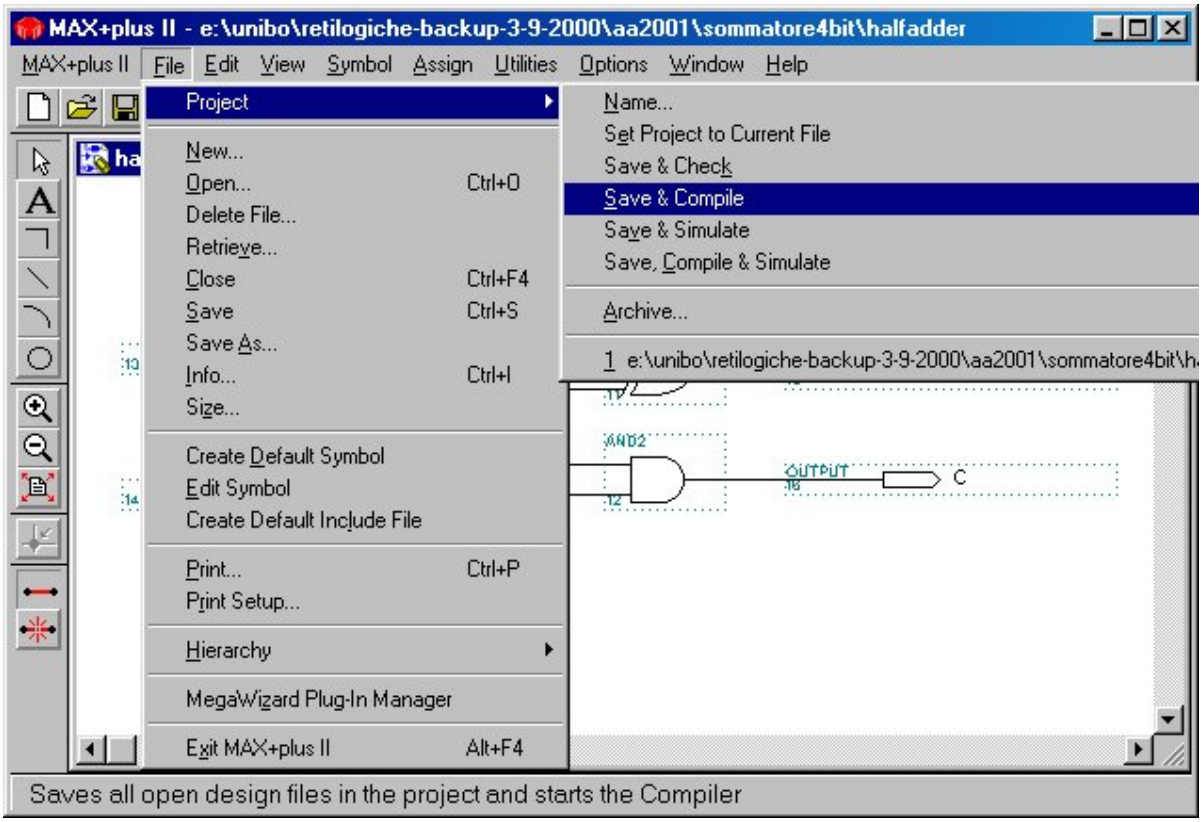
Una volta terminata la fase di *Design Entry* è necessario procedere alla trasformazione degli schemi logici in files interpretabili dal calcolatore per poterne simulare il funzionamento. Tale processo di conversione viene chiamato '**compilazione**' ed è un processo concettualmente analogo a quello utilizzato nei linguaggio di programmazione ad alto livello (ad esempio il C). Il compilatore si occupa di trasformare gli schemi logici, espressione di un linguaggio 'orientato' al progettista in istruzioni comprensibili dal calcolatore.
Poiché l'obiettivo del processo che stiamo descrivendo è la mappatura del progetto su un dispositivo fisico (FPGA), prima di procedere alla compilazione è necessario che il sistema sappia qual è questo dispositivo fisico (detto **target**). Questa scelta può essere effettuata dal menu **Assign->Device**. Se questa operazione non viene effettuata, allora Max + Plus II utilizzerà come target un dispositivo di default.
In fase di compilazione verrà segnalato se il dispositivo fisico selezionato è in grado di soddisfare le specifiche di progetto. Se questo non fosse verificato sarebbe necessario selezionare un dispositivo di classe superiore (attraverso **Assign->Device**).

Per procedere con la compilazione è necessario *settare* il file corrente come *progetto corrente*. Questo viene fatto dal menu **File…** -> **Project…** e selezionando **Set project to current file** come mostrato in Figura 13.



➢ Figura 13 – Imposta il file come progetto di lavoro

A questo punto è possibile procedere alla compilazione del progetto selezionando dal Menu **File… -> Project…** la voce **Save & Compile** che ora è abilitata (vedi Figura 14).



➢ Figura 14 – Save & Compile

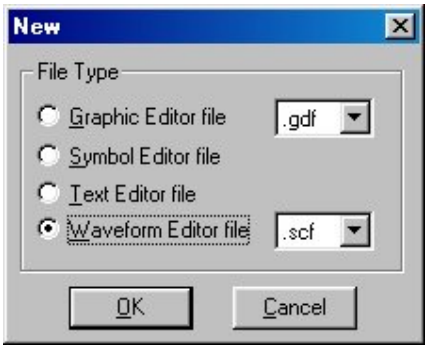
Se tutto si svolge correttamente il programma conclude la compilazione segnalando nessun errore e nessun warning (Figura 15). Altrimenti vengono dati dei suggerimenti su l tipo di errore che si è verificato in fase di compilazione. E' importante notare che eventuali errori segnalati dal compilatore non riguardano la logica di funzionamento del nostro circuito ma semplicemente segnalano che lo schema logico non è stato inserito secondo le specifiche di Max + Plus II. Per analizzare il funzionamento del circuito è necessario procedere alla simulazione.

