

Seminar 0x02

Cristian Rusu

1 Scopul seminarului

În acest seminar vom rezolva niște probleme care implică:

- logică și circuite combinaționale;
- logică și circuite secvențiale;
- mașini cu stări.

2 Exerciții

1. Calculați tabelele de adevăr pentru următoarele expresii logice:

- | | |
|---------------------------|---------------------------------------------------------|
| (a) $X = A + BC$ | (f) $X = AB(C \oplus D)$ |
| (b) $X = A(B + D)$ | (g) $X = A\bar{B}C\bar{D}$ |
| (c) $X = (A + B)(A + C)$ | (h) $X = (A\bar{B} + \bar{A}B) + (C\bar{D} + \bar{C}D)$ |
| (d) $X = \bar{A}(B + C)D$ | (i) $X = (\bar{A}\bar{B} + AB)(\bar{C}\bar{D} + CD)$ |
| (e) $X = A\bar{B}(C + D)$ | (j) $X = A(A \oplus B)(B \oplus C)D$ |

2. Desenați circuitele digitale pentru următoarele expresii logice:

- | | |
|-------------------------------|----------------------------------|
| (a) $X = (A + B)C$ | (d) $X = !(A + B)(C + D)\bar{D}$ |
| (b) $X = A + BC + \bar{D}$ | (e) $X = A \oplus B \oplus CD$ |
| (c) $X = AB + \bar{A}\bar{C}$ | (f) $X = A\bar{B} \oplus CB$ |

3. Legile de Morgan sunt: $\overline{A + B} = \bar{A}\bar{B}$ și $\overline{AB} = \bar{A} + \bar{B}$. Cerințe:

- verificați cele două legi;
- desenați circuitele pentru: $\overline{A + B}$, $\bar{A}\bar{B}$, \overline{AB} și $\bar{A} + \bar{B}$;
- folosind și cele două legi calculați/simplificați expresiile:
 - $\overline{\bar{A} + \bar{B}}$
 - $\overline{\bar{A}\bar{B}}$
 - $\overline{A + B + C}$, generalizați la N variabile digitale A_i
 - \overline{ABC} , generalizați la N variabile digitale A_i
 - $\overline{(A + B)}\bar{A}\bar{B}$
 - $\bar{A}\bar{B}(\bar{A} + \bar{B})$
 - $\overline{(A + B)}(\bar{A} + \bar{B})$
 - $\bar{A}\bar{B}(\overline{AB})$
 - $C + \bar{C}\bar{B}$
 - $\bar{A}\bar{B}(\bar{A} + B)(\bar{B} + \bar{B})$
 - $\overline{(\bar{A}\bar{B})}(\bar{B} + C)$
 - $\overline{(A + \bar{B})}(\bar{B}C)$
 - $\overline{(\bar{A} + C)}(\overline{AB})$

4. Simplificați următoarele expresii logice:

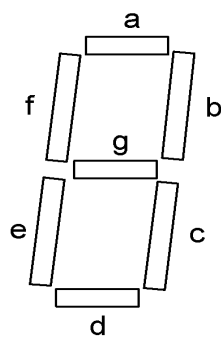
- | | |
|----------------------------------------|----------------------------------------------------------------|
| (a) $A + 0$ | (n) $\bar{A} + B\bar{A}$ |
| (b) $\bar{A}0$ | (o) $(D + \bar{A} + B + \bar{C})B$ |
| (c) $A + \bar{A}$ | (p) $(A + \bar{B})(A + B)$ |
| (d) $A + A$ | (q) $C(C + CD)$ |
| (e) $A + AB$ | (r) $A(A + AB)$ |
| (f) $A + \bar{A}B$ | (s) $\overline{\bar{A} + \bar{A}}$ |
| (g) $A(\bar{A} + B)$ | (t) $\overline{\bar{A} + \bar{A}}$ |
| (h) $AB + \bar{A}B$ | (u) $D + (D\bar{C}BA)$ |
| (i) $(\bar{A} + \bar{B})(\bar{A} + B)$ | (v) $\bar{D}(\overline{DBCA})$ |
| (j) $A(A + B + C + \dots)$ | (w) $AC + \bar{A}B + BC$ |
| (k) fie $f(A, B, C) = A + B + C$ | (x) $(A + C)(\bar{A} + B)(B + C)$ |
| i. $f(A, B, AB)$ | (y) $\bar{A} + \bar{B} + AB\bar{C}$ |
| ii. $f(A, B, \bar{A}\bar{B})$ | (z) $(A + C)(AD + A\bar{D}) + AC + C$ |
| iii. $f(A, B, \bar{A}\bar{B})$ | (N ₀) $\bar{A}(A + B) + (B + AA)(A + \bar{B})$ |
| (l) $A + A\bar{A}$ | (N ₁) $(A + B)^2 + (A + B)^3 + A + 3\bar{A} + A^3$ |
| (m) $AB + A\bar{B}$ | |

