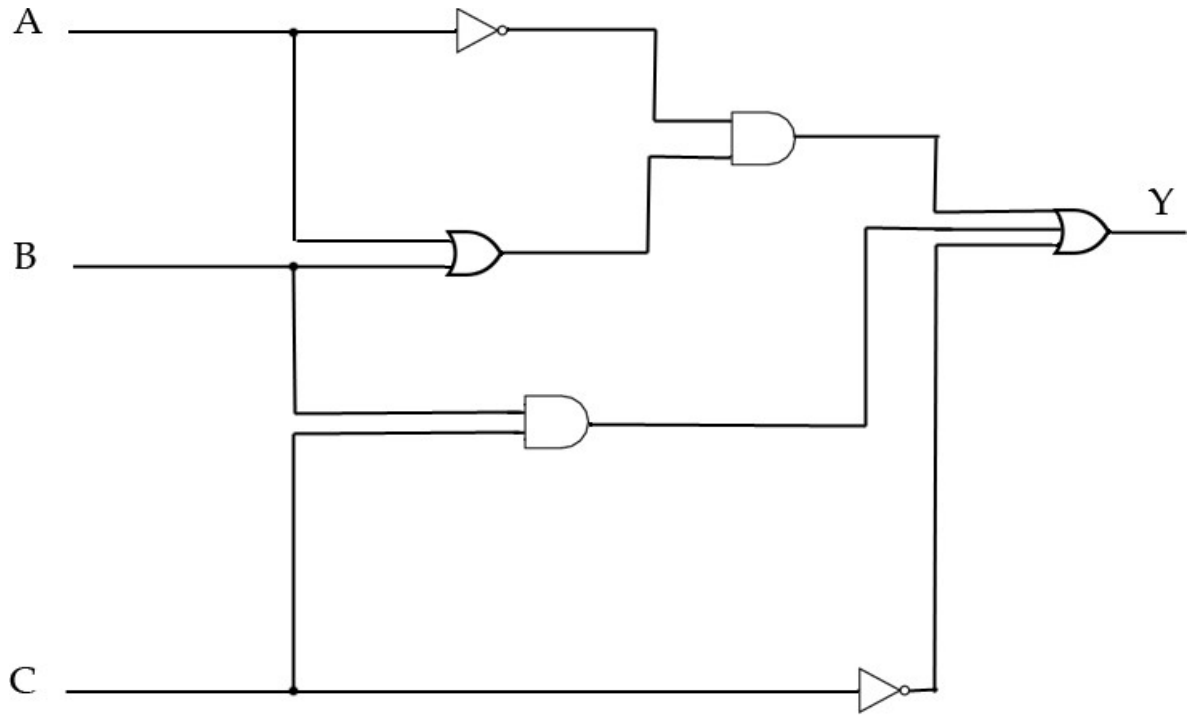
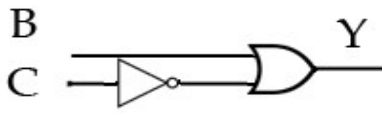
Esercitazione 5 – Architettura dei Sistemi di Elaborazione - 23/01/2017

Algebre di Boole e funzioni logiche

1. **Si consideri il circuito logico in figura. Si determini la funzione logica che implementa e la si semplifichi. Disegnare, poi il nuovo circuito.**



Allora, è molto semplice vedere quale sia la soluzione. Basta risolvere, una per volta, le porte logiche in figura. Quindi, avremo:
$Y = (\overline{A}\,(A + B)) + (BC) + \overline{C}$
Ora, risolviamo la funzione tramite le regole di distributività.
$Y = \overline{A}\,A + \overline{A}B + (BC) + \overline{C}$
Ora, $\overline{A}\,A$ è una contraddizione, e quindi ha valore 0 (nullo per l'operazione OR). Allo stesso tempo, $(BC) + \overline{C}$, si può risolvere nel solo $B + \overline{C}$ grazie alla regola d'assorbimento. Quindi si
ottiene:
$Y = \overline{A}B + B + \overline{C}$
Ora si raggruppa B, si ottiene:
$Y = B(\overline{A} + 1) + \overline{C}$
Ma 1 è l'elemento assorbente per l'operazione OR, quindi tutta l'espressione $(\overline{A} + 1)$ vale 1. Allo stesso tempo, 1 è l'elemento neutro per l'espressione AND, quindi $B \cdot 1$ vale B.
L'espressione finale, dunque, è:
$Y = B + \overline{C}$
Ora, il circuito che lo implementa è quello mostrato in figura.



