

4 Editarea schematică și simularea funcționării circuitelor cu software specializat (II)

4.1 Obiective

Se prezintă modalități de gestionare a librăriilor în utilitarul Logisim și posibilitatea de dezvoltare ierarhică a circuitelor. Se exemplifică un circuit construit în stil ierarhic. Sunt redate modalități de analiză a circuitelor folosind utilitarul de analiză combinațională.

4.2 Gestionarea librăriilor

Un proiect în Logisim are atașată în mod implicit librăria proiectului, din care vor face parte circuitele dezvoltate sub formă de scheme logice. Fișierul în care se salvează proiectul conține librăria acestuia împreună cu toate schemele logice și dependențele sale. Acest lucru dă posibilitatea încărcării librăriei din fișierul respectiv într-un alt proiect, astfel că se pot include circuitele sale ca și componente ale altor scheme logice. Utilizarea într-o schemă logică a unor circuite din librăria curentă sau din alte librării poartă denumirea de *dezvoltare ierarhică*. În Logisim, încărcarea unei librării se realizează cu comanda **Load Library > Logisim-library...** din secțiunea **Project** a meniului. În pasul următor, se va selecta un fișier de proiect de pe disc (Figura 4. 1) din care Logisim va încărca librăria acestuia.

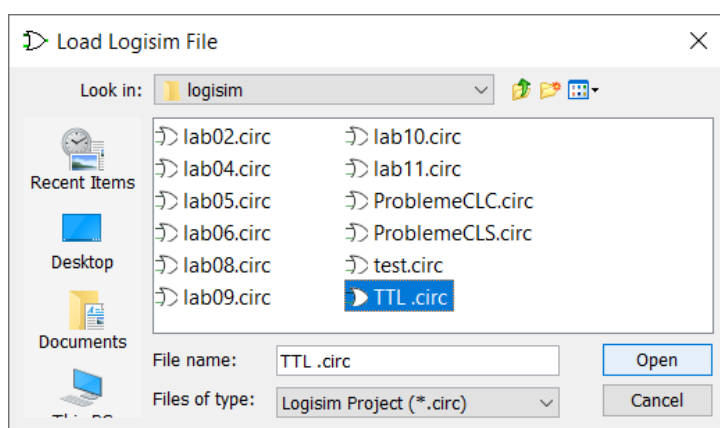


Figura 4. 1 La încărcarea unei librării se selectează fișierul proiectului care implementează librăria respectivă

Ulterior, librăria încărcată va apărea în Toolbox, între celelalte librării disponibile, iar circuitele sale vor fi disponibile sub forma simbolurilor asociate. În Figura 4. 2 este exemplificat accesul la circuitele din librăria TTL. În cadrul unui proiect se pot încărca oricâte librării sunt necesare, ceea ce face posibilă utilizarea componentelor din librării multiple.

Atunci când se dorește eliminarea unei librării din proiect se poate apăsa butonul din dreapta al mouse-ului pe numele său și se alege opțiunea **Unload Library**. Alegând **Reload Library** se poate actualiza o librărie care este deja în Toolbox.

