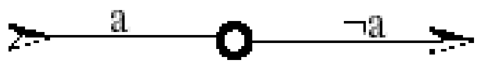
Boolean Gates

$$a = \{N : a_N = 1\} \quad b = \{N : b_N = 1\} \quad c = \{N : c_N = 1\}$$

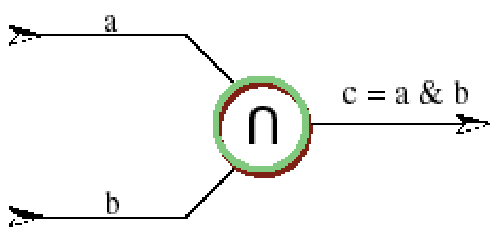
NOT



$$c = \neg a$$

$$c_N = 1 - a_N$$

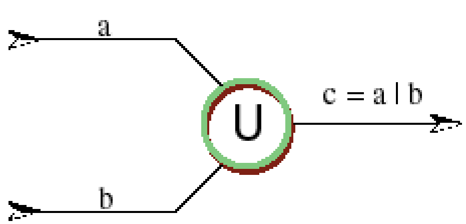
AND



$$c = a \cap b$$

$$c_N = a_N \cap b_N = a_N b_N$$

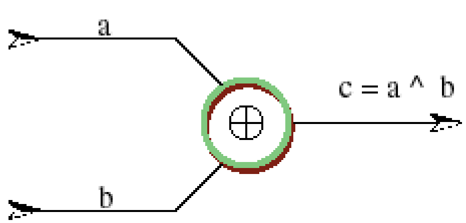
OR



$$c = a \cup b$$

$$c_N = a_N \cup b_N = a_N + b_N - a_N b_N$$

XOR



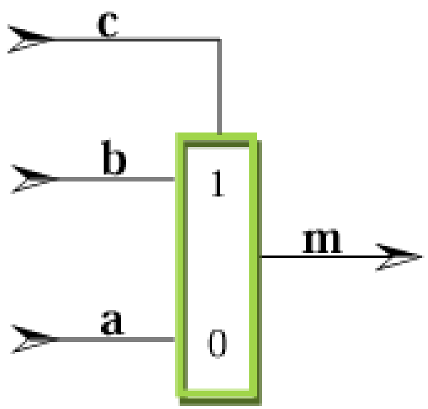
$$c = a \oplus b$$

$$c_N = a_N \oplus b_N = a_N + b_N - 2a_N b_N$$

Multiplexer

$$< x, \neg mux >$$

$$< 0, \overline{0}, x, \mathrm{mux} >$$



$$m = \mathrm{mux}(c, b, a)$$

$$m = c ? b : a$$

$$c_N = 0 : m_N = a_N$$

$$c_N = 1 : m_N = b_N$$

$$\neg x = mux(x, 0, \overline{0})$$

$$x \cap y = mux(x, y, x)$$

$$x \cup y = mux(x, x, y)$$

$$m = (c \cap b) \uplus (\neg c \cap a)$$

$$m_N = c_N b_N + (1 - c_N) a_N$$

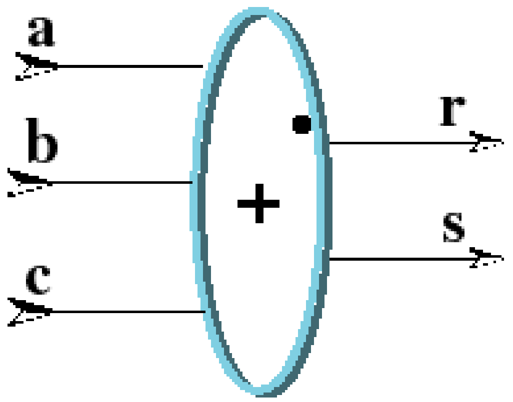
$$m = a \oplus (c \cap (a \oplus b))$$

Shannon Decomposition

$$\begin{aligned} f(x_1 x_2 \cdots x_i) &= (\neg x_1 \cap f(0 x_2 \cdots x_i)) \cup (x_1 \cap f(1 x_2 \cdots x_i)) \\ &= mux(x_1, f(1 x_2 \cdots x_i), f(0 x_2 \cdots x_i)) \\ &= (1 - x_1) f(0 x_2 \cdots x_i) + x_1 f(1 x_2 \cdots x_i) \end{aligned}$$

