

3 EDITAREA SCHEMATICĂ ȘI SIMULAREA FUNCȚIONĂRII CIRCUITELOR CU SOFTWARE SPECIALIZAT (II)

3.1 Scopul lucrării

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. Se descrie simularea circuitelor secvențiale cu terminale de tip Clock și se prezintă modalități de manipulare a acestora în timpul simulării.

3.2 Considerații teoretice

3.2.1 Gestionarea librăriilor

Un proiect în Logisim creează în mod implicit o librărie a proiectului în care se pot dezvolta mai multe circuite 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, pentru a folosi circuitele sale în cadrul dezvoltării 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 din care Logisim va încărca librăria acestuia (Figura 3. 1).

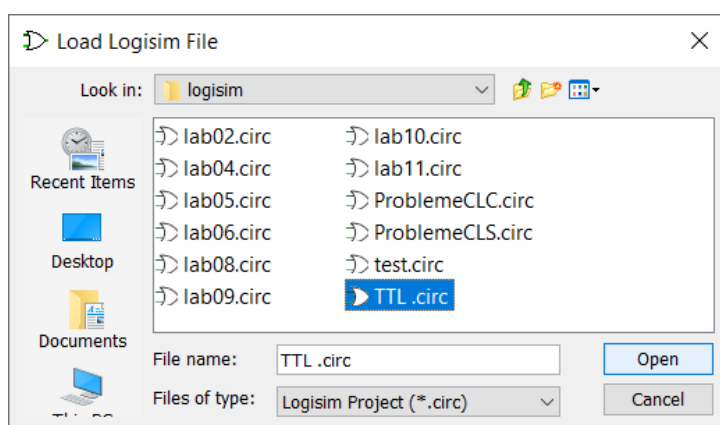


Figura 3. 1 La încărcarea unei librării se selectează fișierul proiectului care implementează librăria respectivă

Ulterior, librăria încărcată va apărea în Toolbar, între celelalte librării disponibile, iar circuitele sale vor fi disponibile sub forma simbolurilor care le reprezintă (Figura 3. 2). În cadrul unui proiect se pot încărca mai multe librării, dacă este necesar.

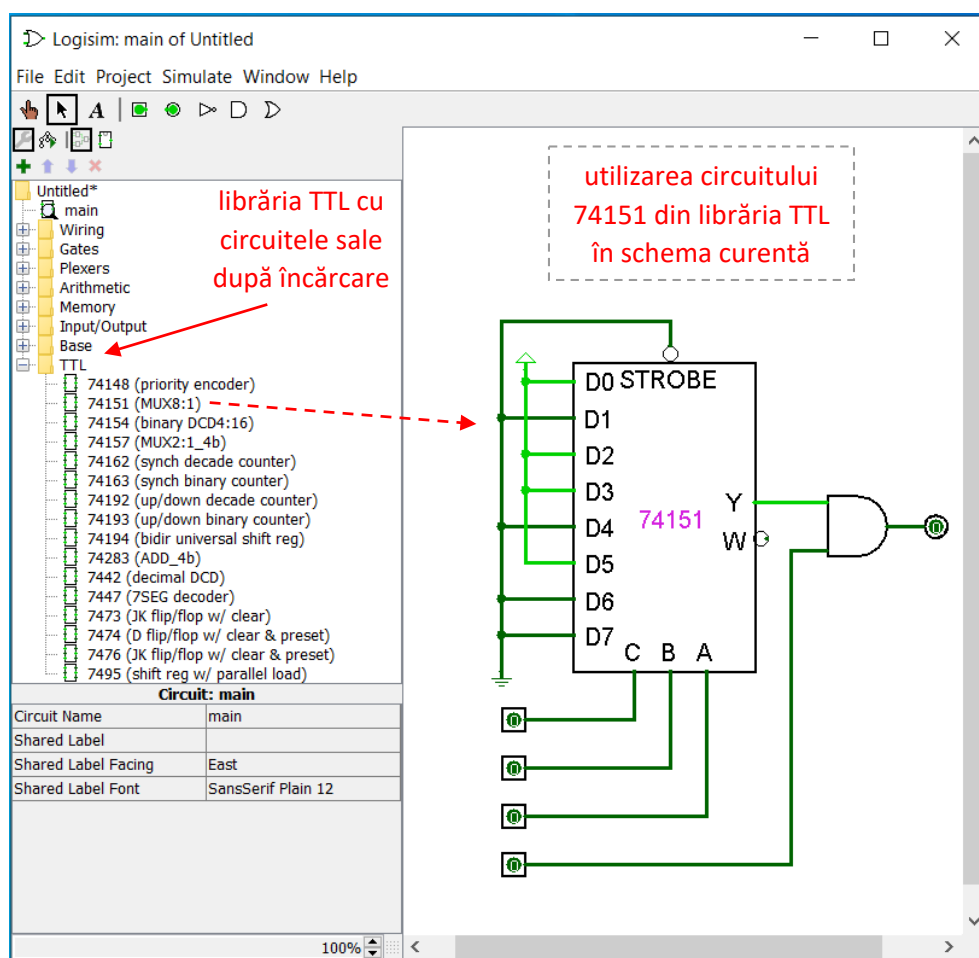


Figura 3. 2 După încărcare biblioteca TTL apare în Toolbox și circuitele sale se pot accesa cu ajutorul simbolurilor asociate