➢ Figura 15 – Rapporto di compilazione

Simulazione

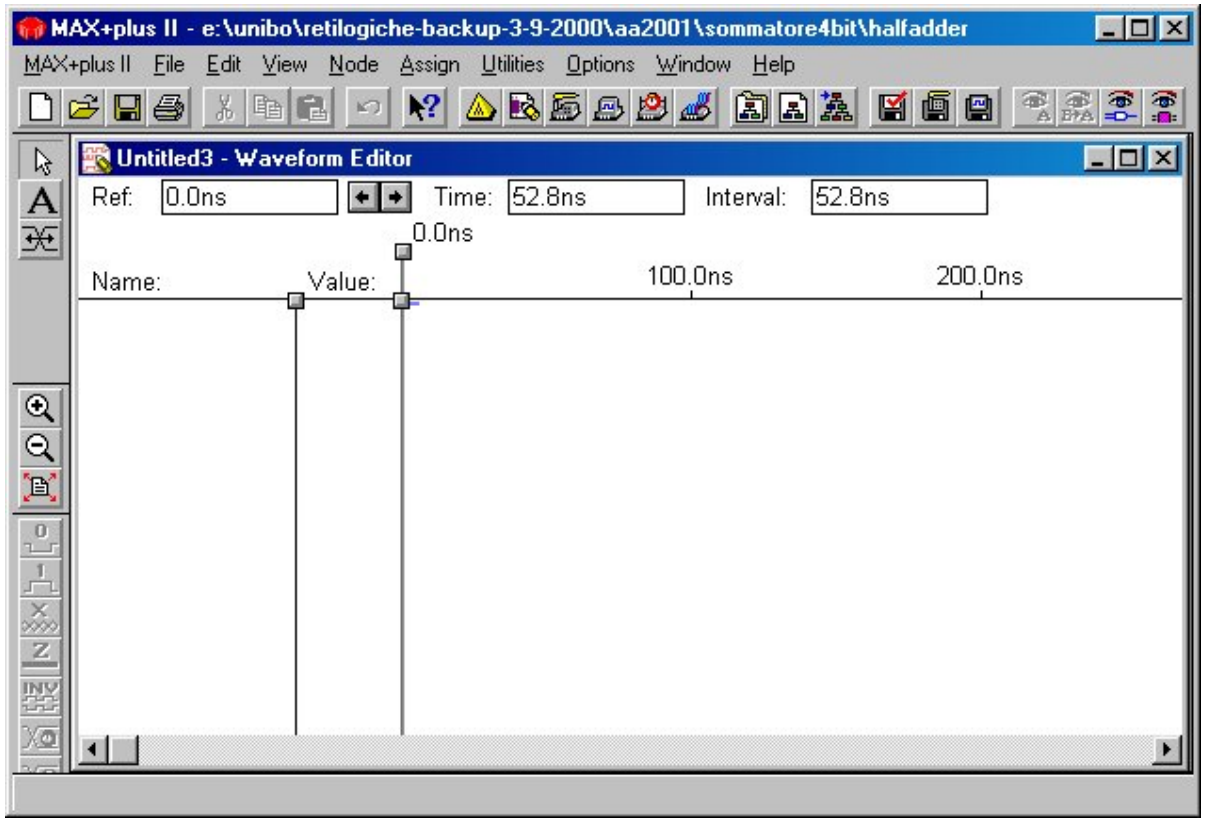
Una volta inserito lo schema logico è possibile simularne il funzionamento per verificare che il funzionamento sia quello desiderato.

Per prima cosa è necessario creare un file con le forme d'onda dei segnali di ingresso, che dovranno essere specificati, e dei segnali di uscita che si desiderano visualizzare.
Per fare questo da menu **File** selezionare **New**. Appare la seguente finestra di dialogo



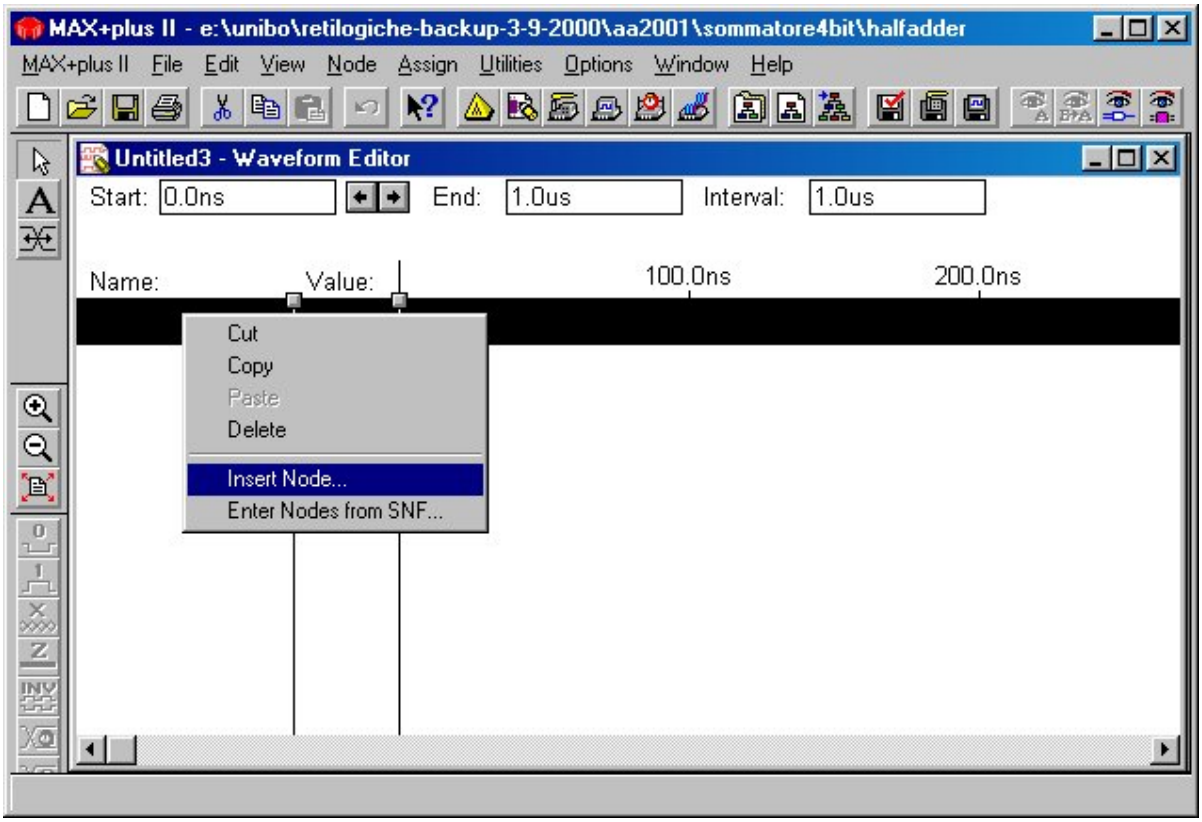
➢ Figura 16 – New File Dialog

Selezionando **Waveform Editor File** (estensione del file ".scf") si entra nel *Waveform Editor* con un nuovo foglio di lavoro in cui inserire e visualizzare l'andamento temporale dei segnali.



