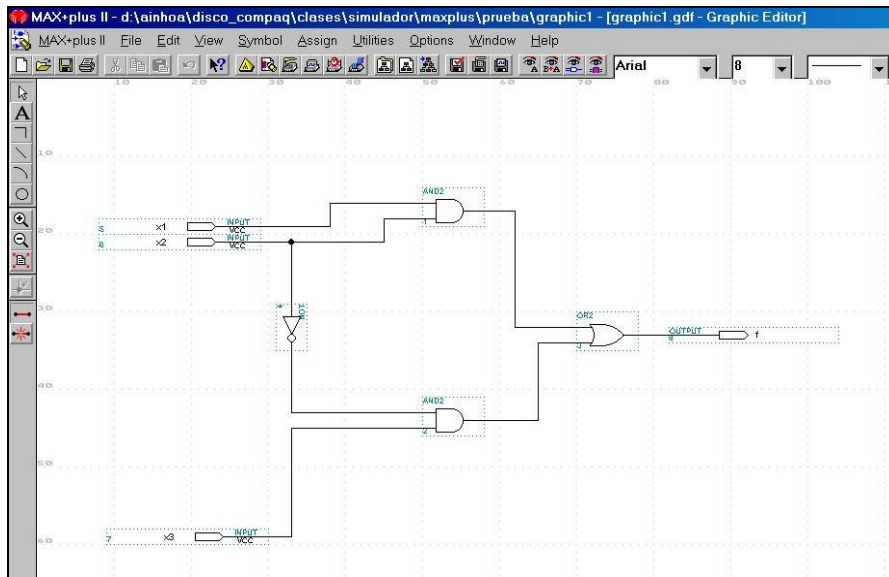




82. Irudia

*Uhin*en Atzipena. Diseinuaren deskribapena uhinak erabiliz eta denbora-diagrama osatuz garatzen da. Diagrama horretan erlazionatu behar dira sarrerak irteerekin.

Hardwarea deskribatzeko lengoia (HDL). Zirkuituaren diseinua egiteko lengoia berezi baten bidez zirkuituaren funtzionalitatea deskribatzen da. Deskripzio horretan, irteerak sarrerekin erlazionatu behar dira. Gaur egun, gehien erabiltzen diren lengoaiak *Verilog HDL* eta *VHDL* dira. Azken hori, agian, garrantzitsuenetakoa da IEEE estandarra delako.

Adibide moduan, 4 biteko bi zenbakiren arteko batuketa garatzen duen zirkuituaren VHDL programazioa ematen da:

```

LIBRARY ieee;
USE ieee.std_logic1164.all;
USE work.fulladd_package.all;

```

```

ENTITY adder4 IS
    PORT (Cin           : IN  STD_LOGIC;
          x3,x2,x1,x0:   : IN  STD_LOGIC;
          y3,y2,y1,y0:   : IN  STD_LOGIC;
          s3,s2,s1,s0:   : OUT STD_LOGIC;
          Cout          : OUT STD_LOGIC);
END adder4

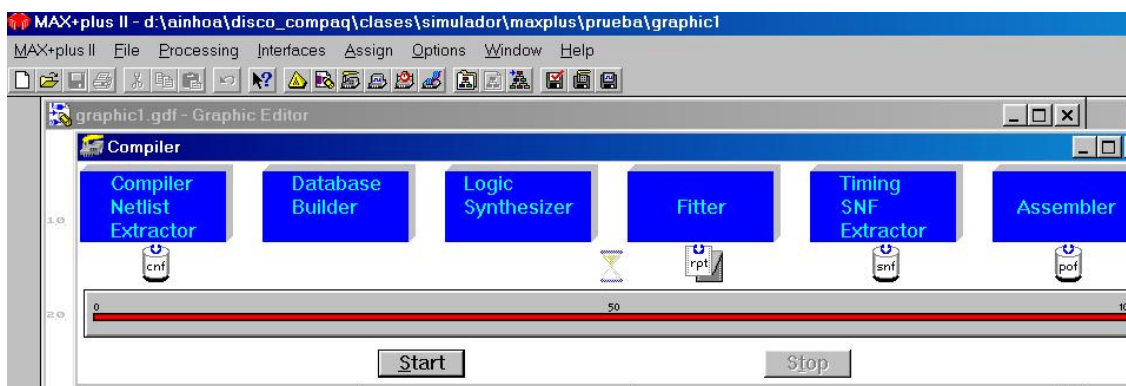
ARCHITECTURE Structure OF adder4 IS
    SIGNAL c1,c2,c3: STD_LOGIC;
BEGIN
    Stage0: fulladd PORT MAP (Cin,x0,y0,s0,c1);
    Stage1: fulladd PORT MAP (c1,x1,y1,s1,c2);
    Stage2: fulladd PORT MAP (c2,x2,y2,s2,c3);
    Stage3: fulladd PORT MAP (
        Cin => c3, Cout => Cout, x => x3, y => y3, s => s3);
END Structure;

