Ternarylogic LLC

Peter Lablans

Switching Models of Binary Memory Latches and New Designs

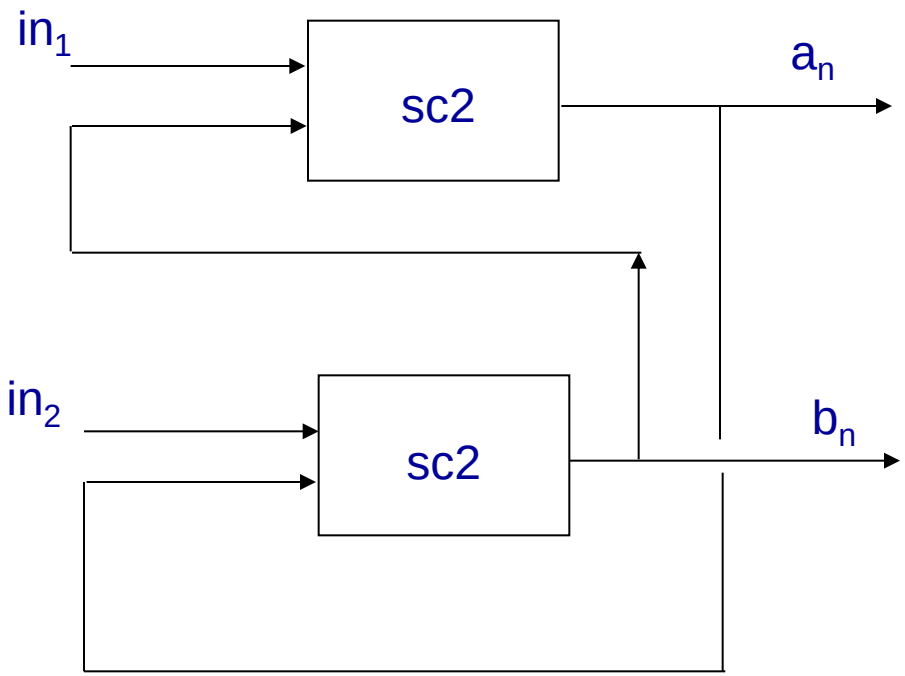
- Reasons for the Model
- The traditional binary latch
  - Traditional form in NAND and NOR
  - The latch switching model generated in Matlab
  - The a>b latch
  - The modified OR/AND latch
- Operation in Multimedia Logic and SparkFun LogicBlocks
- Another Model
- Downloads
- Quick display of Program Output

Ternarylogic

Reasons for the Model

- To create non-binary latches;
- Binary (NAND and NOR) latches are known, but are provided as a given;
- It is unclear how the latch works in detail and the significance of the forbidden state;
- What is clear is that the two devices work in a combined feedback switching configuration. While the result appears to be static, it is achieved by active switching;
- We want to look behind the scenes of the static result.

The traditional binary latch



1) a_n and b_n should be stable after inputs are stable
2) known latching functions are NAND and NOR
3) a useful configuration has output state reflect an input state

Ternarylogic

The traditional binary latch

The NAND latch:

Set	Reset	
1	1	No change
0	1	Output = 1
1	0	Output = 0
0	0	Invalid

The NOR latch:

Set	Reset	
0	0	No change
1	0	Output = 1
0	1	Output = 0
1	1	Invalid

MultimediaLogic Simulation

Ternarylogic

The traditional binary latch

It appears an oracle decided which functions to use. There is no “design” approach that tells us which functions to select.