➢ Figura 17 – *Waveform editor*: nuovo foglio di lavoro

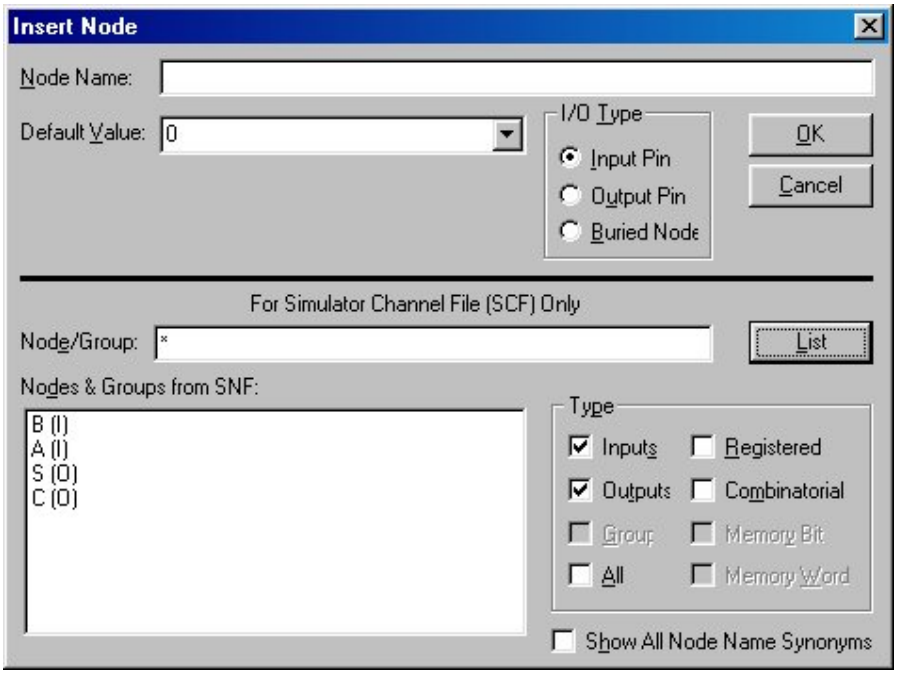
Cliccando sotto la regione con la scritta *name* con il tasto destro del mouse appare la seguente *dialog box*.



➢ Figura 18 - Insert node…

Selezionando **Insert Node…** appare una finestra di dialogo dalla quale selezionare i segnali che desideriamo visualizzare.

Per avere la lista dei segnali definiti nel progetto è sufficiente *cliccare* su **List**. A questo punto si selezionano i segnali che si vogliono inserire nel *waveform editor*.



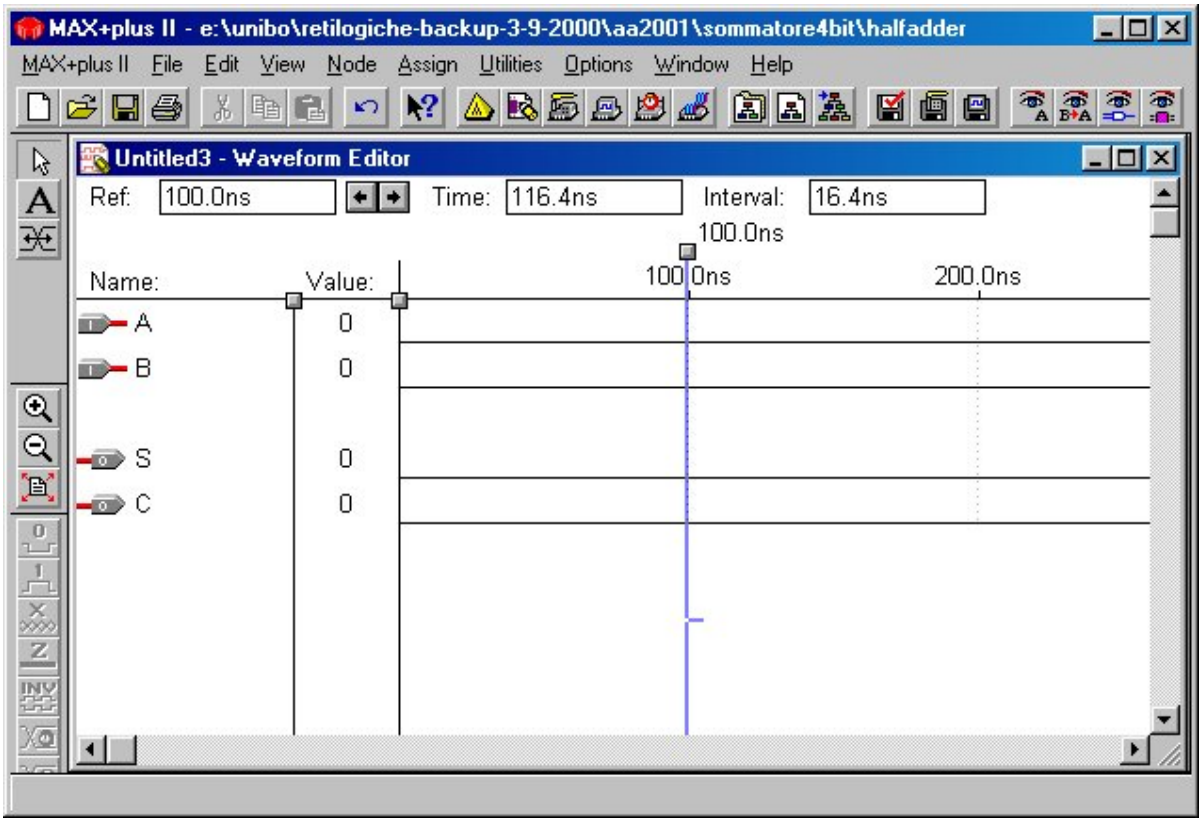
➢ Figura 19 – Insert Node Dialog

Nel nostro caso i segnali definiti sono A,B,S,C. Si noti che a fianco di ogni segnale è mostrato se di *input* (I) o di *output* (O).

