НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

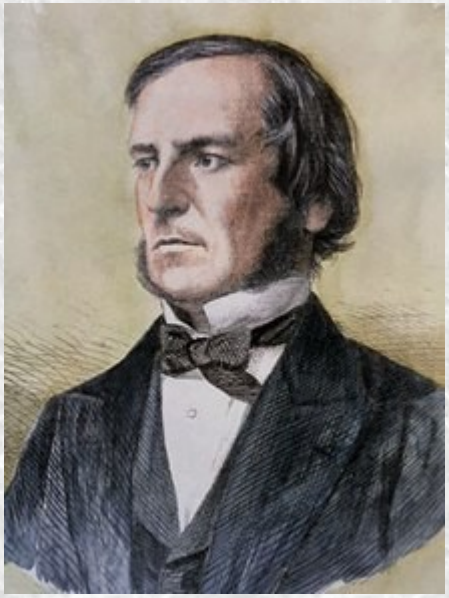
Компютърни архитектури
CSCB008

Приложения на мултиплексорите

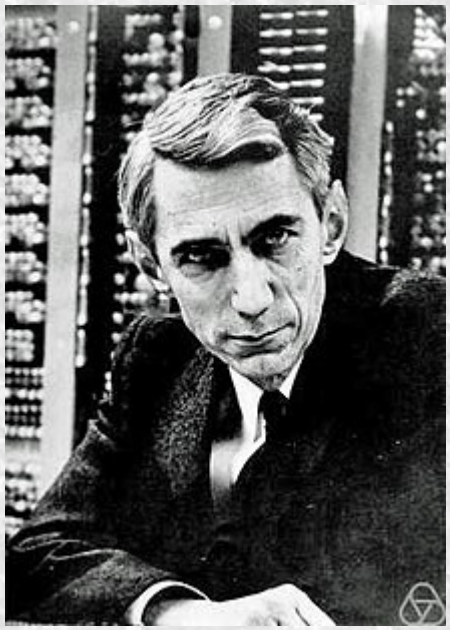
доц. д-р Ясен Горбунов

2021

Декомпозиция на Шенън



Джордж Бул (1815 – 1864)
самоук английски математик, философ, логик



Клод Шенън (1916 – 2001)
Американски математик, електроинженер,
криптограф – "баща на информационната теория"

Бул представя неговото разложение в Proposition II, "To expand or develop a function involving any number of logical symbols", в неговия труд Laws of Thought (1854).

Шенън споменава разложението в статия от 1948 г. и показва интерпретацията на превключванията на тъждествата.

- BDD (binary decision diagrams – разклоняващи се дървета) следва от системното използване на теоремата за разложението.
- Всяка булева функция може да бъде реализирана непосредствено чрез използването на йерархична структура от мултиплексори чрез повтарящо се прилагане на теоремата.
- Микропроцесорите изобилстват от мултиплексори

Декомпозиция на Шенън

Разложение на Бул = Разложение на Шенън = Декомпозиция на Шенън = „фундаментална теорема на Булевата алгебра“

Произволна булева функция $f(x_1, x_2, \ldots, x_n)$ може да бъде декомпозирана на по-малки „под“-функции, които могат да бъдат реализирани изцяло чрез използване на двувходови мултиплексори.

Декомпозицията има вида:

$$f(x_1, x_2, \ldots, x_n) = \overline{x_1} . f(0, x_2, \ldots, x_n) + x_1 . f(1, x_2, \ldots, x_n)$$

$$f(x_1, x_2, \ldots, x_n) = \overline{x_1} . f_{\overline{x_1}} + x_1 . f_{x_1}$$

негативен кофактор позитивен кофактор

Декомпозицията на Шенън позволява синтезирането на булеви **функции с произволна сложност** чрез използването единствено на двувходови мултиплексори.

Декомпозиция на Шенън – пример

Нека е дадена следната булева функция: $f(x_1, x_2, x_3) = x_1 . x_2 + x_1 . x_3 + x_2 . x_3$

Променливата x_1 трябва да бъде последователно заместена с 0 и 1

$$f(x_1, x_2, x_3) = \overline{x_1} . f_{\overline{x_1}} + x_1 . f_{x_1} = \overline{x_1} . (x_2 . x_3) + x_1 . (x_2 + x_3 + x_2 . x_3) = \overline{x_1} . (x_2 . x_3) + x_1 . (x_2 + x_3)$$