Full Adder

Icon



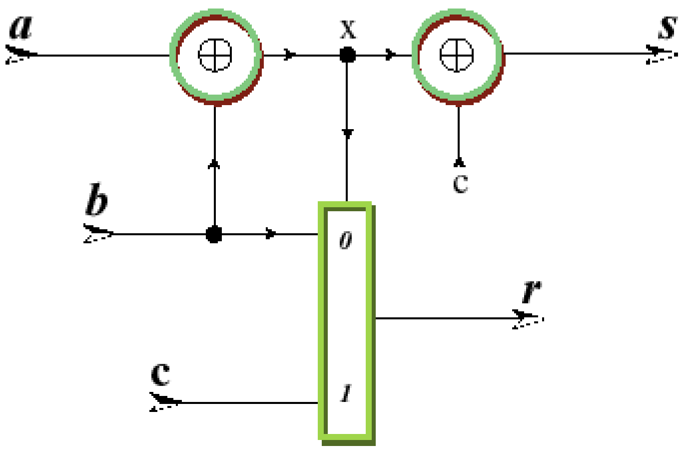
Function

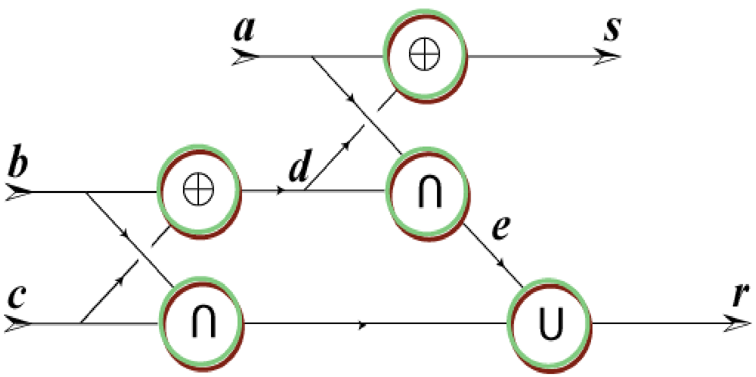
$$a_N + b_N + c_N = s_N + 2r_N$$

Specification

$$s_N = a_N \oplus b_N \oplus c_N$$
$$r_N = a_N b_N \oplus b_N c_N \oplus c_N a_N$$