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

3.2.2 Analiza unui circuit

Odată finalizată o schemă logică 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. În urma lansării analizei va apărea utilitarul de analiză combinațională în fereastra **Combinational Analysis**, care are mai multe tab-uri:

- **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 intrărilor și ieșirilor detectate în cadrul circuitului. Apăsând (cu mouse-ul) 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 cu mouse-ul în celulele diagramei valorile oscilează între 0, 1 și x, iar grupările și minimizarea se actualizează automat (Figura 3. 3). Dacă butonul **Set As Expression** este activ, atunci trebuie apăsat pentru ca expresia minimizată să fie luată în considerare.

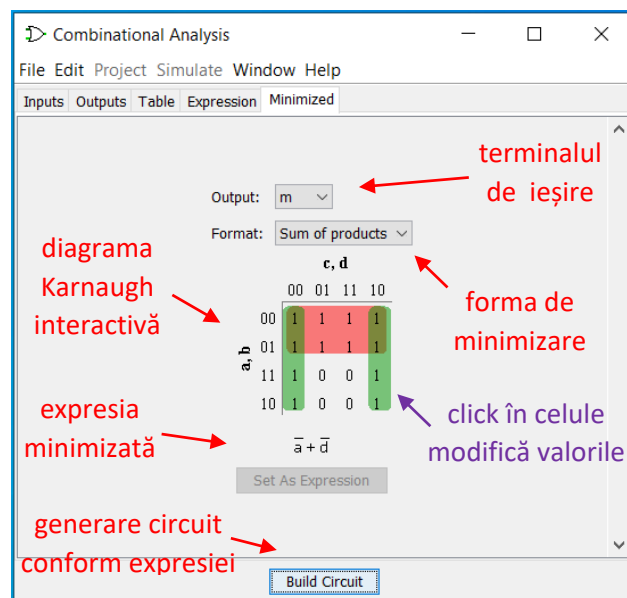


Figura 3. 3 Analiza combinațională a circuitului din Figura 3. 4

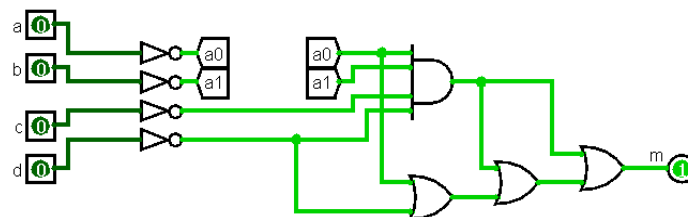


Figura 3. 4 Schemă logică complexă a cărei expresie se poate simplifica cu utilitarul de analiză combinațională

Notă: Analiza circuitului funcționează corect 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 un circuit care implementează funcțiile logice curente sau modificate de utilizator, în

forma minimizată. Se va solicita numele noului circuitul și apoi va fi inclus în librăria proiectului. De exemplu circuitul generat pentru expresia din Figura 3. 3 este redat în Figura 3. 5.

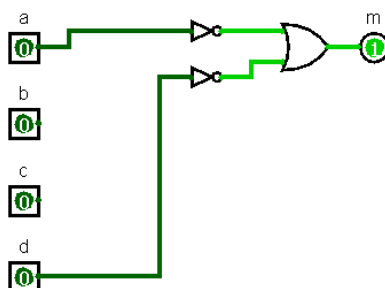


Figura 3. 5 Schema logică a circuitului generat cu comanda **Build Circuit** pentru expresia minimizată din Figura 3. 3. Acest circuit este echivalent cu cel din Figura 3. 4

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 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** sau prin tabelă 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 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 3. 6).

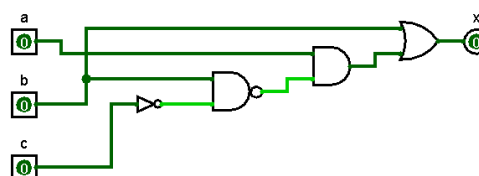
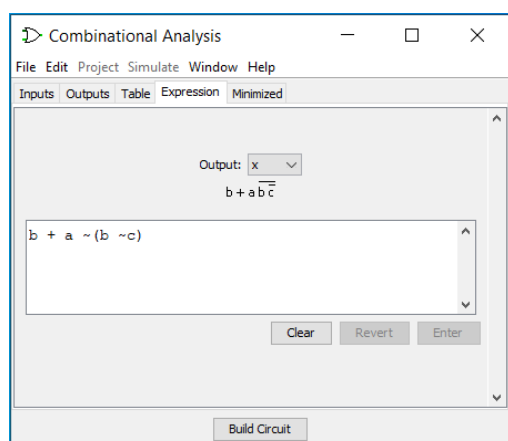



Figura 3. 6 Introducerea unei expresii în utilitarul de analiză combinațională (stânga).
Circuitul generat corespunzător expresiei (dreapta)

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.