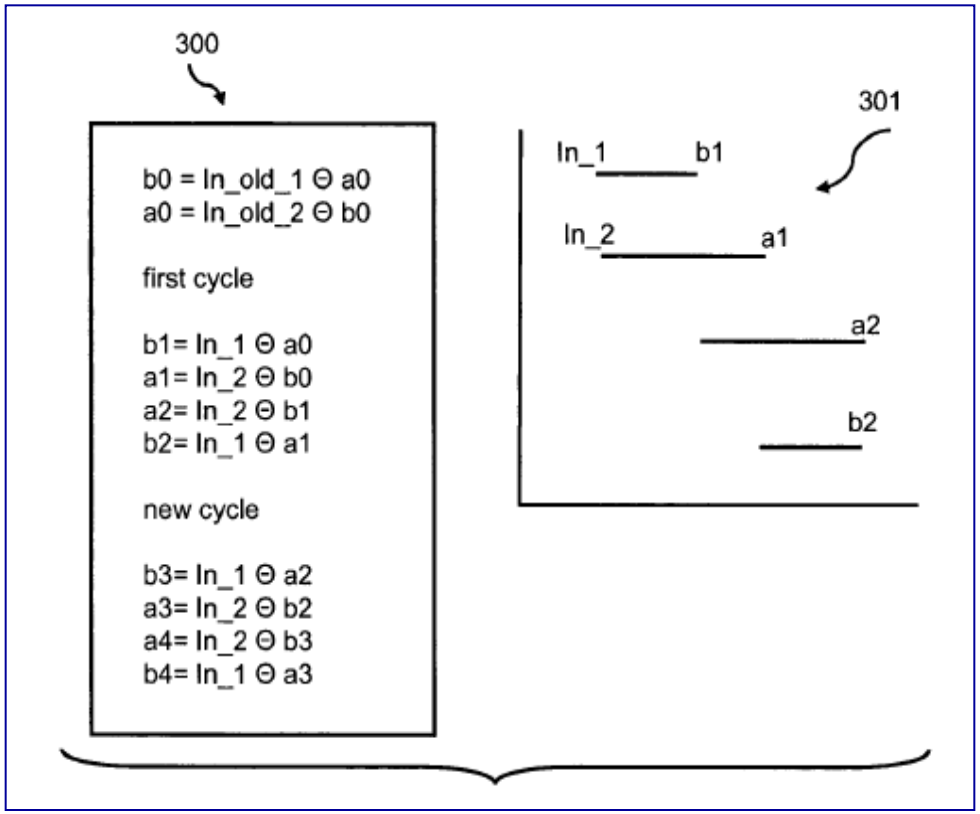
We would like to create an n-state non-binary feedback latch. However it is unclear how to select the appropriate switching functions.

Step 1. Develop analysis and model for binary latch. See if other binary latch configurations exist

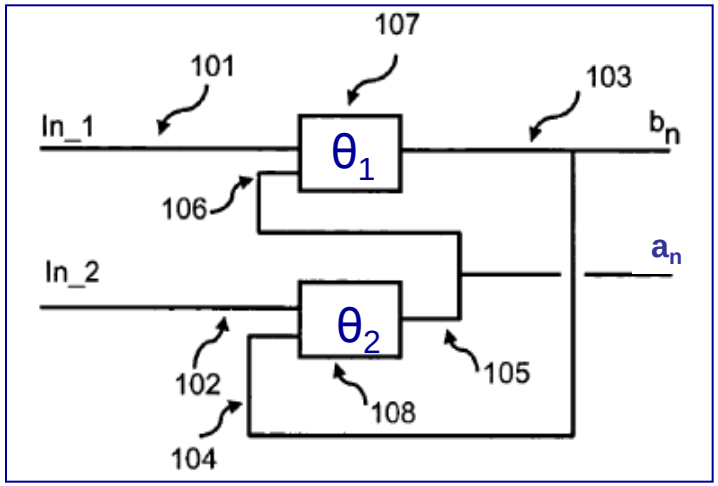
Step. 2 Extend binary model to n-state non-binary latches

The switching model

The delay/switching model



From U.S. Patent 7,365,576



Select function θ which creates stable outputs b_n and a_n.

Or:

Select functions θ_1 and θ_2 which create stable outputs b_n and a_n

Ternarylogic

The switching table in Matlab

input signals | initial outputs | generated outputs

in1	in2		b0	a0		b1	b2	b3	b4	b5		a1	a2	a3	a4	a5
0	0	8	0	0	8	1	1	1	1	1	8	1	1	1	1	1
0	0	8	1	0	8	1	1	1	1	1	8	1	1	1	1	1
0	0	8	0	1	8	1	1	1	1	1	8	1	1	1	1	1
0	0	8	1	1	8	1	1	1	1	1	8	1	1	1	1	1
0	1	8	0	0	8	1	1	1	1	1	8	1	0	0	0	0
0	1	8	1	0	8	1	1	1	1	1	8	0	0	0	0	0
0	1	8	0	1	8	1	1	1	1	1	8	1	0	0	0	0
0	1	8	1	1	8	1	1	1	1	1	8	0	0	0	0	0
1	0	8	0	0	8	1	0	0	0	0	8	1	1	1	1	1
1	0	8	1	0	8	1	0	0	0	0	8	1	1	1	1	1
1	0	8	0	1	8	0	0	0	0	0	8	1	1	1	1	1
1	0	8	1	1	8	0	0	0	0	0	8	1	1	1	1	1
1	1	8	0	0	8	1	0	1	0	1	8	1	0	1	0	1
1	1	8	1	0	8	1	1	1	1	1	8	0	0	0	0	0
1	1	8	0	1	8	0	0	0	0	0	8	1	1	1	1	1
1	1	8	1	1	8	0	1	0	1	0	8	0	1	0	1	0

```
>> sc2-1

ans =

     1     1
     1     0
```

outputs remain unchanged when input is [1 1]

outputs remain unstable when input is [1 1] after [0 0]

stable after one cycle

The NAND function

Ternarylogic

Other binary latches

a) The AND/OR latch
b) The a>b (agtb) latch
c) The ~(b>a) latch
d) The ~(a>b)/(b>a) latch

$(a>b)$	b=0	b=1
a=0	0	0
a=1	1	0

$\sim(b>a)$	b=0	b=1
a=0	1	0
a=1	1	1

$\sim(a>b)$	b=0	b=1
a=0	1	1
a=1	0	1

$(b>a)$	b=0	b=1
a=0	0	1
a=1	0	0

a>b or
a greater than b
or
'agtb' truth table

Ternarylogic

The ‘agtb’ latch switching table in Matlab

in1	in2		b0	a0		b1	b2	b3	b4	b5		a1	a2	a3	a4	a5
0	0	8	0	0	8	0	0	0	0	0	8	0	0	0	0	0
0	0	8	1	0	8	0	0	0	0	0	8	0	0	0	0	0
0	0	8	0	1	8	0	0	0	0	0	8	0	0	0	0	0
0	0	8	1	1	8	0	0	0	0	0	8	0	0	0	0	0
0	1	8	0	0	8	0	0	0	0	0	8	1	1	1	1	1
0	1	8	1	0	8	0	0	0	0	0	8	0	1	1	1	1
0	1	8	0	1	8	0	0	0	0	0	8	1	1	1	1	1
0	1	8	1	1	8	0	0	0	0	0	8	0	1	1	1	1
1	0	8	0	0	8	1	1	1	1	1	8	0	0	0	0	0
1	0	8	1	0	8	1	1	1	1	1	8	0	0	0	0	0
1	0	8	0	1	8	0	1	1	1	1	8	0	0	0	0	0
1	0	8	1	1	8	0	1	1	1	1	8	0	0	0	0	0
1	1	8	0	0	8	1	0	1	0	1	8	1	0	1	0	1
1	1	8	1	0	8	1	1	1	1	1	8	0	0	0	0	0
1	1	8	0	1	8	0	0	0	0	0	8	1	1	1	1	1
1	1	8	1	1	8	0	1	0	1	0	8	0	1	0	1	0

outputs are stable
when input is [1 1]

outputs remain unstable
when input is [1 1] after [0 0]

```
>> sc2-1

ans =

     0     0
     1     0
```

sc2 = a>b

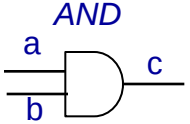
Ternarylogic

The a>b latch states

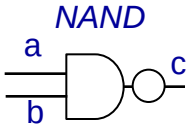
The a>b latch:

Set	Reset	
1	1	No change
0	1	Output = 0
1	0	Output = 1
0	0	Invalid

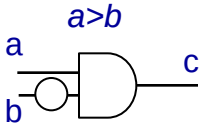
Implementing binary functions with AND, OR and inverter gates