2. **Data la funzione logica $Y= BC\overline{D} + CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C$, scriverla in forma normale congiuntiva, ed eventualmente minimizzarla. Disegnare il circuito corrispondente.**

In primo luogo, eseguiamo la minimizzazione tramite le regole delle algebre di Boole. Inizialmente, raggruppiamo C per tutte le espressioni della funzione, ottenendo:
$Y = C(B\overline{D} + D + A\overline{B}\,\overline{D} + \overline{A}\,\overline{B})$
Ora raggruppiamo $\overline{D}$ per tutte le espressioni che lo contengono, ottenendo:
$Y = C(\overline{D}\,(B + A\overline{B}) + D + \overline{A}\,\overline{B})$
Ora si utilizza la stessa regola d'assorbimento usata all'esercizio precedente.
$Y = C(\overline{D}\,(B + A) + D + \overline{A}\,\overline{B})$
Si risolve la parentesi relativa all'elemento $\overline{D}$.
$Y = C(\overline{D}\,B + \overline{D}\,A + D + \overline{A}\,\overline{B})$
Assorbimento tra $\overline{D}\,A + D$
$Y = C(\overline{D}\,B + A + D + \overline{A}\,\overline{B})$
Assorbimento tra $\overline{D}\,B + D$
$Y = C(B + A + D + \overline{A}\,\overline{B})$
Assorbimento tra $B + \overline{A}\,\overline{B}$
$Y = C(B + A + D + \overline{A}\,)$
Ma $A + \overline{A}$ è una tautologia, che dunque vale 1. Ora, 1 è l'elemento assorbente per l'operazione OR, quindi tutta l'espressione dentro la parentesi può essere semplificata in 1. Si ottiene dunque:
$Y = C$.
Il circuito, è, dunque, banale.
Il passaggio in CNF, a dirla tutta, è abbastanza noioso. E, soprattutto, orribile da scrivere qua sopra. Quindi ve lo lascio scritto orribilmente a mano. Ad ogni modo, l'idea è quella di applicare una doppia negazione alla funzione, risolvere quella interna utilizzando le leggi di DeMorgan e poi verificare un po' di contraddizioni e idempotenze. Alla fine si ri-applica DeMorgan e si ottiene la CNF.

$$\neg\left(\overline{BC\bar{D}}\cdot\overline{C\bar{D}}\cdot\overline{A\bar{B}C\bar{D}}\cdot\overline{\bar{A}\bar{B}C}\right)=\neg\left(\left(\bar{B}+\bar{C}+D\right)\left(\bar{C}+\bar{D}\right)\left(\bar{A}+B+\bar{C}+D\right)\left(A+B+\bar{C}\right)\right)=$$

$$=\neg\left(\left(\bar{B}\bar{C}+\bar{B}\bar{D}+\bar{C}+D\bar{D}\right)\left(\bar{A}+B+\bar{C}+D\right)\left(A+B+\bar{C}\right)\right)=$$

$$=\neg\left(\left(\bar{B}\bar{C}+\bar{B}\bar{D}+\bar{C}\right)\left(\bar{A}+B+\bar{C}+D\right)\left(A+B+\bar{C}\right)\right)=$$

$$=\neg\left(\left(\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}\bar{D}+\bar{A}\bar{C}+B\bar{B}\bar{C}+B\bar{B}\bar{D}+B\bar{C}+\bar{B}\bar{C}\bar{C}+\bar{B}\bar{C}\bar{D}+\bar{C}\bar{C}+\bar{B}\bar{C}D+\bar{B}\bar{D}D+B\bar{C}D\right)\left(A+B+\bar{C}\right)\right)=$$

$$=\neg\left(\left(\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}\bar{D}+A\bar{C}+B\bar{C}+\bar{B}\bar{C}+\bar{B}\bar{C}\bar{D}+\bar{C}+\bar{B}\bar{C}D+\bar{C}D\right)\left(A+B+\bar{C}\right)\right)=$$