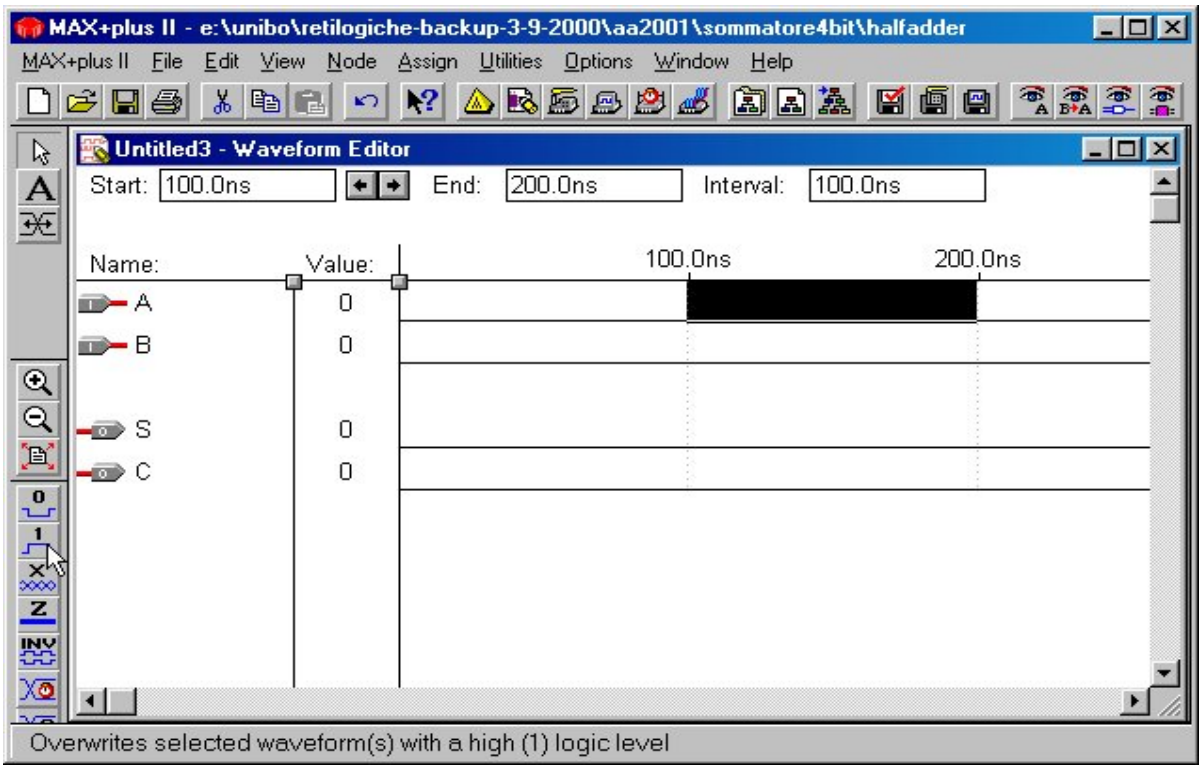
Una volta inseriti tutti i segnali nel *WaveForm Editor* (Figura 20) si procede ad assegnare ai segnali di ingresso forme d'onda di prova. A tal fine si seleziona con il tasto sinistro del mouse un certo intervallo di tempo e *si setta* l'intervallo di tempo in cui il segnale sta a livello logico "1" o "0". A tal fine si utilizzano i pulsanti posti alla sinistra nel *WaveForm Editor* (Figura 21).

N.B. E' possibile inserire all'interno del *WaveForm Editor* tutti i nodi con una sola operazione selezionando "***Enter Node from SNF…***" dal menu di Figura 18.

Ad esempio:
Si consideri il segnale di ingresso A. Si supponga che per tale segnale il comportamento desiderato sia questo: *'0' in tutto l'intervallo di tempo della simulazione eccetto che nell'intervallo da 100.0 ns a 200.0 ns in cui deve valere '1'*.
Poiché per *default* un segnale è posto a '0' è necessario selezionare l'intervallo da 100.0 ns a 200.0 ns con il mouse (tenendo premuto contemporaneamente il tasto sinistro). Una volta che il segnale è evidenziato per l'intervallo di tempo desiderato (Figura 21) è possibile, in tale intervallo, settarlo al livello logico '1' *cliccando* sul bottone '**1**' posto alla sinistra del WaveForm Editor come mostrato in Figura 21.

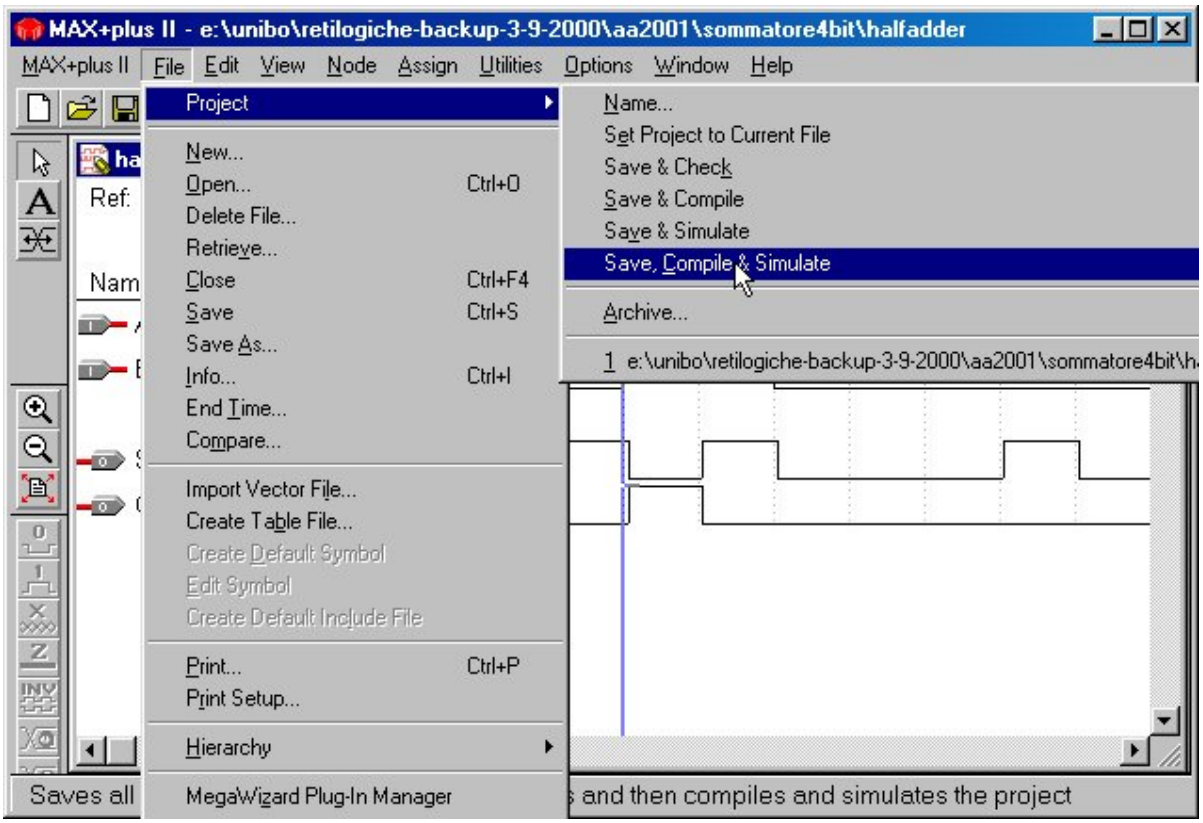


➢ Figura 20 – Waveform editor

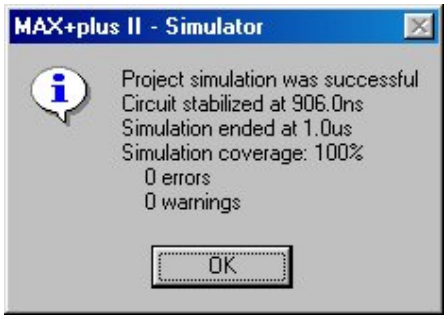


➢ Figura 21 – Waveform editor: impostazione livello logico

Quando si è ultimato di assegnare i valori dei segnali di ingresso è necessario salvare la forma d'onda (**File-> Save**) utilizzando il nome assegnato di default ("halfadder.scf") per procedere alla simulazione. La simulazione viene lanciata selezionando **Save, Compile & Simulate** dal menu **File… -> Project** come mostrato in Figura 22.