теорема за поглъщане

$x_1 + x_1 \cdot x_2 = x_1$

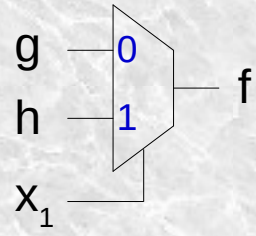
g

h

$g = x_2 . x_3$

$h = x_2 + x_3$

$$f(x_1, x_2, x_3) = \overline{x_1} . g + x_1 . h$$



Декомпозиция на Шенън – пример

$$f(x_1, x_2, x_3) = \overline{x_1}.g + x_1.h$$

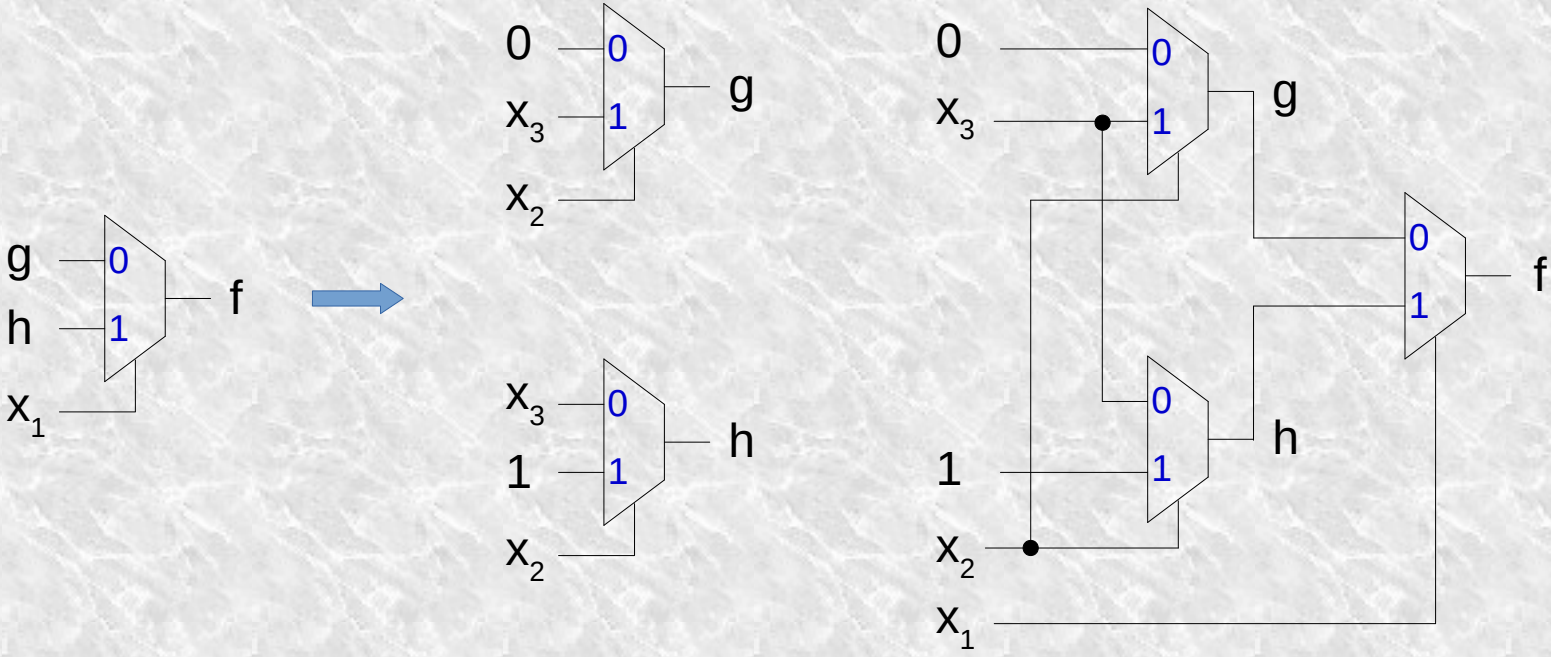
кофактори

$$g = x_2.x_3 = \overline{x_2}.0 + x_2.x_3$$

$$h = x_2 + x_3 = \overline{x_2}.x_3 + x_2.(1 + x_3) = \overline{x_2}.x_3 + x_2.1$$

1

кофактори



Други приложения на мултиплексорите

Barrel Shifter

- двупосочно **преместване** с желан брой битове – умножение, деление
- двупосочно **завъртане**
  - в комбинация с тригери (памет) – **преместващи регистри** и **кръгови буфери**
- **ускорено** извършване на аритметични операции

1) **логическо преместване** – запълва празните позиции с 0

11001 LSR 2 = 00110, LSL 2 = 00100 (Logic Shift Left / Right)

Други приложения на мултиплексорите

Barrel Shifter

- двупосочно **преместване** с желан брой битове – умножение, деление
- двупосочно **завъртане**
  - в комбинация с тригери (памет) – **преместващи регистри** и **кръгови буфери**
- **ускорено** извършване на аритметични операции