5. La curs am vorbit despre multiplexare. Cerințe:

- presupunem că avem un MUX cu două intrări binare și un semnal de un bit de selecție. Scrieți relația ieșire-intrare pentru acest MUX. Desenați și circuitul care implementează acest MUX;
 - un MUX cu două intrări binare și un semnal de selecție binar (1 bit) poate simula porțile NOT, OR, și AND. Desenați MUX-urile (și scrieți tabelul de adevăr) pentru fiecare poartă;
 - am discutat la curs că un MUX poate implementa un “if”. Descrieți un circuit care implementează: $Y = S ? \text{foo}(A) : \text{bar}(B)$, unde $\text{foo}()$ și $\text{bar}()$ sunt două funcții (în cazul nostru, logice). Există o diferență importantă între acest circuit și felul în care evaluăm un “if” într-un limbaj de programare?
 - într-un limbaj de programare, dacă avem un vector (array) \mathbf{x} de dimensiune N cu valori binare putem accesa elementul de pe poziția i apelând x_i . Cum implementăm acest acces folosind un MUX? Câte intrări are acest MUX? Care este dimensiunea (câți biți) semnalului de selecție?
 - avem nevoie să implementăm un MUX cu 4 intrări dar avem la dispoziție doar circuite MUX cu 2 intrări. Cum implementăm MUX-ul cu 4 intrări? Câte MUX-uri cu 2 intrări avem nevoie? Dacă vrem să implementăm un MUX cu N intrări câte MUX-uri de două intrări avem nevoie?
6. În Anexa 1 aveți un display pentru o cifră care utilizează 7 segmente (*7-segment display*). Aveți 10 butoane la dispoziție, scrieți un circuit combinațional (tabelul de adevăr doar) care afișează pe display cifra corespunzătoare butonului apăsat.
7. Am discutat la curs despre porți logice și cum aceste pot fi folosite pentru a implementa circuite de adunare. Cealaltă operație pe biți foarte importantă despre care nu am discutat este operația de deplasare (shift). Cerințe:
- se dă un număr x pe 4 biți, realizați deplasarea la dreapta cu 2 biți: deplasare normală, deplasare aritmetică, și deplasare circulară. Repetați operațiile deplasând la stânga;