3.2.3 Simularea secvențială în Logisim

Circuitele secvențiale sunt circuite cu stare internă. În majoritatea cazurilor aceste circuite cel puțin o intrare rezervată pentru semnalul de tact (clock), adică valoarea lui oscilează periodic între "0 logic" și "1 logic" cu o frecvență bine stabilită. Un circuit secvențial sincron își poate schimba starea la fiecare trecere din 0 în 1 sau la fiecare trecere din 1 în 0, în funcție de caz. Pentru simularea cu aceste circuite utilitarul Logisim pune la dispoziție terminalul de tip  Clock din librăria *Wiring*. Ca și exemplu de utilizare a acestuia se poate considera exemplul numărătorului sincron 74162 din librăria TTL a cărui stare numără crescător în baza 2 pe 4 biți în intervalul $0000_2 - 1111_2$. Circuitul de simulare a funcționării acestuia este redat în Figura 3. 7.

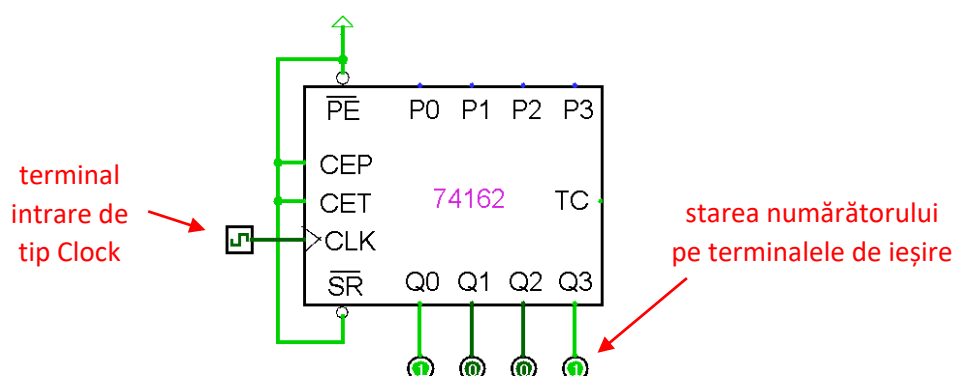


Figura 3. 7 Simularea funcționării numărătorului binar sincron 74162. Ordinea biților valorilor de ieșire de la cel mai semnificativ la cel mai puțin semnificativ este $Q_3Q_2Q_1Q_0$

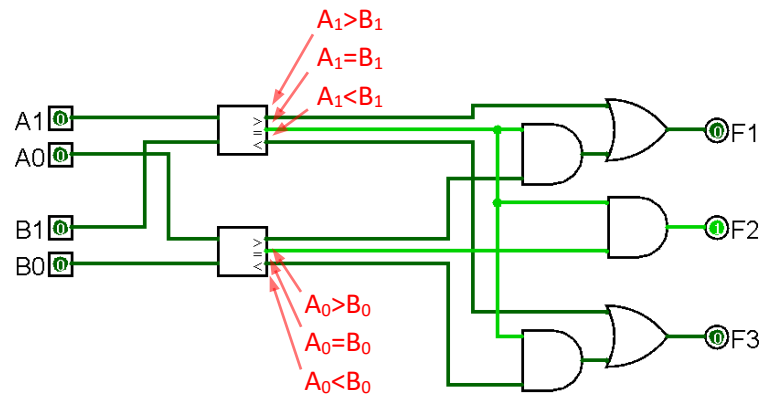
Configurările legate de semnalele de tip Clock se află în secțiunea **Simulate** a meniului. Astfel pornirea sau oprirea oscilațiilor periodice în timpul simulării a se face cu comanda **Ticks Enabled** sau cu combinația de taste **Ctrl+K**. Frecvența de oscilație se poate configura cu opțiunea **Tick Frequency**; de obicei o frecvență de 1Hz sau 2Hz este potrivită din punct de vedere vizual. Dacă oscilația este oprită, atunci se poate comanda (în mod repetat) câte o singură oscilație cu opțiunea **Tick Once** sau **Ctrl+T**, obținându-se astfel o simulare pas cu pas.

3.3 Desfășurarea lucrării

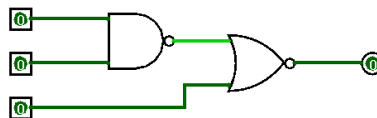
1. Creați un proiect nou în Logisim și încărcați librăria TTL. Vizualizați lista circuitelor din componența sa.
2. Realizați schema din figura următoare care reprezintă un comparator pe 2 biți folosind comparatoare pe 1 bit din librăria *Arithmetic* și alte porți logice fundamentale necesare. Testați 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:

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



3. Creați o nouă schemă cu circuitul din figura următoare și folosiți utilitarul de analiză combinațională pentru a genera circuitul echivalent obținut prin minimizare cu tab-ul **Minimize**.



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

$$f1 = \sim a + b + \sim c \quad f2 = \sim(a + b) (a + c) \quad f3 = \sim a + \sim(\sim b c)$$

5. Implementați circuitul secvențial din Figura 3. 7 și simulați funcționarea acestuia urmărind secvența de numărare binară pe 4 biți.