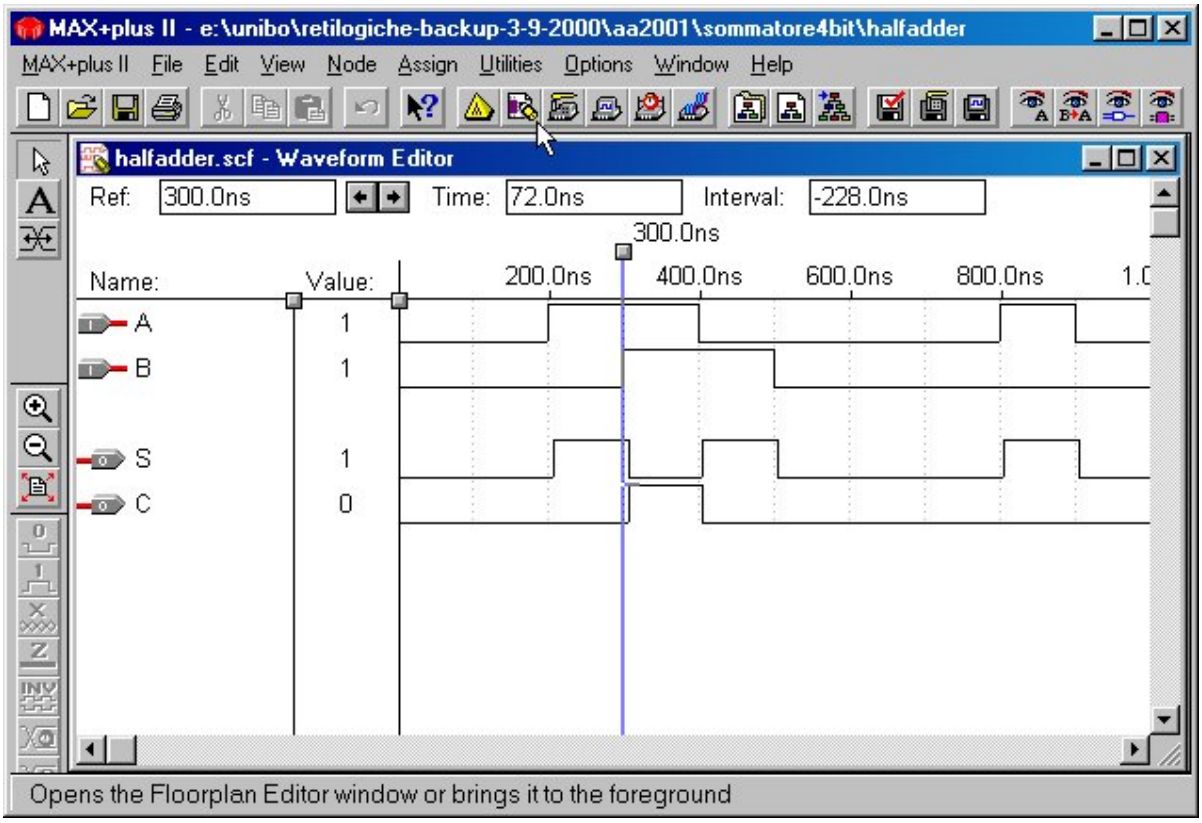


➢ Figura 22 – Salva, compila e simula



➢ Figura 23 – Rapporto della simulazione

Se tutto procede senza errori, Figura 23, è possibile visualizzare il risultato della simulazione (Figura 24).

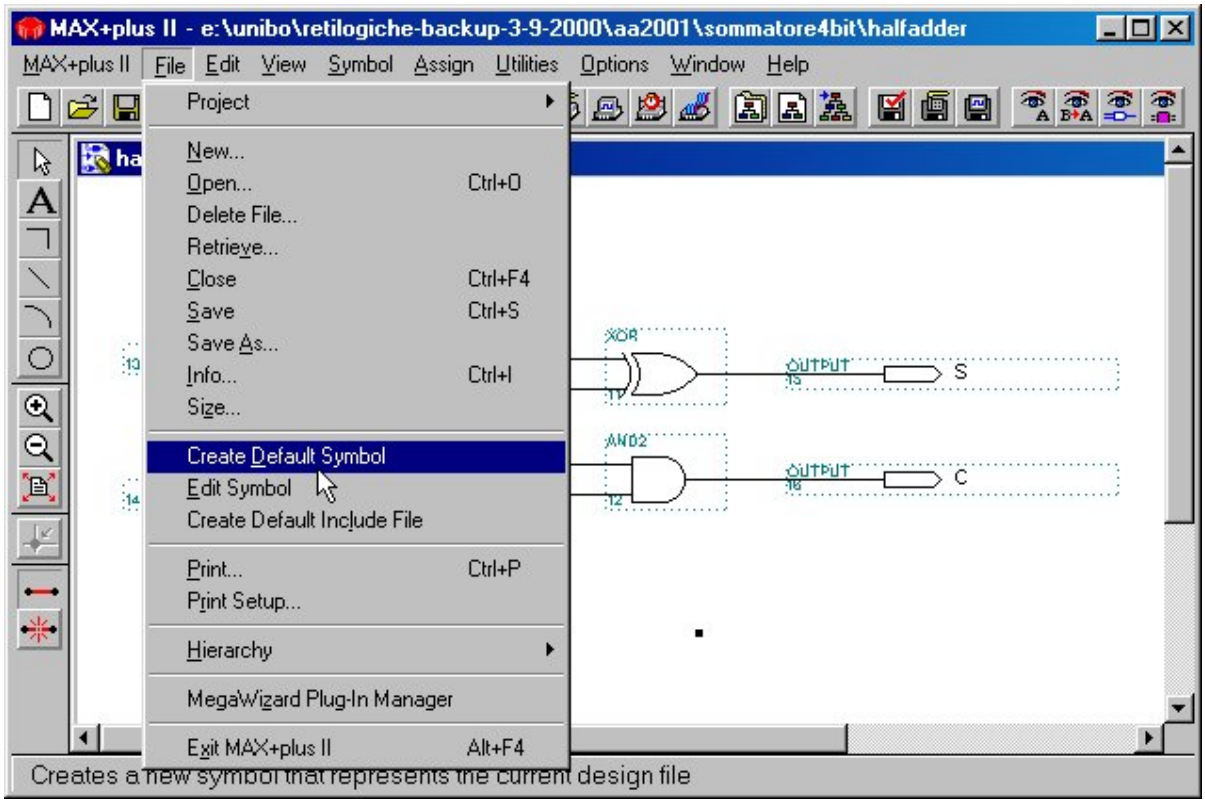


➢ Figura 24 – Simulazione Half-Adder

Come è possibile osservare da Figura 24 il circuito è stato progettato correttamente poiché le forme d'onda relative a S e C riproducono il comportamento atteso da un Half-Adder.