Implementations





Boolean Basis

$$< 0, x, \neg, \cap, \cup, \oplus >$$

$$< x, \neg, \cap, \cup >$$

$$0 = x \oplus x$$
$$a \oplus b = (a \cap \neg b) \cup (\neg a \cap b)$$

$$\overline{0} \notin < 0, x, \cap, \cup, \oplus >$$

$$\overline{0} = \neg 0 = -1 = (1) = \frac{1}{1-z}$$

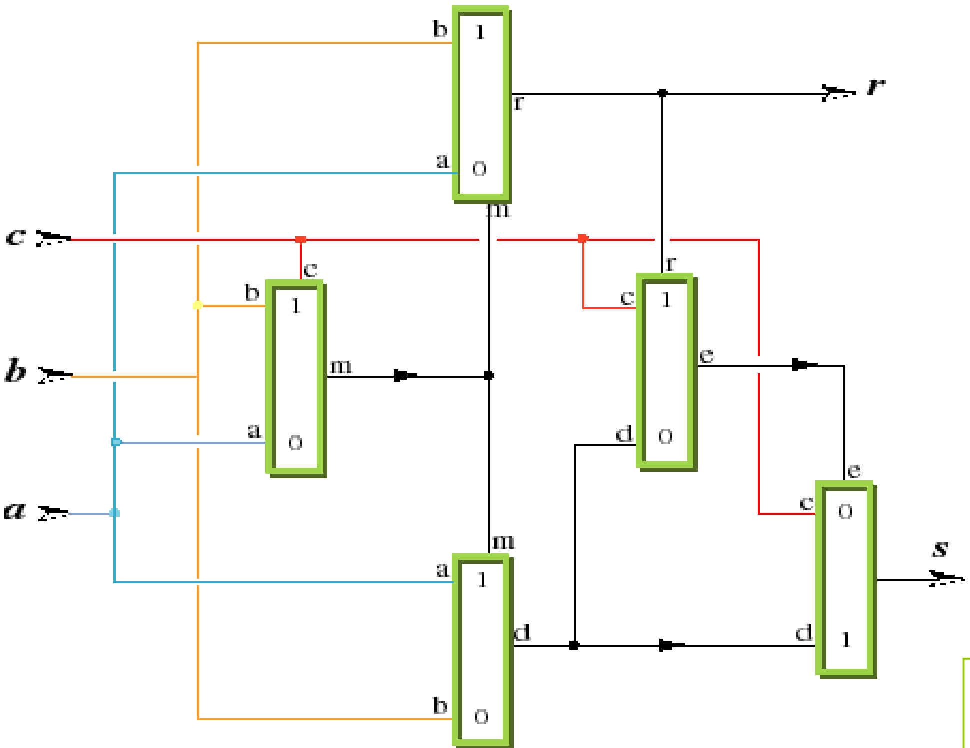
$$< \overline{0}, x, \cap, \oplus >$$

$$\neg x = \overline{0} \oplus x$$
$$a \cup b = a \oplus b \oplus (a \cap b)$$

$$< x, \overline{\cap} >$$

$$\neg a = a \overline{\cap} a$$
$$a \cup b = (\neg a) \overline{\cap} (\neg b)$$
$$a \cap b = \neg (a \overline{\cap} b)$$

Net-list



s = mux(e, d, c)

r = mux(m, b, a)

m = mux(c, b, a)

d = mux(m, a, b)

e = mux(r, c, d)

Topological Sort

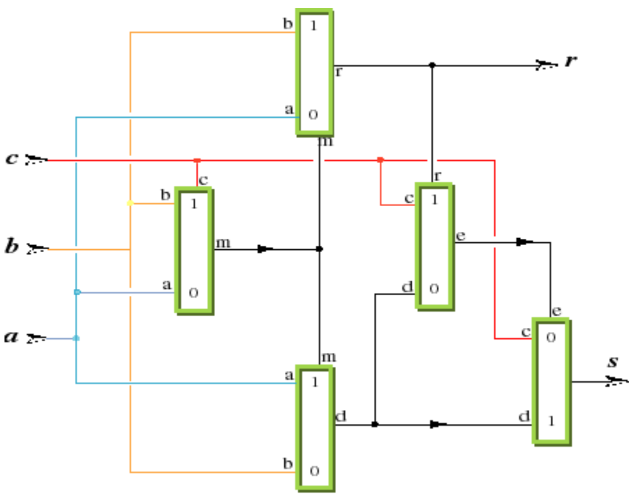
m = mux(c, b, a);

