Arhitectura sistemelor de calcul

- Prelegerea 6 -

Extensii. Cicluri. Clasificare

Ruxandra F. Olimid

Facultatea de Matematică şi Informatică

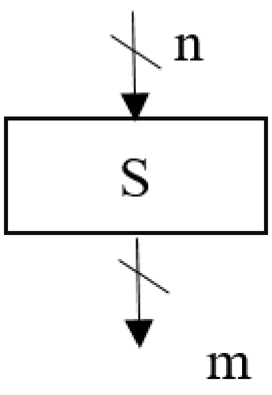
Universitatea din Bucureşti

Cuprins

Sisteme digitale

- Considerăm un element teoretic de bază: *sistemul digital*
- Fie V un alfabet finit şi n, m 2 numere naturale. Se numeşte *sistem digital* o structură $S = (X, Y, f)$, unde $X = V^n, Y = V^m, f: X \rightarrow Y$
- Funcţia f se numeşte *funcţie de transfer*

- Reprezentarea grafică a sistemelor digitale este următoarea:



[ASC]

Sisteme digitale

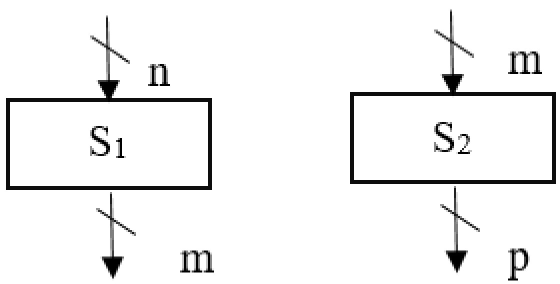
- Vom lucra cu *sisteme digitale binare,* pentru care $V = \{0,1\}$

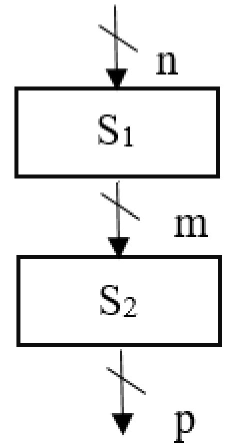
- Un exemplu sunt circuitele combinaţionale, corespunzătoare funcţiilor booleene deja studiate

Extensii

- Fie $S_1 = (X_1, Y_1, f_1)$, $S_2 = (X_2, Y_2, f_2)$ 2 sisteme digitale, unde $X_1 = \{0,1\}^n$, $Y_1 = X_2 = \{0,1\}^m$, $Y_2 = \{0,1\}^p$.

Se numeşte *extensie serială* sistemul $S = (X, Y, f)$, unde $X = X_1, Y = Y_2$ şi $f : X \to Y, f = f_2 \circ f_1$