$$=\neg\left(A\bar{A}\bar{B}\bar{C}+\bar{A}AB\bar{D}+A\bar{A}\bar{C}+AB\bar{C}+A\bar{B}\bar{C}+A\bar{B}\bar{C}\bar{D}+A\bar{C}+A\bar{B}\bar{C}D+A\bar{C}D+\right.$$
$$\bar{A}B\bar{B}\bar{C}+B\bar{A}\bar{B}\bar{D}+A\bar{A}\bar{C}+AB\bar{C}+A\bar{B}\bar{C}+A\bar{B}\bar{C}\bar{D}+A\bar{C}$$
$$+\bar{A}B\bar{C}+B\bar{C}+B\bar{B}\bar{C}+B\bar{B}\bar{C}\bar{D}+B\bar{C}+B\bar{B}\bar{C}D+B\bar{C}D+\bar{A}\bar{B}\bar{C}+$$
$$\left.+\bar{A}\bar{B}\bar{C}\bar{D}+\bar{A}\bar{C}+B\bar{C}+\bar{B}\bar{C}+\bar{B}\bar{C}\bar{D}+\bar{C}+\bar{B}\bar{C}D+\bar{C}D\right)=$$

$$=\neg\left(AB\bar{C}+A\bar{B}\bar{C}+A\bar{B}\bar{C}\bar{D}+A\bar{C}+A\bar{B}\bar{C}D+A\bar{C}D+\bar{A}B\bar{C}+B\bar{C}+\right.$$
$$\left.+B\bar{C}D+\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}\bar{D}+\bar{A}\bar{C}+\bar{B}\bar{C}\bar{D}+\bar{C}+\bar{B}\bar{C}D+\bar{C}D\right)=$$

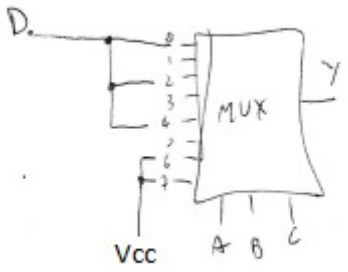
$$=\left(\bar{A}+\bar{B}+\bar{C}\right)\left(\bar{A}+B+C\right)\left(\bar{A}+B+C+D\right)\left(\bar{A}+C\right)\left(\bar{A}+B+C+D\right)\left(\bar{A}+C+\bar{D}\right)\cdot$$
$$\cdot\left(A+\bar{B}+C\right)\cdot\left(\bar{B}+C\right)\cdot\left(\bar{B}+C+\bar{D}\right)\cdot\left(A+B+C\right)\left(A+B+C+D\right)\cdot$$
$$\cdot\left(A+C\right)\left(B+C+D\right)\left(C\right)\left(B+C+\bar{D}\right)\left(C+\bar{D}\right)\qquad OK$$

3. Data la formula logica ¬($\overline{A}$CB + $\overline{A}$C$\overline{B}$ + $\overline{A}$ $\overline{D}$ + $\overline{B}$CD + $\overline{B}$C + $\overline{B}$ $\overline{D}$). Dimostrare se è equivalente alla formula AB+$\overline{C}$D. Disegnare il circuito che la implementa, utilizzando un multiplexer a 8 dati in input e 3 input di controllo.

Ovviamente, come suggerito dalla domanda stessa, l'espressione è equivalente. Quindi, costruiremo il mux relativo a questa seconda espressione.
Y = ¬($\overline{A}$CB + $\overline{A}$C$\overline{B}$ + $\overline{A}$ $\overline{D}$ + $\overline{B}$CD + $\overline{B}$C + $\overline{B}$ $\overline{D}$)
Ora raccoglieremo $\overline{A}$C e $\overline{B}$C per le sotto-espressioni che le contengono
Y = ¬($\overline{A}$C(B + $\overline{B}$) + $\overline{B}$C(D + 1) + $\overline{B}$ $\overline{D}$ + $\overline{A}$ $\overline{D}$)

Sappiamo che (B + $\overline{B}$) è una tautologia (e quindi vale 1) e che 1 è anche l'elemento assorbente per l'operazione +, quindi (D + 1) vale 1. Allo stesso tempo, 1 è l'elemento neutro per l'operazione AND e quindi può essere omesso. Otteniamo:
Y = ¬($\overline{A}$C + $\overline{B}$C+ $\overline{B}$ $\overline{D}$ + $\overline{A}$ $\overline{D}$)
Ora raggruppiamo $\overline{A}$ e $\overline{B}$ per le sotto-espressioni che le contengono. Abbiamo, quindi,
Y = ¬($\overline{A}$ (C + $\overline{D}$) + $\overline{B}$ (C + $\overline{D}$))
Ora si raggruppa per (C + $\overline{D}$)
Y = ¬(($\overline{A}$ +$\overline{B}$)(C + $\overline{D}$))
Usiamo ora la legge di DeMorgan
Y = ¬($\overline{A}$ +$\overline{B}$) + ¬(C + $\overline{D}$)
Usiamo nuovamente la legge di DeMorgan
Y = AB + $\overline{C}$D