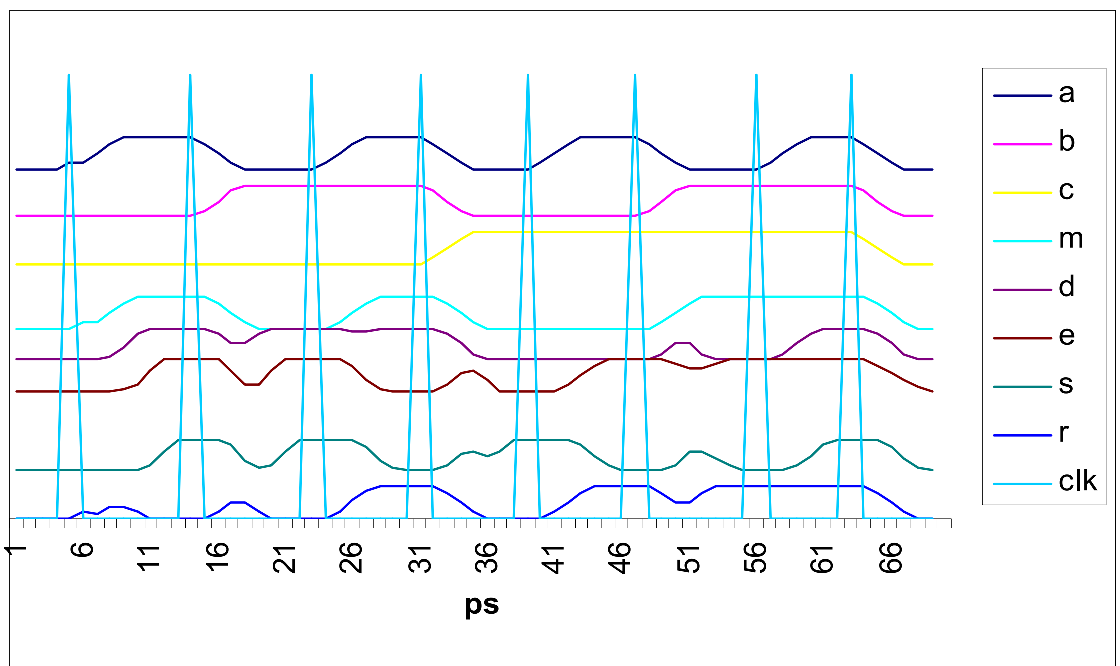
d = mux(m, a, b); **r = mux(m, b, a);**

e = mux(r, c, d);

s = mux(e, d, c);

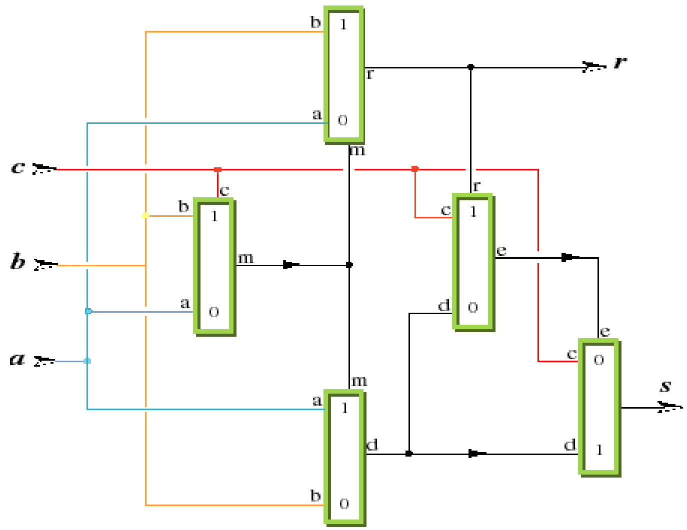
Physical Simulation





ps	a	b	c	m	d	e	s	r
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0
7	1	0	0	1	0	0	0	0
8	1	0	0	1	0	0	0	0
9	1	0	0	1	1	0	0	0
10	1	0	0	1	1	1	0	0
11	1	0	0	1	1	1	1	0
12	1	0	0	1	1	1	1	0
13	1	0	0	1	1	1	1	0
14	1	0	0	1	1	1	1	0
15	1	1	0	1	1	1	1	0
16	0	1	0	1	1	1	1	1
17	0	1	0	0	1	0	0	1
18	0	1	0	0	1	0	0	0
19	0	1	0	0	1	1	0	0
20	0	1	0	0	1	1	1	0
21	0	1	0	0	1	1	1	0
22	0	1	0	0	1	1	1	0
23	0	1	0	0	1	1	1	0
25	1	1	0	0	1	1	1	0
26	1	1	0	1	1	1	1	1
27	1	1	0	1	1	0	1	1
28	1	1	0	1	1	0	0	1
29	1	1	0	1	1	0	0	1
30	1	1	0	1	1	0	0	1
31	1	1	0	1	1	0	0	1

Logical Simulation



Topological Sort

m = mux(c, b, a,)

r = mux(m, b, a), d = mux(m, a, b)

e = mux(r, c, d)