1) **логическо преместване** – запълва празните позиции с 0

11001 LSR 2 = 00110, LSL 2 = 00100 (Logic Shift Left / Right)

2) **аритметично преместване** – запълва празните позиции със знаковия бит при преместване надясно; при преместване наляво повтаря логическото преместване

11001 ASR 2 = 11110, ASL 2 = 00100 (Arithmetic Shift Left / Right)

Други приложения на мултиплексорите

Barrel Shifter

- двупосочно **преместване** с желан брой битове – умножение, деление
- двупосочно **завъртане**
  - в комбинация с тригери (памет) – **преместващи регистри** и **кръгови буфери**
- **ускорено** извършване на аритметични операции

1) **логическо преместване** – запълва празните позиции с 0

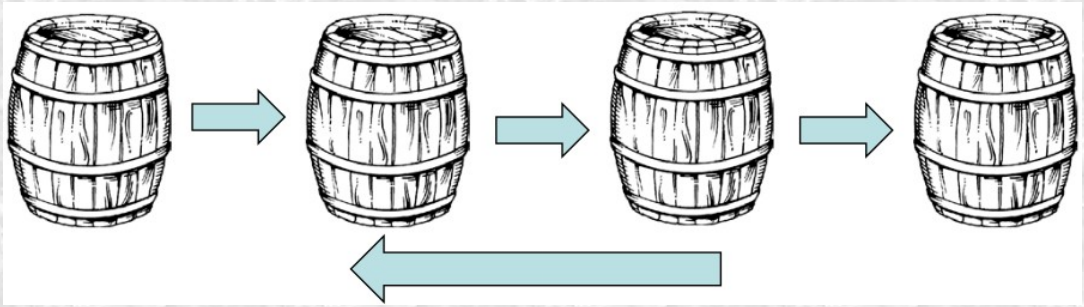
   11001 LSR 2 = 00110, LSL 2 = 00100 (Logic Shift Left / Right)

2) **аритметично преместване** – запълва празните позиции със знаковия бит при преместване надясно; при преместване наляво повтаря логическото преместване

   11001 ASR 2 = 11110, ASL 2 = 00100 (Arithmetic Shift Left / Right)

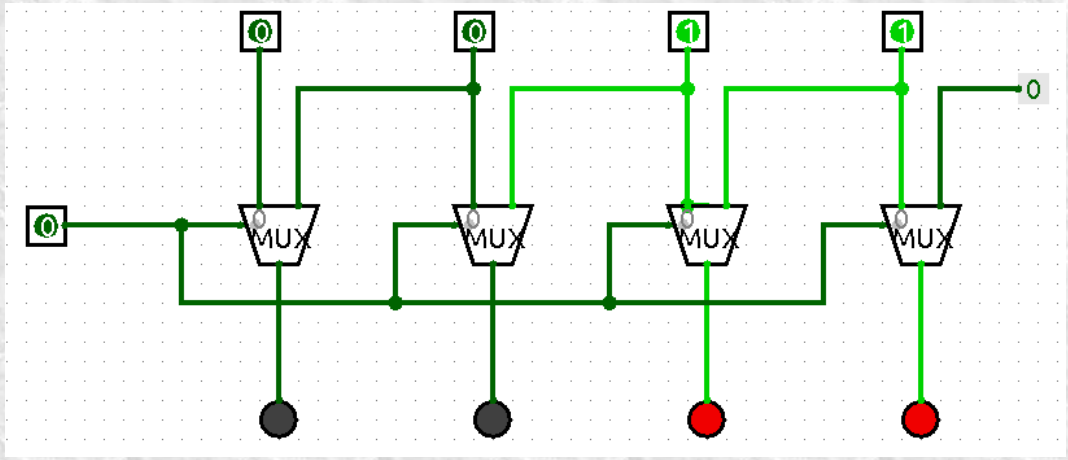
3) **завъртане** – запълва празните позиции с изпадналите битове