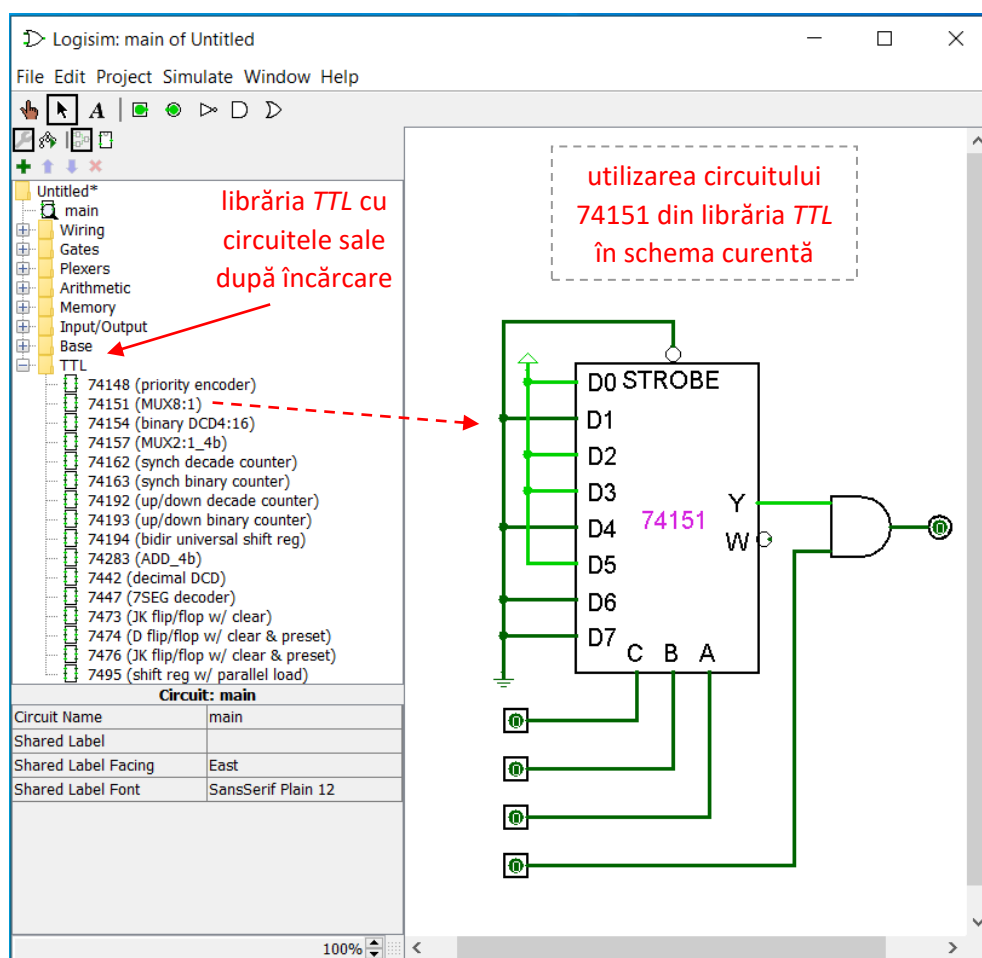


Figura 4. 2 După încărcare, biblioteca TTL va apărea în Toolbox și circuitele sale se pot accesa cu ajutorul simbolurilor asociate

4.3 Analiza unui circuit

Odată schema logică finalizată, se pot realiza următoarele tipuri de analiză:

- Analiza circuitului;
- Crearea de statistici.

Analiza circuitului se realizează cu comanda **Analyze Circuit** din secțiunea **Project** a meniului. Ulterior, va apărea utilitarul de analiză combinațională în fereastra **Combinational Analysis**, care are mai multe tab-uri (Figura 4. 3):

- **Inputs** – conține lista de terminale de intrare detectate. Ele sunt identificate după eticheta lor sau primesc automat un nume în lipsa etichetei.
- **Outputs** – conține lista de terminale de ieșire detectate. Similar cu intrările, ieșirile sunt identificate după etichetă sau primesc automat un nume.
- **Table** – evidențiază tabelul de adevăr al ieșirilor detectate în cadrul circuitului. Apăsând în tabel pe valorile de ieșire acestea își schimbă valoarea, oscilând între 0, 1 și x.

- **Expression** – evidențiază expresiile funcțiilor logice aferente fiecărei ieșiri, în Forma Disjunctivă Minimă sau în Forma Conjunctivă Minimă, în funcție de configurările din tab-ul **Minimized**.
- **Minimized** – evidențiază expresiile minimizate ale funcțiilor logice aferente fiecărei ieșiri, ca sumă de produse (Forma Disjunctivă Minimă) sau ca produs de sume (Forma Conjunctivă Minimă). Dacă sunt maxim 4 intrări, atunci se va evidenția și Diagrama Karnaugh a funcțiilor, împreună cu grupările de minimizare. Apăsând în celulele diagramei, valorile se modifică între 0, 1 și x, iar grupările și minimizarea se actualizează automat. Dacă butonul **Set As Expression** este activ, atunci trebuie apăsat pentru ca expresia minimizată să fie luată în considerare.

