Сумматор Когге-Стоуна, Radix-4, 8-ми битный

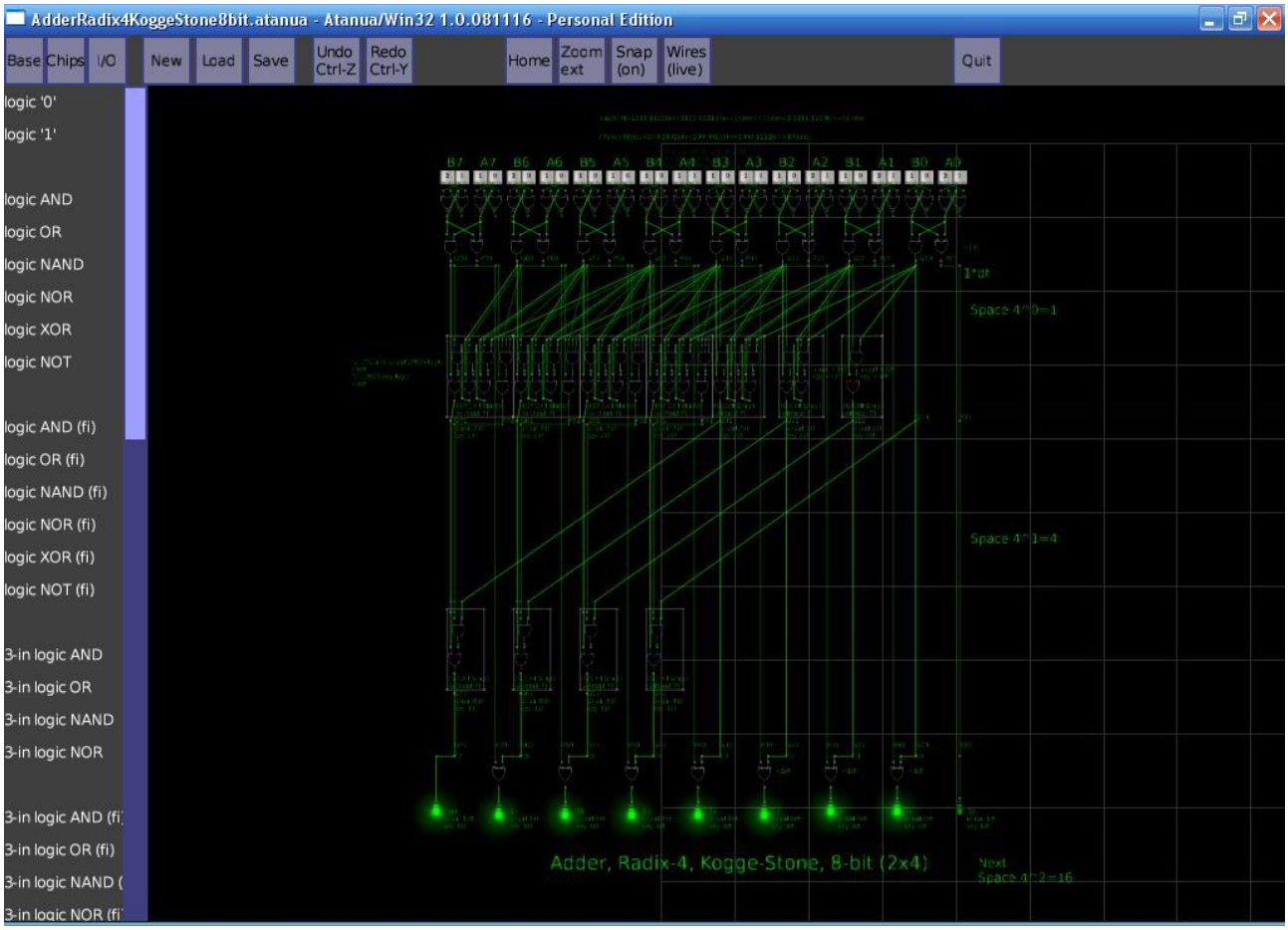

Рис.1. Снимок модели сумматора Когге-Стоуна, Radix-4, 8-ми битного, в логическом симуляторе [Atanua/Win32 1.0.081116 – Personal Edition.]