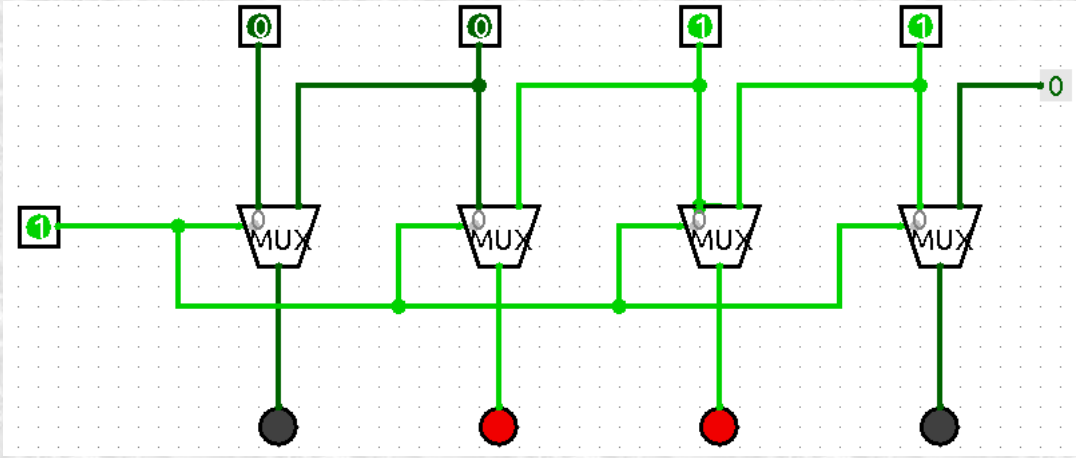
   11001 ROR 2 = 01110, ROL 2 = 00111 (Rotate Left / Right)



Други приложения на мултиплексорите

Barrel Shifter

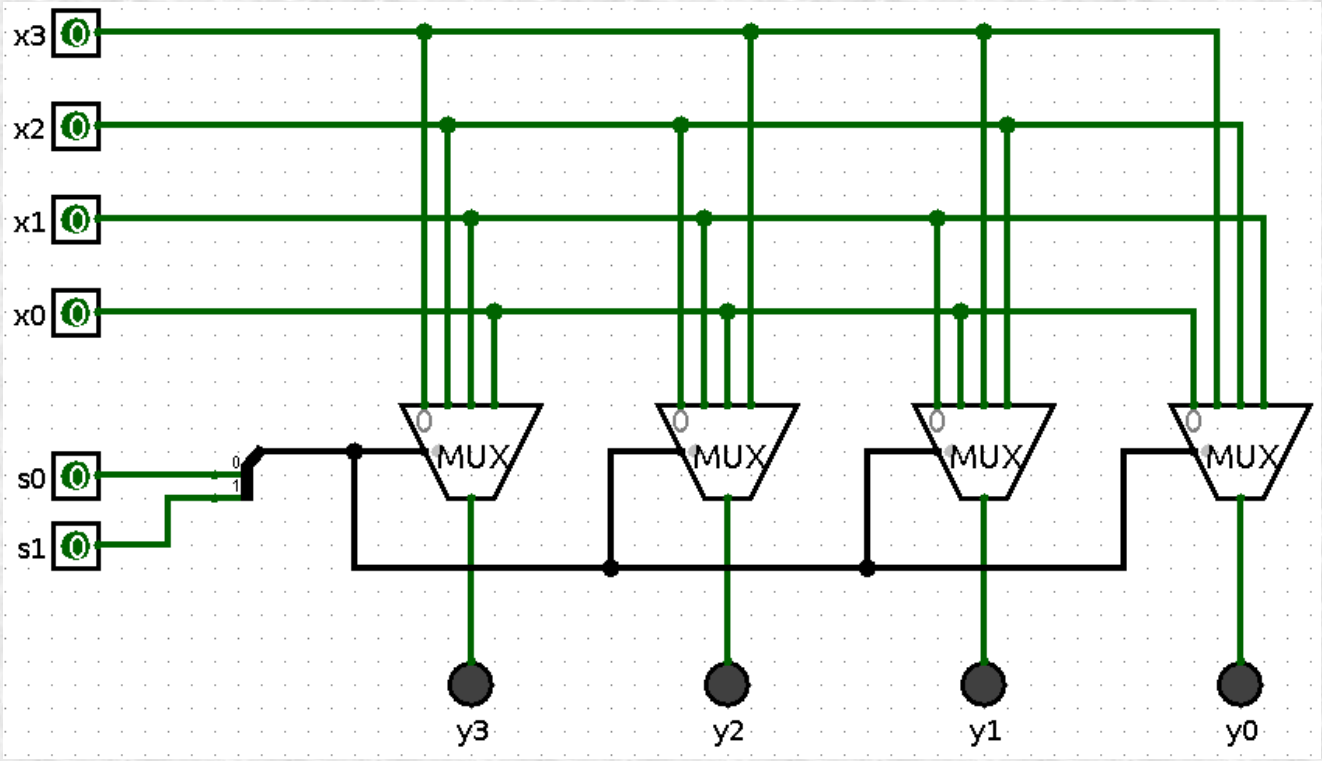




Други приложения на мултиплексорите

Barrel Shifter

s1	s0	y3	y2	y1	y0
0	0	**x3**	x2	x1	x0
0	1	x0	**x3**	x2	x1
1	0	x1	x0	**x3**	x2
1	1	x2	x1	x0	**x3**



декодер