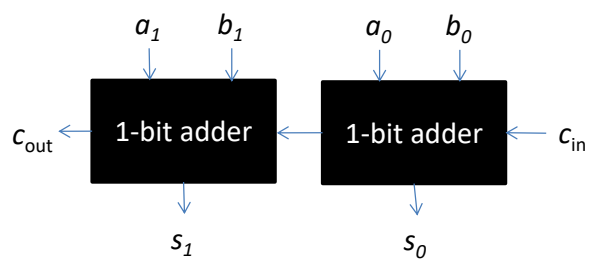
- (b) implementați un circuit care realizează deplasarea unei variabile x pe N biți la dreapta cu d poziții utilizând un singur MUX (cu oricâte intrări sunt necesare);
- (c) puteți implementa problema anterioară mai bine? (hint: deplasare 5 \equiv deplasare 4, apoi cu 1)
8. În Anexele 2 și 3 aveți două circuite de adunare (pe 2 biți și 4 biți, respectiv). Pentru fiecare, scrieți expresia logică între ieșirea c_{out} și intrarea c_{in} . Verificați cursul și folosiți variabilele binare p și g în expresiile logice pe care le calculați.
9. Considerăm un circuit de adunare pe N biți pentru numere întregi (similar celui din Anexa 3, dar generalizat la N). Scrieți tabelul de adevăr pentru un circuit care detectează *overflow/underflow*.
10. Un semnal digital A poate lua valori $\{0, 1\}$ în timp iar noi vrem să detectăm dacă semnalul are valoarea 010 la un moment dat. Dacă această secvență de biți este detectată în A atunci o variabilă Y este setată la 1, altfel această variabilă este 0. Cerințe:
- (a) desenați diagrama de stări și tranzițiile între aceste stări;
- (b) cum arată circuitul secvențial asociat diagramei realizate anterior?
- (c) calculați tabelul de stări pentru circuitul secvențial;
- (d) scrieți expresiile pentru logica combinațională.
11. În Anexa 4 aveți un circuit secvențial și o diagramă cu evoluția în timp a semnalelor (aici, C este CLK). Calculați evoluția în timp a ieșirii Z. (hint: prima dată, redesenați circuitul folosind porți fundamentale NOT, OR, și AND - va fi mai ușor să citiți circuitul)
12. La curs am vorbit despre un circuit secvențial counter pe 2 biți și am scris reprezentarea pe stare / tabel de adevăr. Cerințe:
- (a) scrieți folosind funcții logice relația dintre “starea anterioară” și “starea viitoare”. Desenați și circuitul secvențial care realizează counter-ul pe 2 biți. (hint: fiecare stare este pe 2 biți, notați acești biți cu q_1 și q_0);
- (b) extindeți counter-ul să funcționeze pe 3 biți;
- (c) codul Gray este un cod binar care are proprietatea că de la un simbol la altul diferența este de un singur bit care se schimbă (de exemplu, pe 3 biți codul Gray este 000, 001, 011, 010, 110, 111, 101 și 100). Modificați counter-ul pe 3 biți creat anterior ca să numere în cod Gray.
13. Implementați algoritmul lui Euclid pentru CMMDC folosind un circuit secvențial.
14. Implementați un algoritm care verifică dacă un număr este prim folosind un circuit secvențial.
15. Considerăm două variabile binare A și B pe 3 biți fiecare. Scrieți funcții logice care au valoarea “1” doar dacă:
- (a) A conține un singur bit “0”; (d) $A == B$;
- (b) A conține un număr impar de biți “0”; (e) $A < B$, dacă A și B sunt numere naturale;
- (c) A este un număr mai mic de 4, în baza zecimală; (f) $A < B$, dacă A și B sunt numere întregi scrise în complement față de doi.
16. Arătați că porțile logice NOR și NAND, respectiv, sunt universale (adică cu fiecare dintre ele putem implementa cele 3 porți fundamentale NOT, OR, și AND).
17. Avem un automat care vinde snack-uri. Acesta acceptă monede de 10 bani, 50 bani și 1 leu. Când o monedă este introdusă în mașină un semnal digital este activat în funcție de monedă (A, B, și respectiv C) și un semnal digital general (G) este activat (o monedă a fost introdusă). Există și un semnal de reset (RST) care reduce la zero suma introdusă. Vrem un circuit care activează o ieșire digitală X atunci când suma introdusă depășește 60 de bani.

Anexa 1



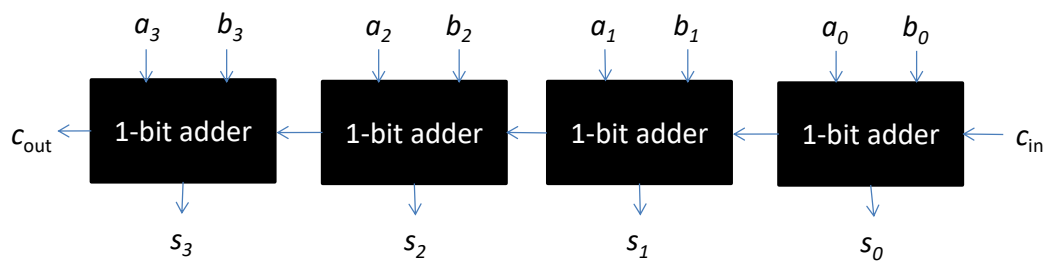
Display cifră cu 7 segmente.

Anexa 2



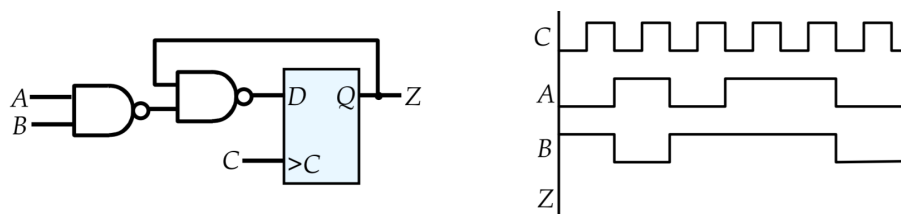
Circuit de adunare pe 2 biți.

Anexa 3



Circuit de adunare pe 4 biți.

Anexa 4



Evoluția semnalelor pentru un circuit secvențial.