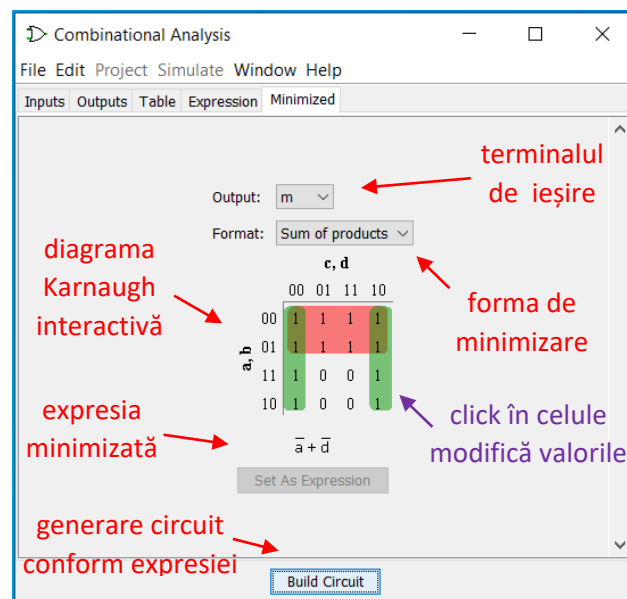


Figura 4. 3 Analiza combinațională a circuitului din Figura 4. 4

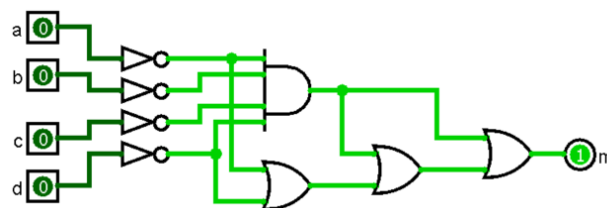


Figura 4. 4 Schemă logică complexă a cărei expresie se poate simplifica cu utilitarul de analiză combinațională

Notă: Analiza circuitului funcționează corect numai pentru circuite combinaționale: ale căror ieșiri depind strict de valorile de pe intrări.

Oricare dintre tab-uri conține butonul **Build Circuit** cu ajutorul căruia se va genera circuitul corespunzător funcțiilor logice curente din utilitarul Combinational Analysis. Se va solicita numele noului circuit și apoi va fi inclus în librăria proiectului. De exemplu, circuitul rezultat după minimizarea schemei din Figura 4. 4, cu utilitarul de analiză combinațională, este redat în Figura 4. 5.

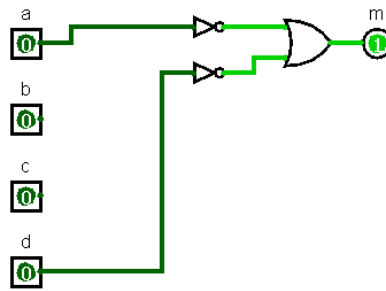


Figura 4. 5 Schema logică rezultată prin minimizarea circuitului din Figura 4. 4

Utilitarul de analiză se poate folosi și pentru generarea schemei unui circuit definit în formă analitică sau grafică. Chiar dacă nu există niciun element pe schemă, utilitarul se poate lansa și cu comanda **Combinational Analysis** de la secțiunea **Window** a meniului. În cadrul acestei ferestre, la tab-ul **Inputs** se vor introduce etichetele terminalelor de intrare, iar la tab-ul **Outputs**, pe cele ale terminalelor de ieșire. Funcțiile logice corespunzătoare terminalelor de ieșire se pot defini analitic la tab-ul **Expression**, prin tabel de adevăr la tab-ul **Table** sau prin diagramă Karnaugh, dacă au cel mult 4 intrări, la tab-ul **Minimized**. După definirea funcțiilor într-una din forme se poate crea circuitul corespunzător cu comanda **Build Circuit**.