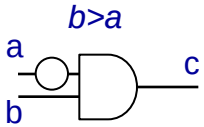
AND		b	
	c	0	1
a	0	0	0
	1	0	1



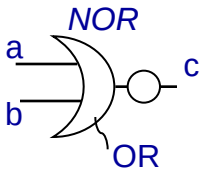
NAND		b	
	c	0	1
a	0	1	1
	1	1	0



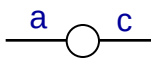
a > b		b	
	c	0	1
a	0	0	0
	1	1	0



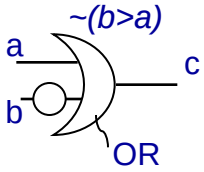
b > a		b	
	c	0	1
a	0	0	1
	1	0	0



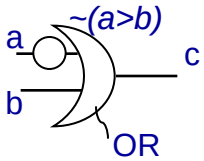
NOR		b	
	c	0	1
a	0	1	0
	1	0	0



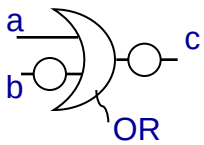
0→1 and 1→0



~ (b > a)		b	
	c	0	1
a	0	1	0
	1	1	1



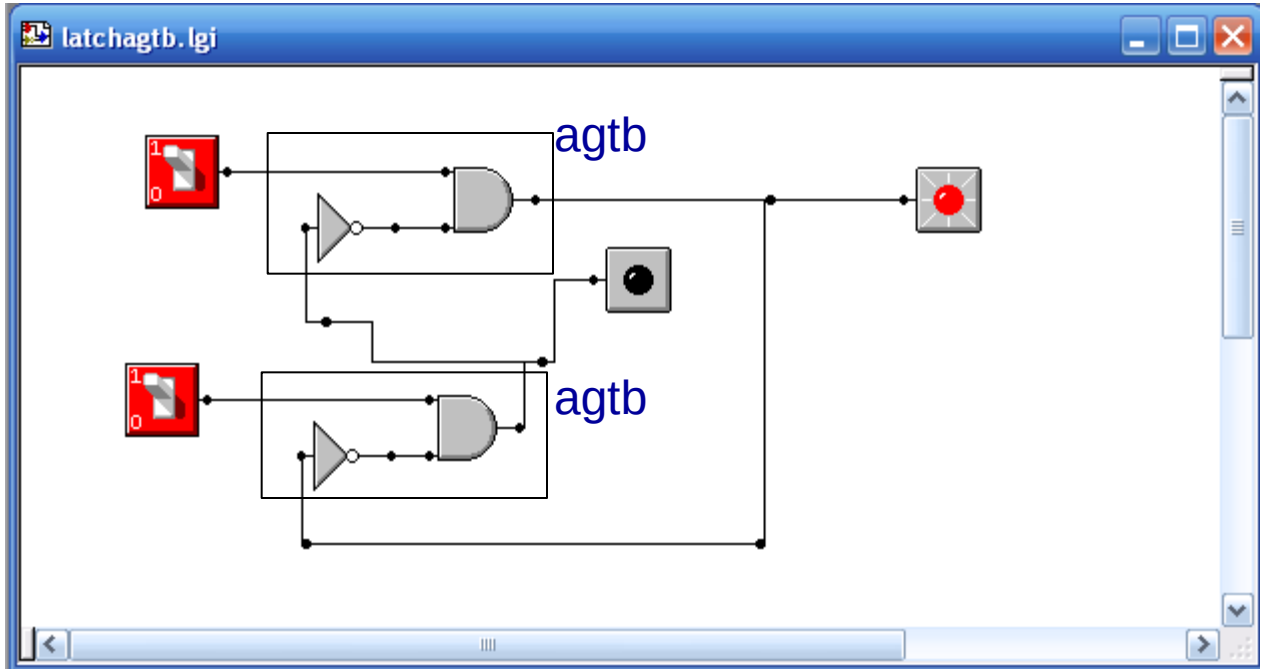
~ (a > b)		b	
	c	0	1
a	0	1	1
	1	0	1



b > a		b	
	c	0	1
a	0	0	1
	1	0	0

Ternarylogic

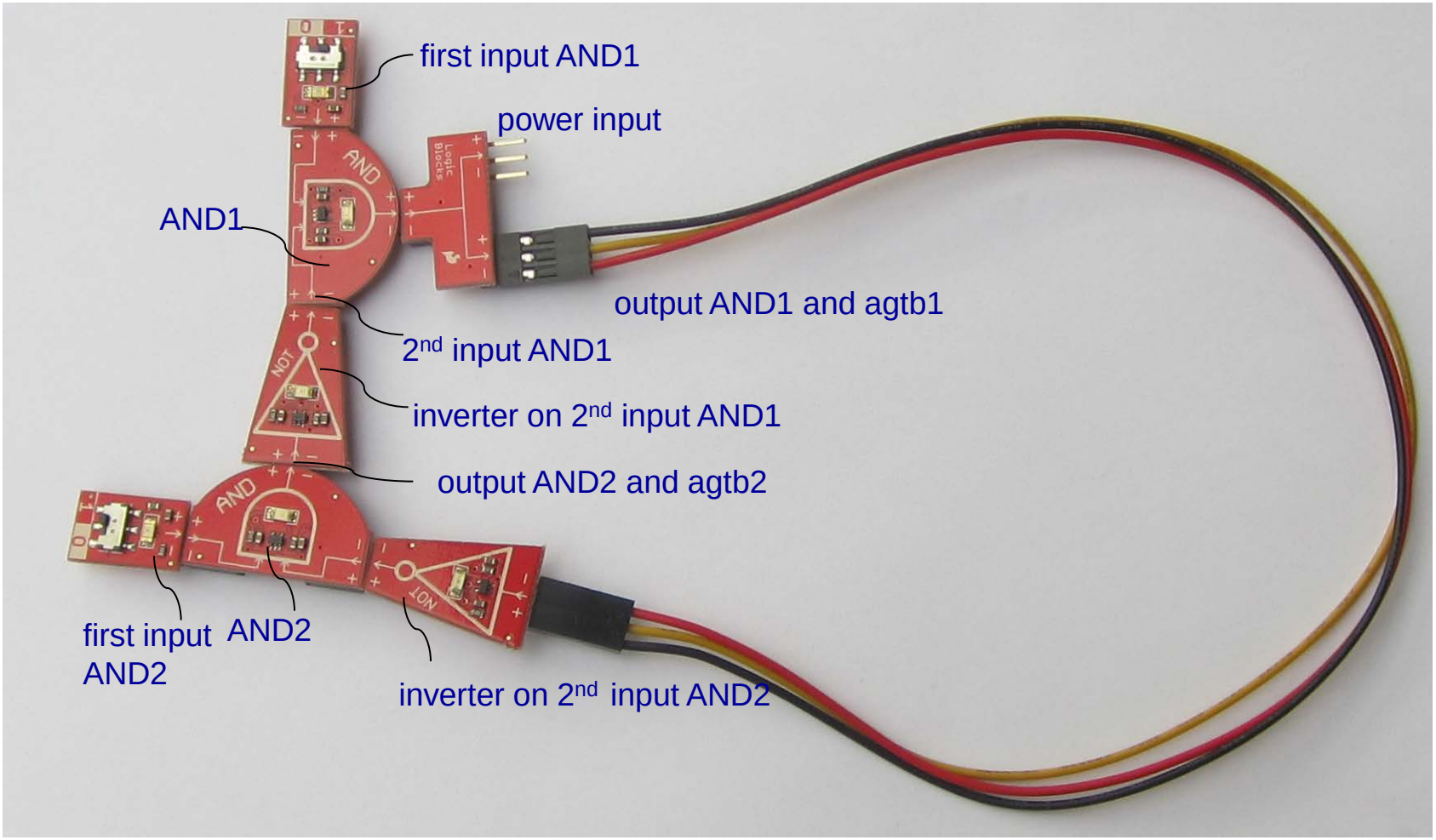
The a>b latch in Multimedia Logic



MultiMedia Logic is FreeWare available from http://www.softronix.com/logic.html

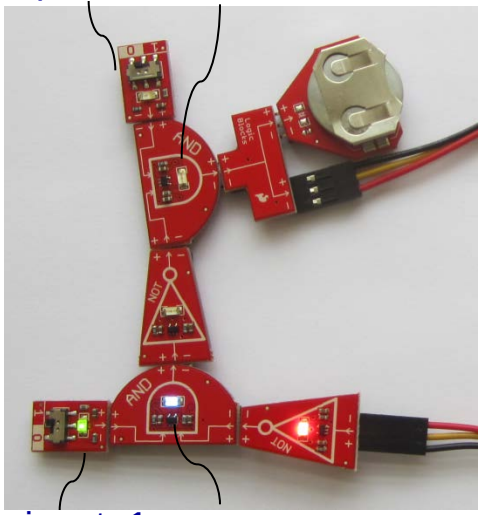
Ternarylogic

The a>b latch in actual circuitry using the Sparkfun LogicBlocks Kit