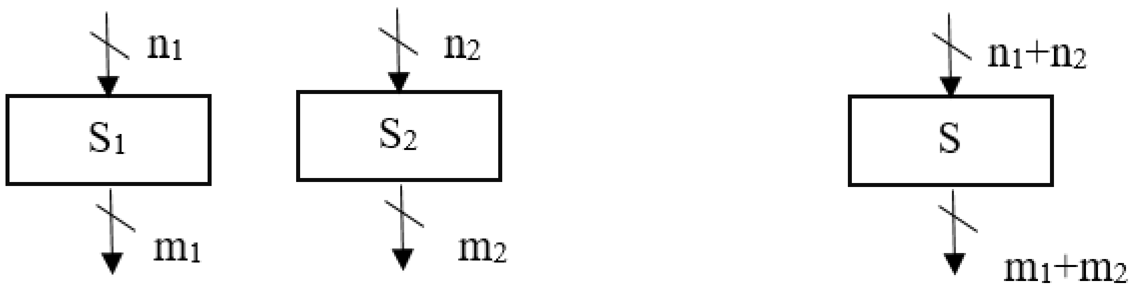




Extensii

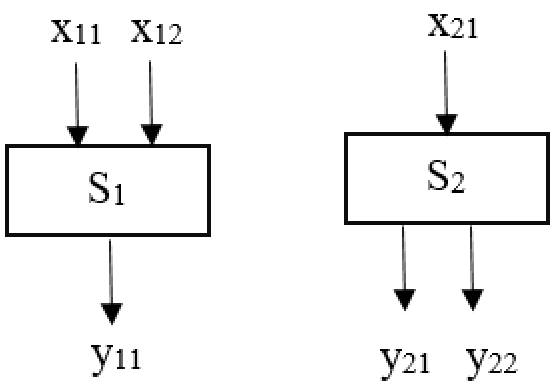
- Fie $S_1 = (X_1, Y_1, f_1)$, $S_2 = (X_2, Y_2, f_2)$ 2 sisteme digitale.

Se numeşte *extensie paralelă* sistemul $S = (X_1 \times X_2, Y_1 \times Y_2, f)$, unde $f: X_1 \times X_2 \rightarrow Y_1 \times Y_2, f(x, y) = (f_1(x), f_2(y))$



Exemplu

- Fie sistemele digitale $S_1 = (X_1, Y_1, f_1)$, $S_2 = (X_2, Y_2, f_2)$ definite mai jos:

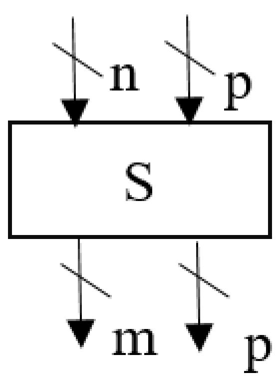


- *Întrebare:* Care sunt extensiile serială, respectiv paralelă?

Cicluri

- Pentru creşterea gradului de complexitate al sistemelor digitale se defineşte noţiunea de *ciclu*.

- Fie sistemul digital $S_1 = (X \times X_1, Y \ \times Y_1, h)$, unde $X_1 = Y_1, h = (f, f_1)$
$f: X \times X_1 \to Y; f_1: X \times X_1 \to Y_1$

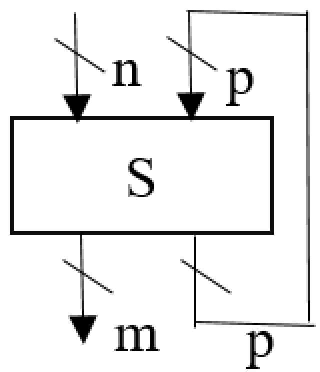


- Considerăm sistemul digital şi mulţimile $X, X_1 = Y_1, Y$ exprimate $n, p,$ respectiv m biţi

Cicluri

- Prin conectarea lui Y_1 la $X_1 (X_1 = Y_1)$ se obţine un ciclu

- Noul sistem devine $= (X, Y, g)$, unde g: $X \to Y, g(x) = f(x, f_1(x, y))$



- f_1 se numeşte *funcţie de tranziţie* şi verifică definiţia recursivă $y = f_1(x, f_1(x, y))$

Cicluri

- Apare o variabilă internă (pe p biţi), care ia valori în mulţimea $Q = X_1 = Y_1$

- Q se numeşte *stare internă* sau pe scurt *stare*

- În funcţie de stare, funcţiile devin

$$f: X \times Q \to Y;\ f_1: X \times Q \to Q$$

- f se numeşte *funcţie de tranziţie ;* f_1 se numeşte *funcţie de stare*

Clasificare

- Considerăm un sistem fără cicluri un *0-DS*

- Se defineşte recursiv un system *n-DS* ca un sistem (n-1)-DS la care se adaugă un ciclu peste toate cele n-1 cicuri existente

- ✓ **0-DS** : *circuite combinaţionale* (fără autonomie)
- ✓ **1-DS**: *circuite cu memorie* (prezintă autonomie pe spaţiul stărilor)
- ✓ **2-DS**: *automate finite* (prezintă autonomie pe tipul de comportare)
- ✓ **3-DS**: *procesoare* (prezintă autonomie pe interpretarea stărilor interne)
- ✓ *calculatoare*

Referințe bibliografice

[AAT] A. Atanasiu, Arhitectura calculatorului

Schemele [Xilinx - ISE] au fost realizate folosind

http://www.xilinx.com/tools/projnav.htm