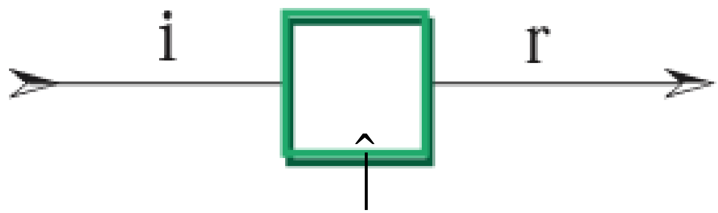
s = mux(e, d, c)

a	b	c	s	r	d	m	e	a+b+c	s+2r
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	1	1	1
0	1	0	1	0	1	0	1	1	1
1	1	0	0	1	1	1	0	2	2
0	0	1	1	0	0	0	0	1	1
1	0	1	0	1	0	0	1	2	2
0	1	1	0	1	0	1	1	2	2
1	1	1	1	1	1	1	1	3	3

Synchronous Register

$$\boxed{\begin{aligned} i &= \sum i_N 2^N \\ r &= \sum r_N 2^N \end{aligned}}$$

r = z i



$$\boxed{\begin{aligned} r_{N+1} &= i_N \\ r_0 &= 0 \end{aligned}}$$

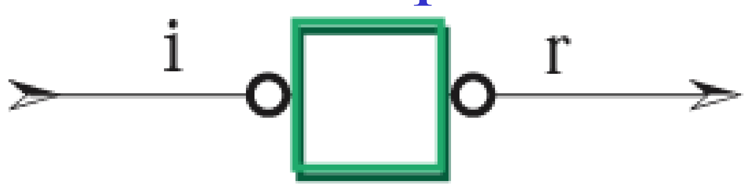
$$\boxed{r = 2i}$$

Clock is implicit: $t = \sum_{N \in \mathbb{N}} \partial(t - N)$

All registers share the same clock.

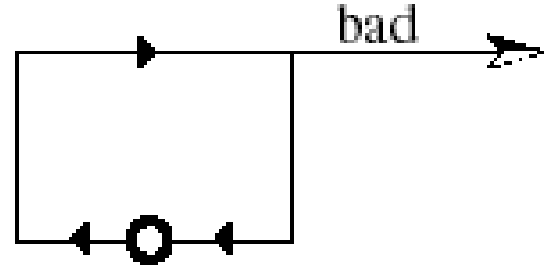
$$\boxed{r = 1 + 2i}$$

r = z$_1$ i



$$\boxed{\begin{aligned} r_{N+1} &= i_N \\ r_0 &= 1 \end{aligned}}$$

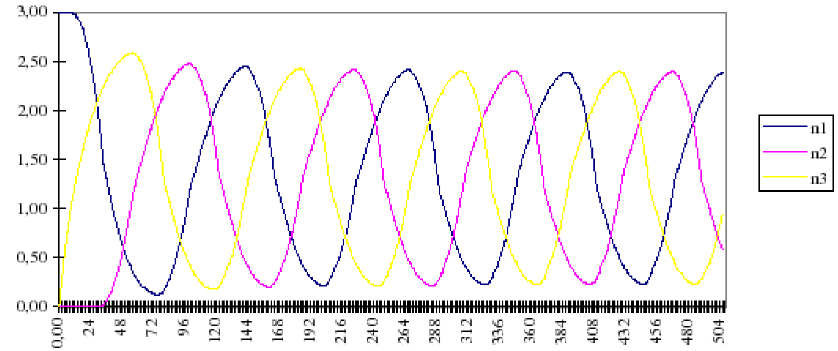
Feedback



Problem: bad = ¬bad

$$bad_0 = 1 - bad_0$$

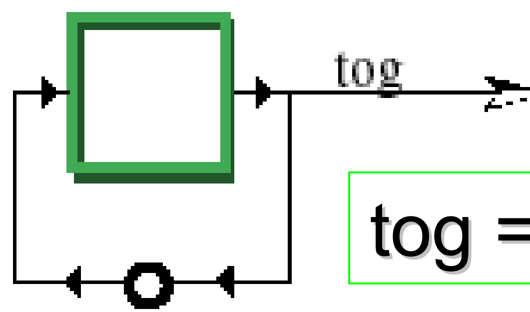
$$bad_0 = \frac{1}{2} \notin \{0,1\}$$



No problem: tog = z¬tog

$$tog_0 = 0 \qquad tog_{2N} = 0$$

$$tog_{N+1} = 1 - tog_N \qquad tog_{2N+1} = 1$$



tog = (01)