Notă: La introducerea prin expresie analitică se acceptă operatorii NOT, AND, OR, XOR. Inversarea NOT se reprezintă cu caracterul \sim , operația AND se reprezintă cu un spațiu, OR cu caracterul $+$, iar XOR cu caracterul \wedge . Se pot folosi paranteze dacă este necesar. După introducerea unei expresii se apasă butonul **Enter** (Figura 4. 6).

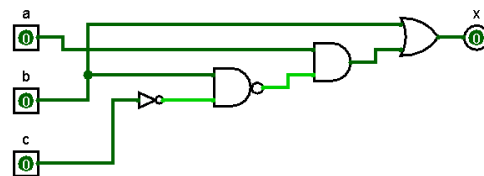
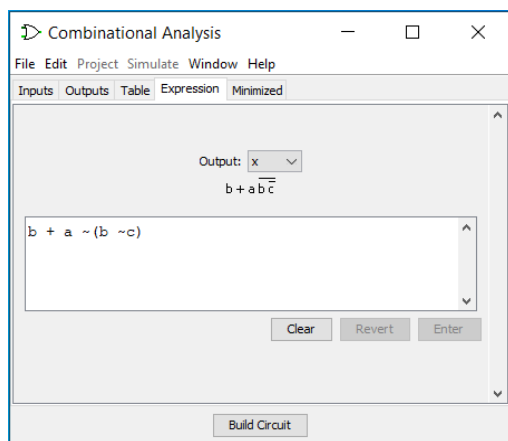
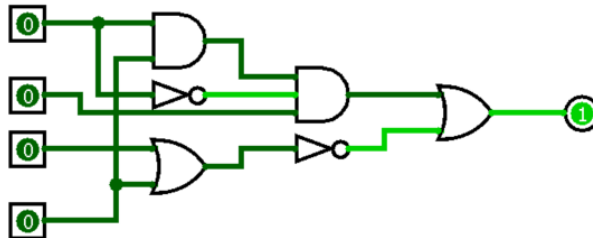


Figura 4. 6 Introducerea unei expresii în utilitarul de analiză combinațională (stânga).
Circuitul generat corespunzător expresiei (dreapta)

Generarea de statistici se realizează cu comanda **Get Circuit Statistics** din secțiunea **Project** a meniului. Va apărea o fereastră cu diverse statistici legate de circuitele din componența schemei logice curente.

4.4 Activități practice

1. Creați un proiect nou în Logisim și încărcați librăria *TTL*. Vizualizați lista circuitelor din componența sa.
2. Creați o nouă schemă pentru circuitul din figura următoare și folosiți utilitarul de analiză combinațională pentru a genera circuitul echivalent obținut prin minimizare cu tab-ul **Minimize**.



3. Folosiți tab-urile **Inputs**, **Outputs** și **Expression** ale utilitarului de analiză combinațională pentru a implementa următoarele funcții logice într-o singură schemă logică:

$$f_1 = \sim a + b + \sim c \quad f_2 = \sim(a + b) (a + c) \quad f_3 = \sim a + \sim(\sim b c)$$

4. Realizați schema din figura următoare, care reprezintă un comparator fără semn pe 2 biți folosind comparatoare fără semn pe 1 bit (cu atributul *Numeric Type=Unsigned*) din librăria *Arithmetic*, și alte porți logice fundamentale necesare. Testați funcționarea comparatorului în simulator, pentru toate combinațiile posibile pe intrări.

Explicații la schemă: Cele 2 numere pe 2 biți care se compară sunt A_1A_0 și B_1B_0 . Se iau în considerare următoarele reguli pentru cele 3 situații posibile:

- $F_1 = A_1A_0 > B_1B_0$: $F_1=1$ dacă $A_1 > B_1$ SAU ($A_1=B_1$ ȘI $A_0 > B_0$);
- $F_2 = A_1A_0 = B_1B_0$: $F_2=1$ dacă $A_1=B_1$ ȘI $A_0=B_0$;
- $F_3 = A_1A_0 < B_1B_0$: $F_3=1$ dacă $A_1 < B_1$ SAU ($A_1=B_1$ ȘI $A_0 < B_0$).