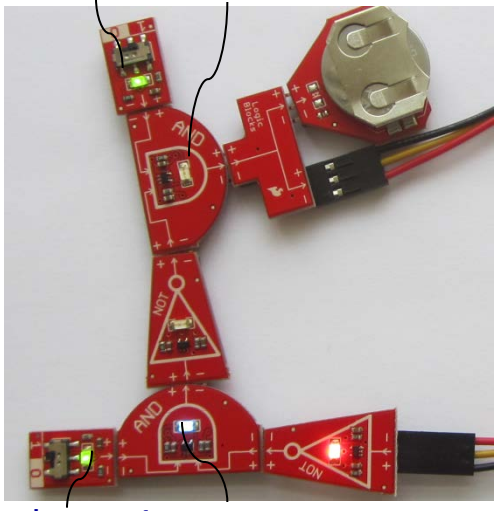
Ternarylogic

input=0 output=0



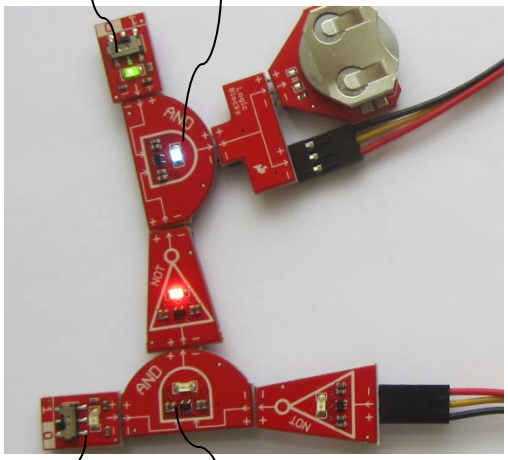
input=1 output=1

input=1 output=0



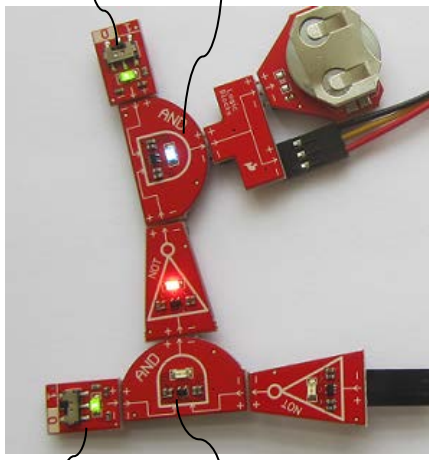
input=1 output=0

input=1 output=1



input=0 output=0

input=1 output=1



input=1 output=0

Ternarylogic

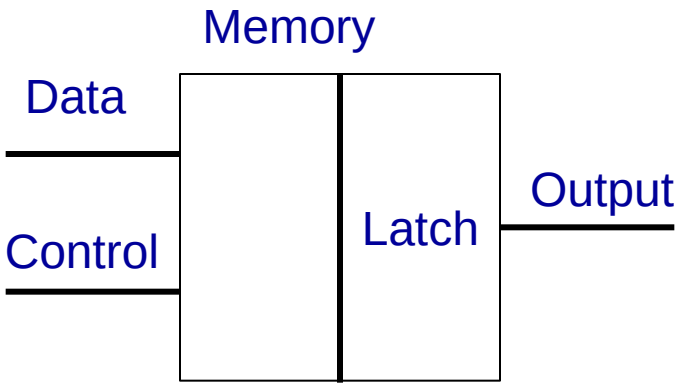
Operating the ‘agtb’ latch in

A: Multimedia Logic Simulation

