

Сумматор Линга Склянского, Radix-2, 4-х битный

Ling Sklansky, Radix-2

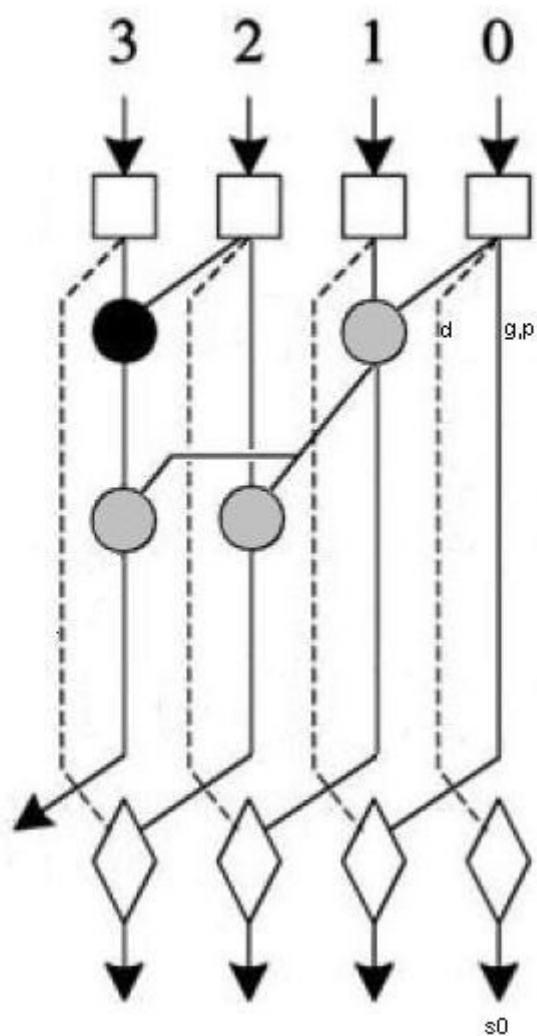


Рис.1. Граф сумматора Линга Склянского, Radix-2, 4-х разрядного [1].



Рис.2. Снимок модели сумматора Линга, архитектура Склянского, Radix-2, 4-х битного (полубайтного) на уровне логических элементов в логическом симуляторе [Atanua/Win32 1.0.081116 - Personal Edition](http://atanua.com/).

Код модели сумматора Линга Склянского, Radix-2, 4-х битного (полубайтного) в логическом симуляторе Atanua/Win32:

<http://andserkul.narod.ru/AdderRadix2LingSklansky4bit.atanua>

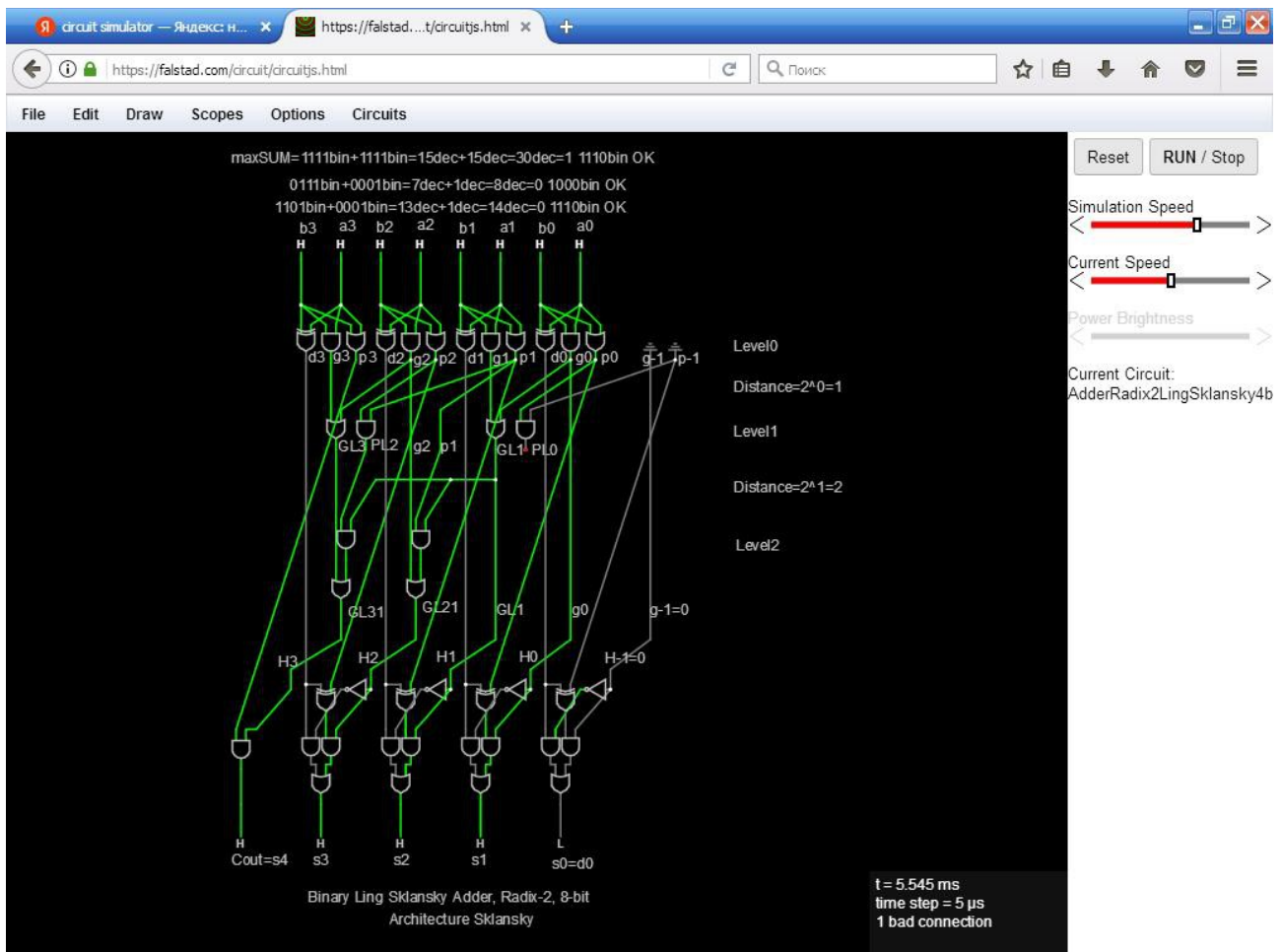


Рис.3. Снимок модели сумматора Линга, архитектура Скланского, Radix-2, 4-х битного (полубайтного) на уровне логических элементов в Circuit Simulator'e.

Адрес модели в Circuit Simulator'e: <https://tinyurl.com/ya5nhkud>

Схема физической реализации сумматора Линга Скланского на уровне CMOS транзисторов, приведённая ниже, отличается от схемы на уровне логических элементов, но полностью соответствует схеме на уровне логических элементов, той же системе логических уравнений и той же программе проверки сумматора.