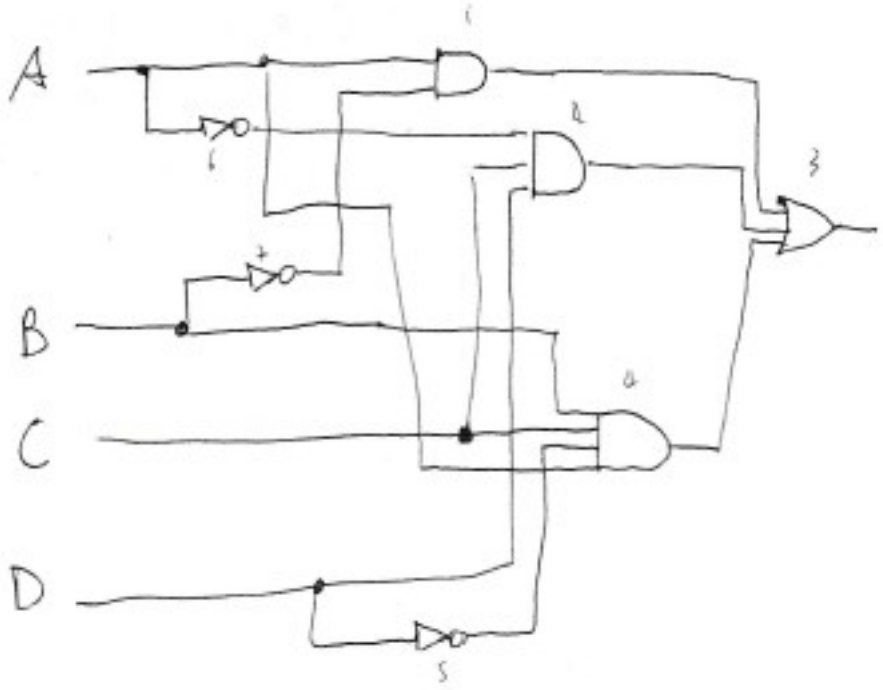
La costruzione del mux è spiegata più nel dettaglio nella soluzione dell'esercitazione numero 2, che vi consiglio di leggere. Quindi, qui vi espongo solo il mux corretto.



Una soluzione alternativa, e forse che piace di più al professore (e quindi vi consiglio caldamente di usarla, in sede d'esame) è fornita nella dispensa mux_con_chip_select, che trovate nella stessa cartella Google Drive. Andatevela a leggere.

4. **Considerato il circuito corrispondente ad A$\overline{B}$ + $\overline{A}$CD + ABC$\overline{D}$, costruire un circuito equivalente, ma che utilizzi almeno 10 porte logiche in più.**

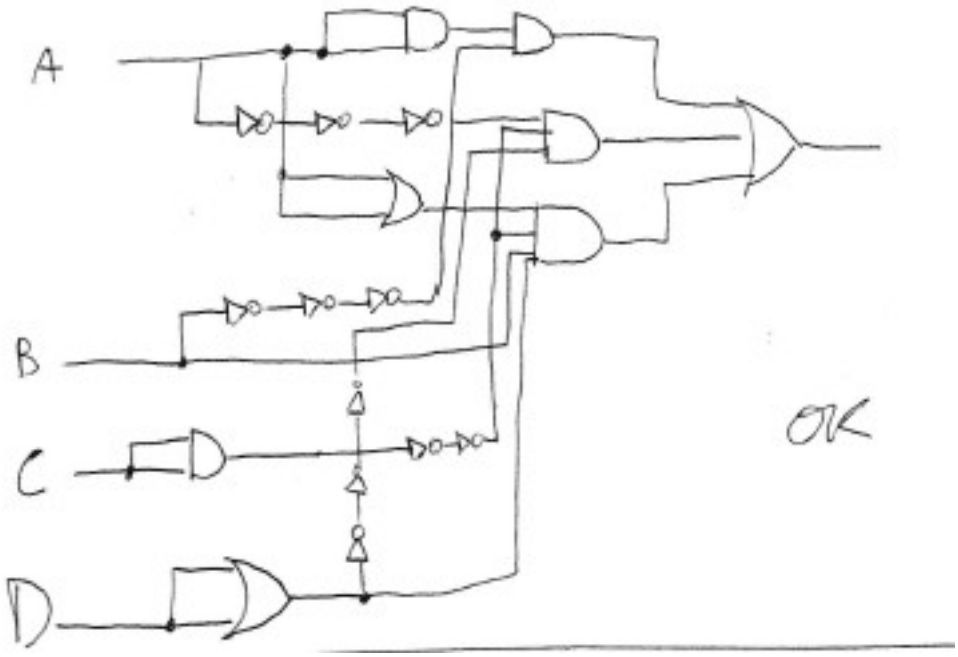
Per iniziare, il circuito che implementa la funzione descritta è il seguente:



Ora, per aggiungere porte senza modificare la valenza logica del circuito, ci basiamo su alcune semplici idee:

- $A = \overline{\overline{A}}$ [due porte logiche in più]
- $A = A + A$ [una porta logica in più]
- $A = AA$ [una porta logica in più]
- $A + B + C = (A+B) + C$ [una porta logica in più]

Partendo da queste idee, costruiamo un circuito equivalente (ovviamente, ne esistono infiniti).



Algoritmi di paginazione

1. **Utilizzando l'algoritmo di paginazione Aging (con contatore a 8 bit), si considerino le seguenti 4 pagine, con valore di bit R, rispettivamente, 1000. Ai cicli successivi, i valori**

sono 0111, 0011, 1101, 1100, 0001, 1010 e 1110. Si forniscano i valori dei 4 contatori dopo l'ultimo intervallo, specificando quale pagina viene spostata su disco.

Allora, in primo luogo mostriamo come funziona l'algoritmo Aging:
Si considera la tabella coi contatori, scrivendo i valori dei bit R. Ad ogni nuova scrittura, c'è uno shift verso destra, ottenendo dunque la seguente soluzione:

Pag								
P_0	1	1	0	1	1	0	0	1
P_1	1	0	0	1	1	0	1	0
P_2	1	1	0	0	0	1	1	0
P_3	0	0	1	0	1	1	1	0

Ora, la pagina che sarà swappata è quella con valore binario minore, quindi la pagina denominata P_3 sarà spostata su disco.

2. **Usando l'algoritmo di paginazione LRU (least recently userd), nel caso in cui la macchina abbia 4 pagine fisiche, e nel caso in cui le pagine siano usate nell'ordine:**
   0 1 2 3 1 0 3 2
   Stabilire quale pagina verrà spostata su disco, tenendo presente la matrice delle pagine.

   L'idea di fondo dell'algoritmo è quella di estrarre la pagina usata più lontano nel passato. Come gli altri esercizi dell' EX4, trovate la descrizione estesa sul libro "I moderni sistemi operativi", alle pagine 197 e successive, al capitolo 4.4. Algoritmi di rimpiazzamento delle pagine.
   L'algoritmo funziona come segue:
   - Si crea una tabella della dimensione del numero delle pagine per ogni dimensione (4x4 nel nostro caso)
   - La si inizializza completamente a 0
   - Ogni volta che una pagina viene usata, si effettua la seguente procedura:
     - Si pone a 1 tutta la riga corrispondente alla pagina
     - Si pone a 0 tutta la colonna corrispondente alla pagina
   - Si estrae la pagina corrispondente al numero binario (sulla riga) minore.

   Quindi, nel nostro caso, avremo il seguente svolgimento:

	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

	0	1	2	3
0	0	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

	0	1	2	3
0	0	0	0	1
1	1	0	0	0
2	1	1	0	1
3	0	0	0	0

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	0	0	0
1	1	0	1	1
2	1	0	0	0
3	1	0	1	0

	0	1	2	3
0	0	1	1	1
1	0	0	1	1
2	0	0	0	0
3	0	0	1	0

	0	1	2	3
0	0	1	1	0
1	0	0	1	0
2	0	0	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	0	0
1	0	0	0	0

2	1	1	0	1
3	1	1	0	0

Viene, quindi, estatta la pagina 1.