and in

B: SparkFun LogicBlocks Kit

The problem with binary latches as memory

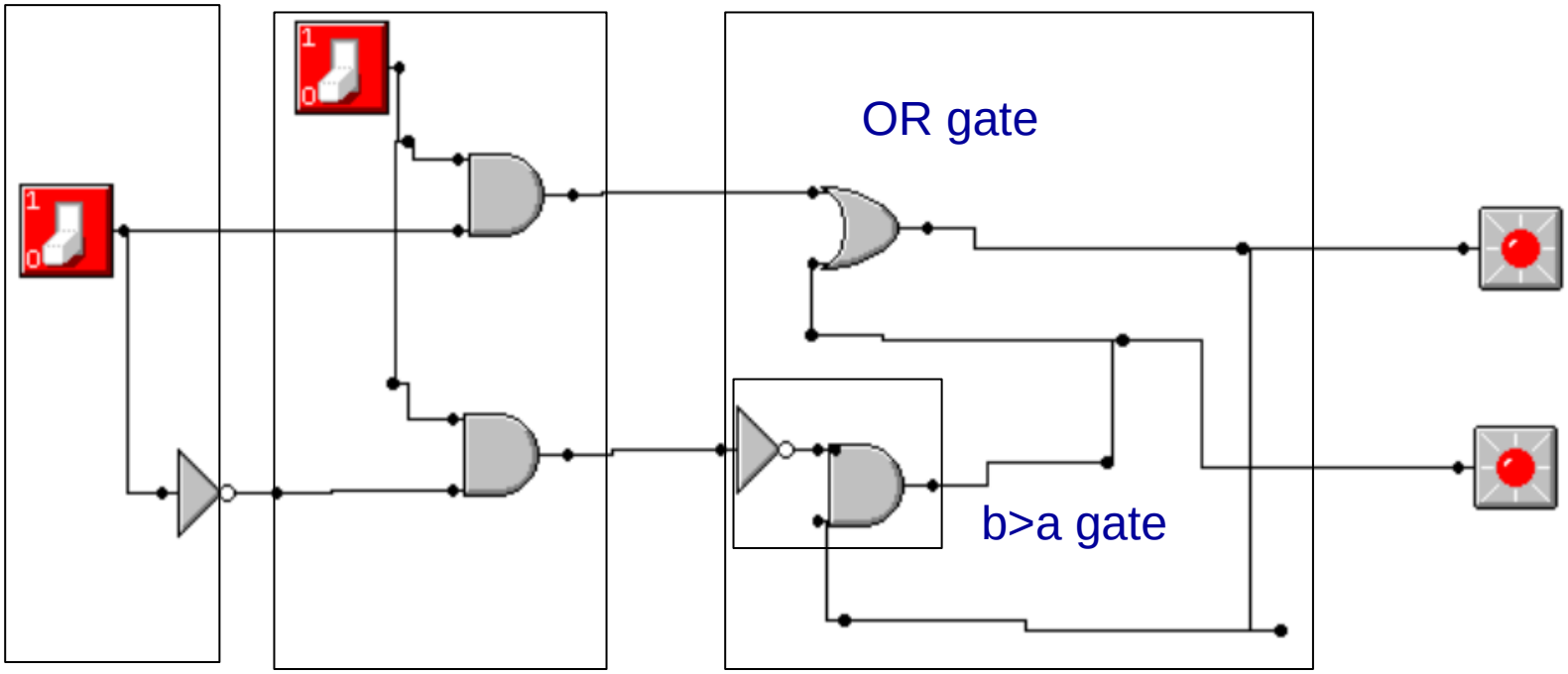


In one configuration it is desired to have a memory that will:

1) receive Data when a Control signal is ON
2) be insensitive to change when the Control signal is OFF
3) provides the content of the Memory on Output

A binary latch requires additional circuitry
A non-binary latch is a memory device