```

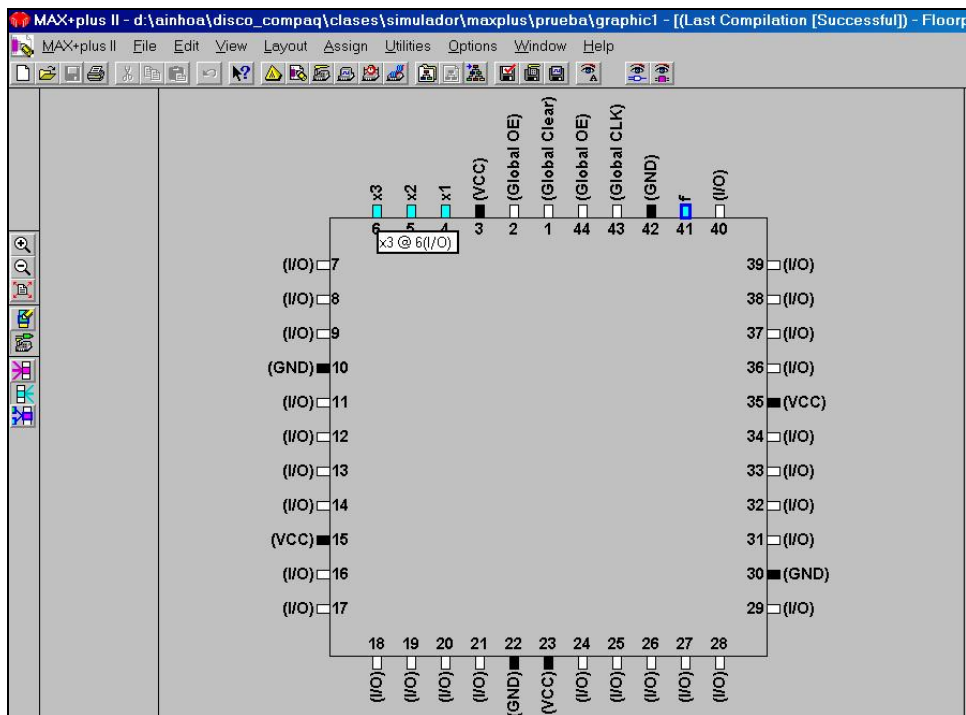
Badira funtzio zehatzak garatzen dituzten zirkuituen VHDL deskripzio aurrez eginikoak eta horretaz diharduen edozein ordenagailu-programatan eskuragarri daude LPM librerietan (Library Parameterized Modules).

5.2.Konpilazioa

Deskripzioa zuzena den arakatzeko duen prozesua da; hots, idazkera akatsik ez duela, eta horrela bada, gailuaren programazioa egiteko beharrezkoak diren artxiboak sortzen dituen prozesua da. Ekintza horri konpilazio deritzogu eta konpiladorea da horren arduraduna. Bere zereginak betetzeko konpiladoreak ematen dituen pausuak dira (83. Irudia): 1) konexio elektrikoak aztertu, 2) optimizatu, 3) programaziorako aukeratu den gailuan zirkuitua sartzeko modurik egokiena aukeratu, 4) barneko konexio eta pinen esleipenaren planoak eman (84. Irudia) , eta 5) gailua programatzeko beharrezkoa den **JEDEC** (*Joint Electron Device Engineering Council*) fitxategia sortu.