Creazione simbolo logico

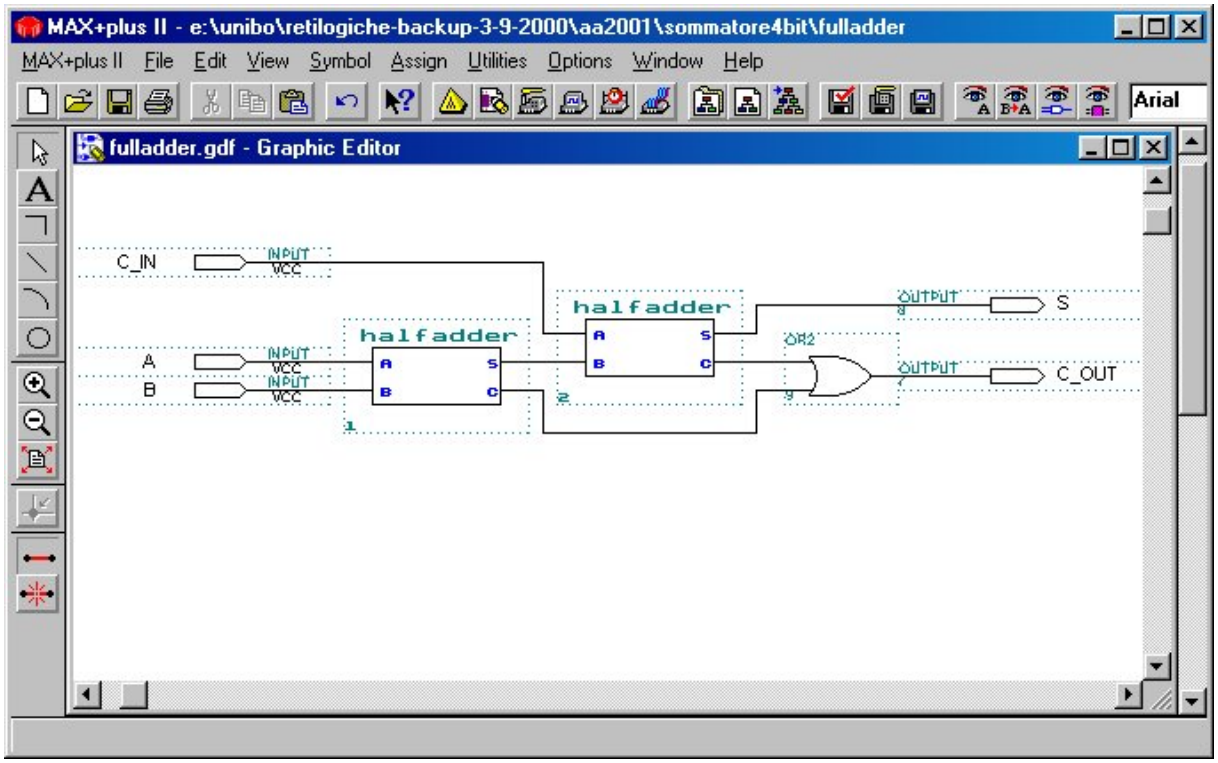
A questo punto è possibile definire un simbolo logico per il circuito progettato scegliendo dal menu **File….** la voce **Create Default Symbol** (Figura 25) e selezionare il nome da attribuire al simbolo logico. Nel nostro esempio potremmo creare un simbolo denominato "halfadder.sym". L'estensione dei file contenenti simboli logici è ".sym". In questo modo sarà possibile inserire direttamente il simbolo dell'Half-Adder progettato, seguendo la procedura descritta precedentemente, nei livelli superiori della gerarchia di progetto.



➢ Figura 25 – Creazione di un simbolo

Full-Adder a 1 bit (Level 2)

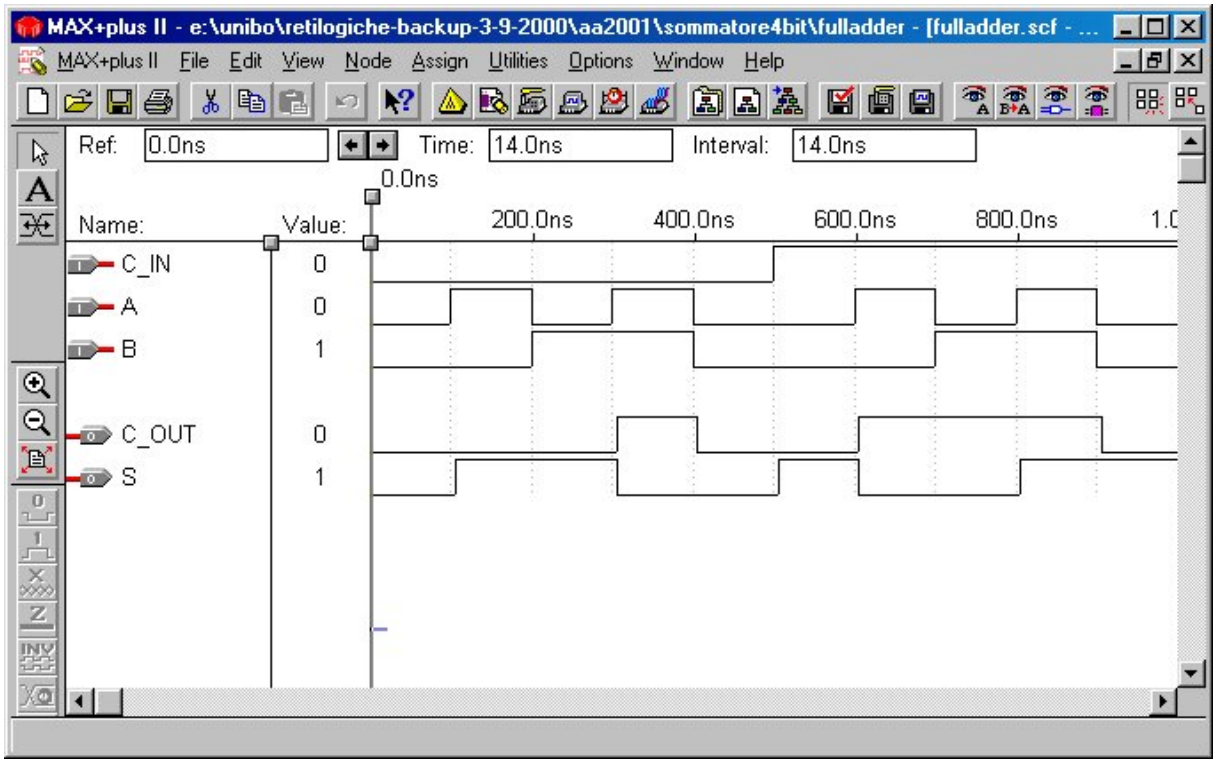
Partendo dall'Half-Adder appena simulato, e salendo nella gerarchia di progetto, inseriremo lo schema logico di un Full-Adder creando per prima cosa un nuovo Graphic Editor File (Figura 3).
Inseriamo i 2 simboli logici dell'Half-Adder necessari seguendo la procedura descritta precedentemente (Figure 5-6-7). In questo caso il simbolo logico dell'Half-Adder andrà cercato, all'interno del filesystem, nella posizione in cui è stato precedentemente memorizzato. Introducendo gli altri simboli necessari per la progettazione del Full-Adder e connettendoli nel modo opportuno si ottiene lo schema logico di figura 26.