A Latch Memory



b>a table

$b > a$	0	1
0	0	1
1	0	0

Ternarylogic

State Table or Switching Table of ‘OR/bgta’ Latch

in1	in2		b0	a0		b1	b2	b3	b4	b5		a1	a2	a3	a4	a5
in1	in2		b0	a0		b1	b2	b3	b4	b5		a1	a2	a3	a3	a5
0	0	8	0	0	8	0	0	0	0	0	8	0	0	0	0	0
0	0	8	1	0	8	0	1	0	1	0	8	1	0	1	0	1
0	0	8	0	1	8	1	0	1	0	1	8	0	1	0	1	0
0	0	8	1	1	8	1	1	1	1	1	8	1	1	1	1	1
0	1	8	0	0	8	0	0	0	0	0	8	0	0	0	0	0
0	1	8	1	0	8	0	0	0	0	0	8	0	0	0	0	0
0	1	8	0	1	8	1	0	0	0	0	8	0	0	0	0	0
0	1	8	1	1	8	1	0	0	0	0	8	0	0	0	0	0
1	0	8	0	0	8	1	1	1	1	1	8	0	1	1	1	1
1	0	8	1	0	8	1	1	1	1	1	8	1	1	1	1	1
1	0	8	0	1	8	1	1	1	1	1	8	0	1	1	1	1
1	0	8	1	1	8	1	1	1	1	1	8	1	1	1	1	1
1	1	8	0	0	8	1	1	1	1	1	8	0	0	0	0	0
1	1	8	1	0	8	1	1	1	1	1	8	0	0	0	0	0
1	1	8	0	1	8	1	1	1	1	1	8	0	0	0	0	0
1	1	8	1	1	8	1	1	1	1	1	8	0	0	0	0	0

unstable states

Ternarylogic

OR/bgta latch in

A: Multimedia Logic as part of a memory

and in

B: SparkFun LogicBlocks Kit

Another Binary Model

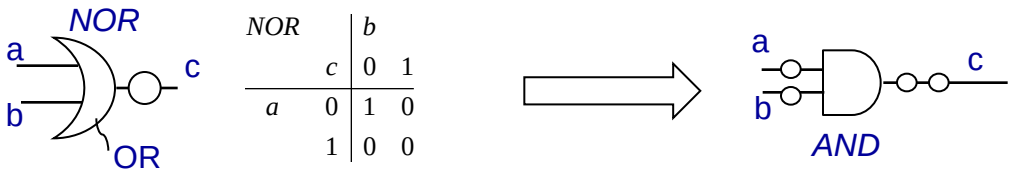
One can use another model to ‘explain’ other binary latch configurations:

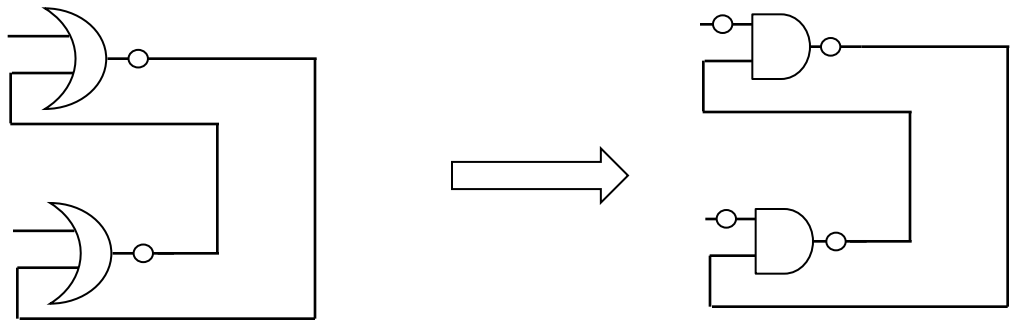
- consider functions as being formed from AND functions and inverters
- move outputs of the feedback devices around:
  1) in the NAND configuration the latch outputs are the outputs of the inverters at the ANDs
  2) in the ‘bgta’ latch configuration the latch outputs are outputs of the AND an the inputs of the inverters are the inputs to the ‘agtb’ devices

In the AND/inverter model all latches are modifications of one basic type: the NAND latch for instance.. It requires the deconstruction of the NAND into AND and inverter and rearrangement of ANDs and inverters, followed by reconstruction of the logic functions. This is not obvious and has not been done before.

The AND/inverter model does not explain the role of the “forbidden state,” the need to consider multiple switching cycles and it cannot be extend to design non-binary latches.

Equivalent Realization





NOR latch

The NOR latch can be built from a NAND latch with inverted inputs.

However, from a perspective of inputs and outputs the NOR latch and NAND latch are different devices.

Ternarylogic

Download

A copy of this presentation and of a Matlab program that also runs under Freemat are available from www.ternarylogic.com for zipped download. You can find the link at the bottom of the www.ternarlylogic.com website.

The zip file contains the Matlab programs:

latchbincomb.m which generates the switching tables for all 256 binary feedback combinations; and

makescn2.m which is automatically called inside latchbincomb.m to generate all 16 2 inputs/1 output binary logic functions.

The zip file also contains Multimedia Logic configuration files.