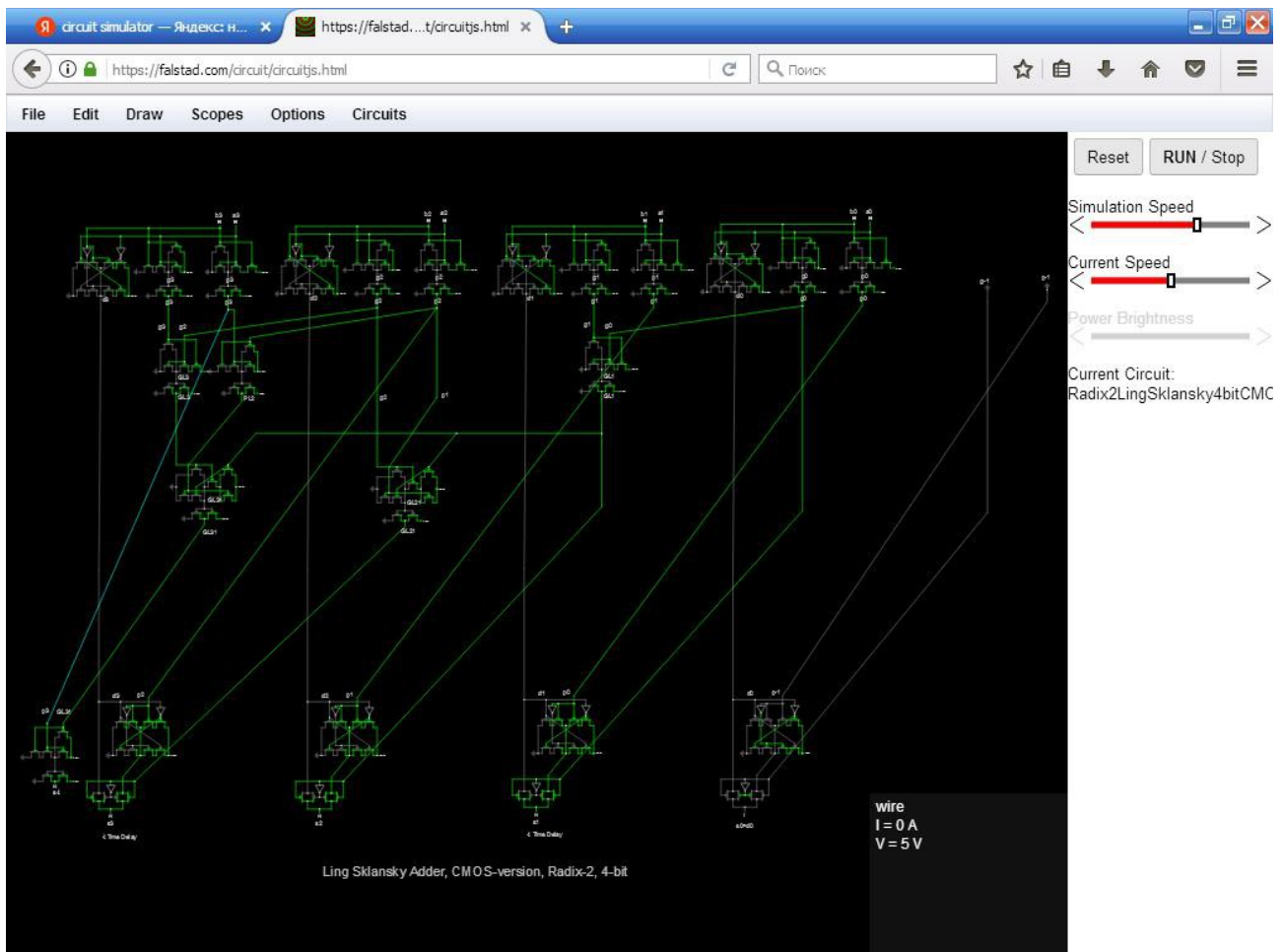


Рис.4. Снимок модели сумматора Линга, архитектура Склянского, Radix-2, 4-х битного (полубайтного) на уровне CMOS транзисторов в Circuit Simulator'e.

Адрес файла с кодом модели в Circuit Simulator'e:

<http://andserkul.narod.ru/Radix2LingSklansky4bitCMOS.noext>

Сумматор Линга Склянского, Radix-2, 4-х битный (полубайтный), в виде системы логических уравнений:

```
'Level0-----Warning-----
p0 = a0 OR  b0   'Initial only CLA & Ling Propagate (not in PPA)
g0 = a0 AND b0   'Initial CLA & Ling & PPA Generate
d0 = a0 XOR  b0   'Only Ling Initial half bit generate (p0 in PPA)

p1 = a1 OR  b1
g1 = a1 AND b1
d1 = a1 XOR  b1

p2 = a2 OR  b2
g2 = a2 AND b2
d2 = a2 XOR  b2

p3 = a3 OR  b3
g3 = a3 AND b3
d3 = a3 XOR  b3

'Level1-----Distance=2^0=1
'(G,P)=(g,p) o (g,p)=(g,p)
'GLi=gi OR gi-1
'PLi=pi AND pi-1 (9)  'Distance=1
```

```

GLm1 = 0          'for k<0
PLm2 = 0          'for k<0

' (GL0, PLm1)
GL0 = g0 OR gm1
PLm1=0           'for k<0

' (GL1, PL0)
GL1 = g1 OR g0
PL0 = p0 AND pm1 'Distance=1

' (GL3, PL2)
GL3 = g3 OR g2
PL2 = p2 AND p1  'Distance=1

'Level2-----Distance=2^1=2
'(G,P)=(g,p) o (g',p')= (g OR (p AND g'),p AND p')
'G=g OR (p AND p')
'P=      p AND p'

'(GL2, PL1) o (GL1, PLm1)      '
GL21 = g2 OR (p1 AND GL1)      '

'(GL3, PL2) o (GL1, PL0)      '
GL31 = GL3 OR (PL2 AND GL1)    '

'Ling PsevdoCarry (H)-----
Hm1 = GLm1          'Ling PsevdoCarry
H0 = g0             'Ling PsevdoCarry
H1 = GL1           'Ling PsevdoCarry
H2 = GL21
H3 = GL31

'SUM-----
'si=(/Hi-1 AND di) OR (Hi-1 AND (di XOR pi-1))

s0 = ((1-Hm1) AND d0) OR (Hm1 AND (d0 XOR pm1)) 's0=d0
s1 = ((1-H0) AND d1) OR (H0 AND (d1 XOR p0))
s2 = ((1-H1) AND d2) OR (H1 AND (d2 XOR p1))
s3 = ((1-H2) AND d3) OR (H2 AND (d3 XOR p2))
s4 = p3 AND H3 'Cout=s4

```

Программа проверки системы логических уравнений сумматора Линга
Склянского, Radix-2, 4-х битного (полубайтного), на TurboBasic'e:
<http://andserkul.narod.ru/R2LSK4B.bas>

Литература:

1. [High-Speed Parallel-Prefix VLSI Ling Adders. Giorgos Dimitrakopoulos and Dimitris Nikolos.](#)
2. [Logical Effort of Higher Valency Adders. David Harris. Harvey Mudd College 301](#)

[E.Twelfth St. Claremont, CA 91711](#)

3. [Сумматор Линга Склянского, Radix-2, 8-ми битный. Куликов А. С.](#)
4. [Сумматор Линга Склянского, Radix-2, 16-ти битный. Куликов А. С.](#)
5. [Сумматор Линга Когге-Стоуна, Radix-2, 4-х битный. Куликов А. С.](#)
6. [Сумматор Линга Когге-Стоуна, Radix-2, 8-ми битный. Куликов А. С.](#)
7. [Сумматор Линга Когге-Стоуна, Radix-2, 16-ти битный. Куликов А. С.](#)

Приложение 1.

[TurboBasic 1.0](#)

Куликов А.С., Россия-Русь, Москва, Царицыно, версия 2022.02.14.