➢ Figura 26 – Half-Adder

Prima di procedere alla compilazione del progetto, con **File…-> Project …-> Save & Compile,** è necessario salvare lo schema logico inserito attraverso li menu' **File…-> Save as** e settando lo stesso come progetto corrente (**File…-> Project …-> Set Project to Current File**) .

Per la simulazione si procederà (**File…-> Project…->Save Compile & Simulate**) come descritto precedentemente creando le forme d'onda per i segnali di ingresso utilizzando il **Waveform Editor**. Nel caso del Full-Adder vi sono 3 segnali di ingresso (A,B,C_IN) e due segnali di uscita (S,C_OUT). Il risultato della simulazione è mostrato nella figura seguente.

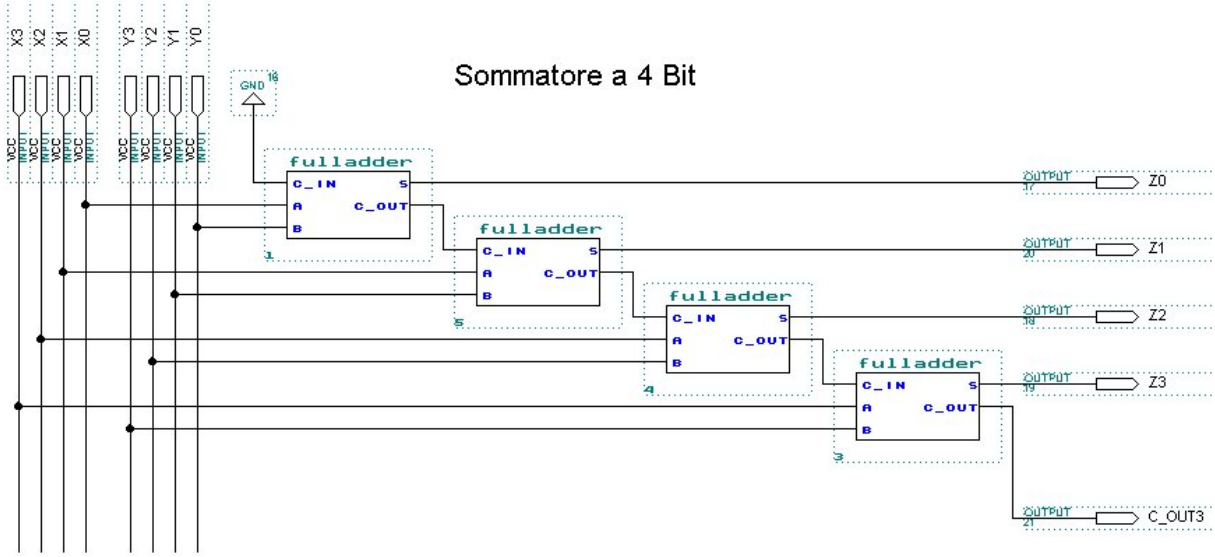


➢ Figura 27 –Simulazione Full-Adder

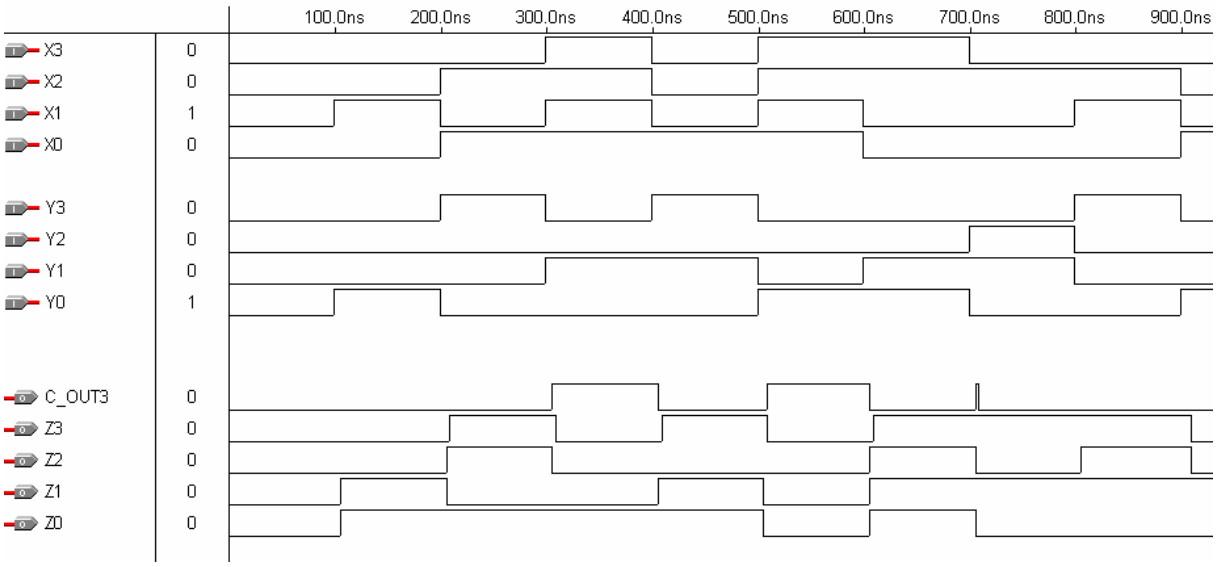
La simulazione dimostra che lo schema logico di Figura 26 è una corretta realizzazione di un Full-Adder. E' quindi possibile creare il simbolo logico del Full-Adder ed inserirlo nella nostra libreria personale. Questo viene ottenuto dal menu **File-> Create Default Symbol** (il simbolo creato è "*fulladder.sym*").