Solutions:

1. ***No*** combinational feedback
2. ***Stable*** combinational feedback

Sequential Circuit

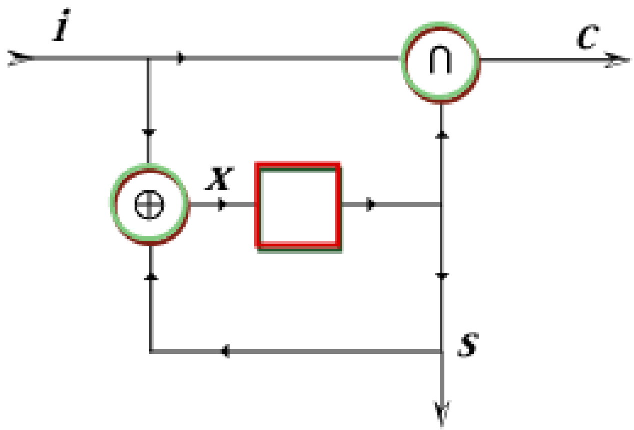
Net-list

$$cm2(i) = (s, c)$$

$$\{\ s = z(i \oplus s)$$

$$c = i \cap s\}$$

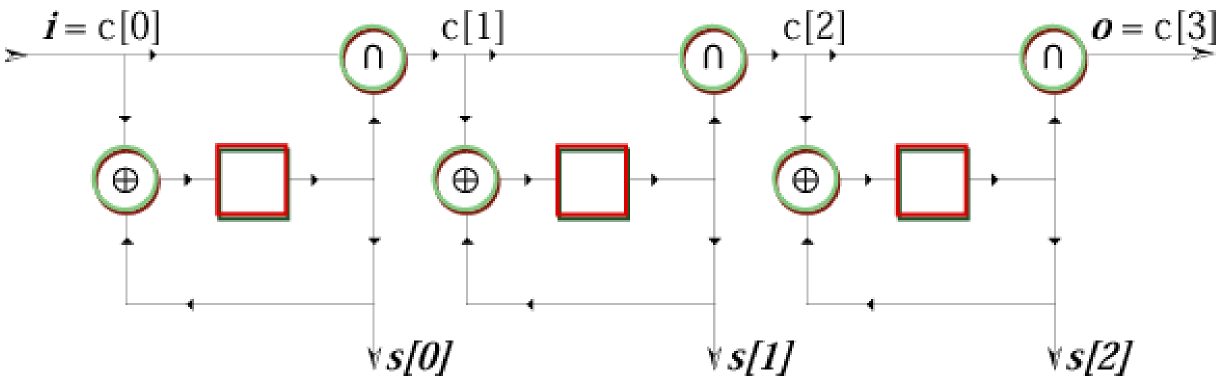
Schema



Infinite Boolean System

$$\begin{array}{llll} s_0 = 0 & s_1 = x_0 & & s_N = x_{N-1} \\ x_0 = i_0 & x_1 = i_1 \oplus s_1 & \cdots & x_N = i_N \oplus s_N \quad \cdots \\ c_0 = 0 & c_1 = i_1 \cap s_1 & & c_N = i_N \cap s_N \end{array}$$

Binary Counter




```
// Binary n bit counter
fun Counter(n:int)(incr:net)= (s:net[n], ovfl:net) {
  c[0]=incr;   // carry in
  ovfl=c[n];   // carry out
  for (k<n)    (s[k] , c[k+1] ) = cm2(c[k]); }


// Counter modulo 2
fun cm2 (incr:net)= (s:net, ovfl:net) {
  s = reg(incr ^ s);
  ovfl = incr & s;     }
```