Код модели сумматора Когге-Стоуна, Radix-4, 8-ми битного, в логическом симуляторе Atanua/Win32:
http://andserkul.narod.ru/AdderRadix4KoggeStone8bit.atanua

Сумматор Когге-Стоуна, Radix-4, 8-ми битный, в виде логических уравнений:

```
'--------------------
P00 = A0 XOR B0
G00 = A0 AND B0

P10 = A1 XOR B1
G10 = A1 AND B1

P20 = A2 XOR B2
G20 = A2 AND B2

P30 = A3 XOR B3
G30 = A3 AND B3

P40 = A4 XOR B4
G40 = A4 AND B4

P50 = A5 XOR B5
G50 = A5 AND B5
```

```
P60 = A6 XOR B6
G60 = A6 AND B6

P70 = A7 XOR B7
G70 = A7 AND B7

'-------------------------
G11 = G10 OR (P10 AND G00)

G21 = G20 OR (P20 AND (G10 OR (P10 AND G00)))

G31 = G30 OR (P30 AND (G20 OR (P20 AND (G10 OR (P10 AND G00)))))

P41 = P40 AND P30 AND P20 AND P10
G41 = G40 OR (P40 AND (G30 OR (P30 AND (G20 OR (P20 AND G10)))))

P51 = P50 AND P40 AND P30 AND P20
G51 = G50 OR (P50 AND (G40 OR (P40 AND (G30 OR (P30 AND G20)))))

P61 = P60 AND P50 AND P40 AND P30
G61 = G60 OR (P60 AND (G50 OR (P50 AND (G40 OR (P40 AND G30)))))

P71 = P70 AND P60 AND P50 AND P40
G71 = G70 OR (P70 AND (G60 OR (P60 AND (G50 OR (P50 AND G40)))))

'-------------------------
G42 = G41 OR (P41 AND G00)

G52 = G51 OR (P51 AND G11)

G62 = G61 OR (P61 AND G21)

G72 = G71 OR (P71 AND G31)

'-------------------------
S0 = P00

S1 = P10 XOR G00

S2 = P20 XOR G11

S3 = P30 XOR G21

S4 = P40 XOR G31

S5 = P50 XOR G42

S6 = P60 XOR G52

S7 = P70 XOR G62

Cout = G72
```

Программа проверки логических уравнений сумматора Когге-Стоуна, Radix-4, 8-ми битного, на TurboBasic'e:
http://andserkul.narod.ru/R4KS8B.bas

Разница в схемотехнике.
Логические уравнения оператора вычисления P и G одинаковы и для оператора вычисления P и G на обычной физической реализации логики (TTL и CMOS) и для

оператора вычисления P и G на физической реализации логики на CMOS-ключах, но при физической реализации логики на CMOS-ключах суммарное время задержки (gate delay, delay time, n*dt) вычисляется иначе и значительно меньше, чем на обычной логике, т. е. быстродействие оператора вычисления P и G при физической реализации логики на CMOS-ключах значительно выше. Блоки в рамках при физической реализации логики на CMOS-ключах вычисляют значения P и G за время равное 1*dt, где dt – время задержки в одном типовом физическом логическом элементе.

Так как параллельно префиксные сумматоры (Parallel Prefix Adders, PPA), в том числе и сумматоры Когге-Стоуна с основаниями больше 2, строятся не трёхаргументными (трёхоперандными) блоками с единицей переноса на входе и с последовательным соединением блоков, а целиком двухаргументными (двухоперандными), то в них исчезают понятия «полусумматор» и «полный сумматор», но сохраняются понятия «двухаргументный» и «трёхаргументный» (с единицей переноса на входе), причём «трёхаргументные» (с единицей переноса на входе) теоретически возможны, но практически в них нет почти никакой нужды.

Литература:

1. Kogge–Stone adder. Wikipedia.

2. Logical Effort of Higher Valency Adders. David Harris

3. Design Space Exploration for Power-Efficient Mixed-Radix Ling Adders. Chung-Kuan Cheng. Computer Science and Engineering Depart. University of California, San Diego.

Приложение 1.

TurboBasic 1.0

Куликов А.С., Россия-Русь, Москва, Царицыно, версия 2021.09.22.