Full-Adder a 4 Bit (Level 1)

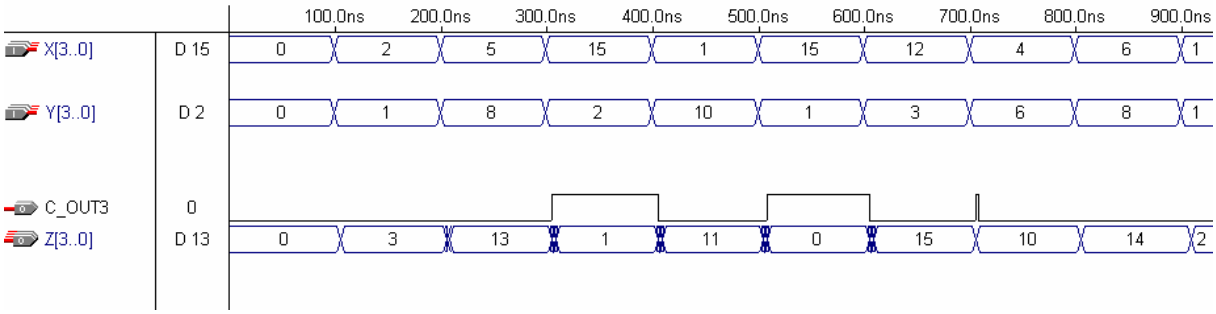
Analogamente a quanto fatto fino ad ora saliamo ulteriormente nella gerarchia del progetto utilizzando il Full-Adder precedentemente simulato come base per realizzare lo schema logico di un sommatore a 4 bit (Figura 28).



➢ Figura 28 – Livello gerarchico 1



➢ Figura 29 – Risultato simulazione



➢ Figura 30 – Risultato simulazione

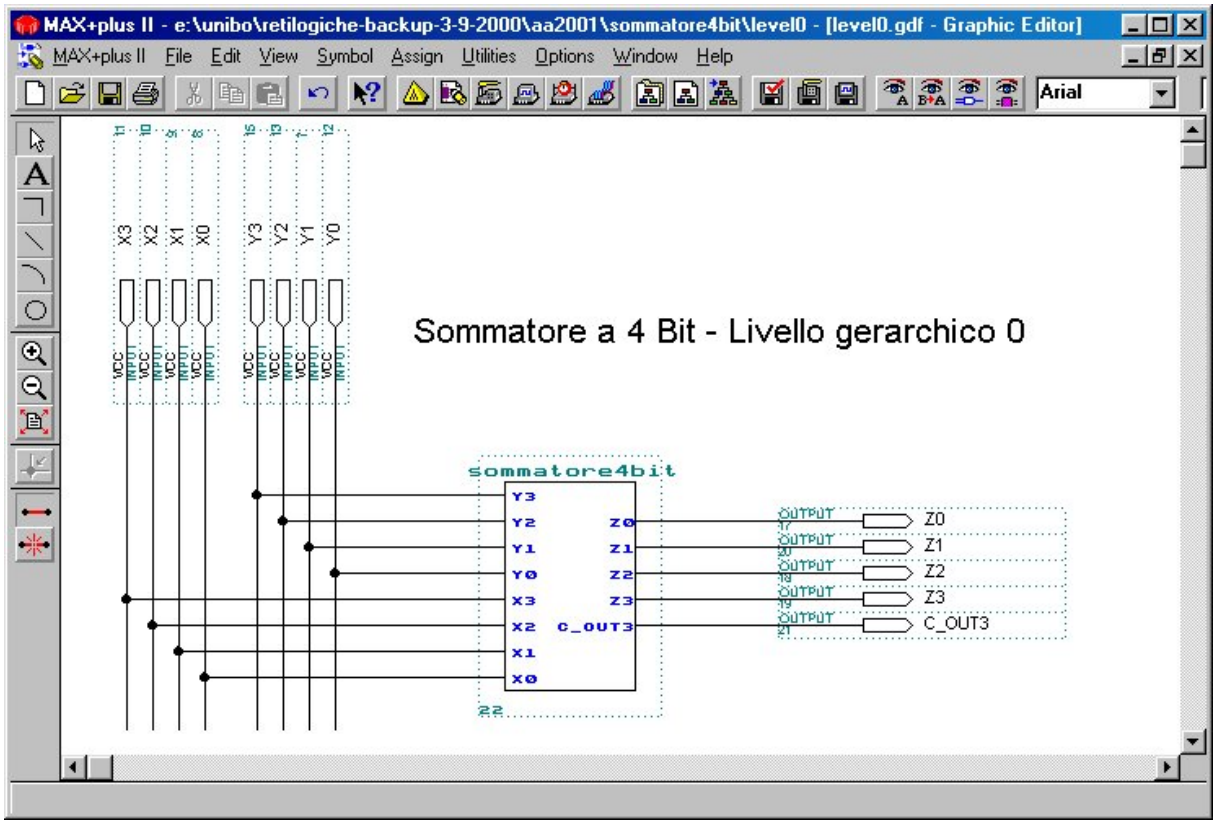
Il risultato della simulazione, Figure 29 e 30, mostra che in alcune circostanze, quando i segnali di ingresso variano, possono verificarsi delle Alee sul bit di Carry Out (C_OUT).

A cosa sono dovute tali alee?
E' possibile evitarle ?
Come funziona una rete che deve essere insensibile a queste alee ?

Si noti in Figura 30 come i segnali siano stati raggruppati logicamente consentendo una maggiore leggibilità.

NOTA: Quando si raggruppano dei segnali come in figura 20 si devono disporre, all'interno del *WaveForm Editor*, i segnali in ordine decrescente dall'alto verso il basso partendo dai bit più significativi.

Sommatore a 4 bit (Livello gerarchico 0, "Entity")



➢ Figura 31 – Progetto finale

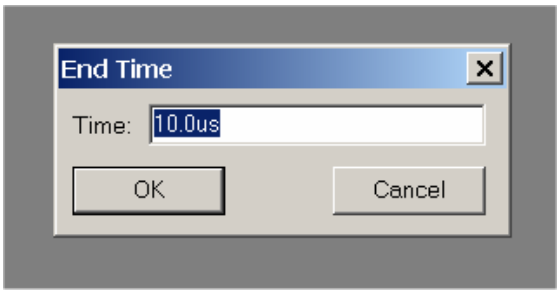
Cambiamento dell'intervallo di simulazione

La modifica della durata dell'intervallo di simulazione può essere ottenuta seguendo questi passi:

1) si rende attiva la finestra contenente il '*Waveform editor*' (ad esempio, come mostrato in Figura 24)

2) dal menu '**File'** si seleziona "**File -> End Time…**" (vedi Figura 22)

3) si introduce il nuovo intervallo di simulazione come indicato nella figura seguente



➢ Figura 32 – Nuovo intervallo di Simulazione

4) si procede alla ricompilazione del progetto

E' importante osservare che aumentando il tempo di simulazione e' necessario ridefinire i segnali di ingresso nell'intervallo di tempo che precedentemente non veniva esaminato.

Indice