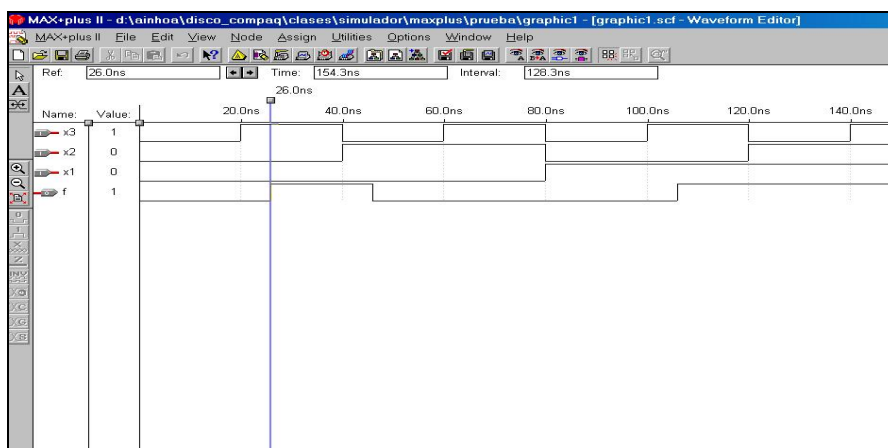
83. Irudia



84. Irudia

5.3. Simulazioa

Garatutako zirkuitu-diseinuak funtzionamendu egokia duela aztertzea ezinbestekoa da gailuaren programazioa egin aurretik. Horretarako, simulaziorako softwarea erabili behar da, baita konpiladoreak emandako informazioa ere. Informazio horretan, erabiltzaileak definitutako sarrera-bektoreak eta aukeratutako gailuen bibliotekak laburbilduta daude. Hortaz, erabiltzaileak eskuragarri dituen sarrera- eta irteera-bektoreak aukeratzen ditu, eta nahi duen sarrera-seinalea definitu ondoren, erantzuna aztertu behar du irteera-bektoreetan (85. Irudia).



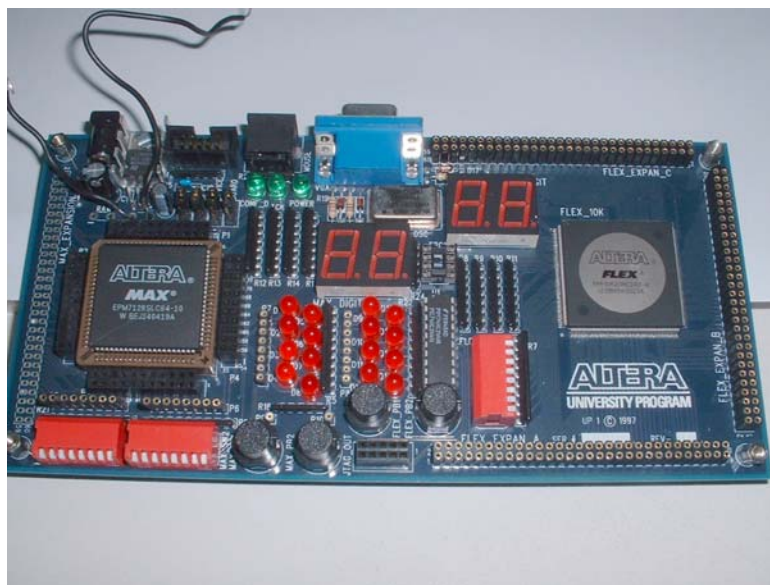
85. Irudia

5.4. Programazioa

Atal hau prozesuaren azkenengo pausoa da, hots, gailuaren programazioa. Horretarako, beharrezkoa da *programadore* izeneko hardware-sistema eskuragarri izatea. Programadorea ordenagailuari konektatu behar zaio, eta gailuaren programazioa lortzen da JEDEC artxiboaz baliatuz.

Programadore mota asko daude, fabrikatzaile bakoitzak bere gailuak programatzeko hardwarea baitu, baina badaude *unibertsal* deituriko programadoreak, zeinek edozein fabrikatzailearen gailuak programatu ditzaketen.

86. Irudian, MAX7000 eta FLEX 10K gailuak programatzen dituen programadorea aurkezten da.



86. Irudia