Presentation Slides

Chapters 1 and 2

A Sufficiently Expressive Logic

Logically Determined Design:

Clockless System Design With NULL Convention Logic

by Karl Fant

John Wiley & Sons, Inc.

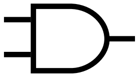
Introduce NULL Convention Logic

Diagrams by permission of John Wiley & Sons, Inc.

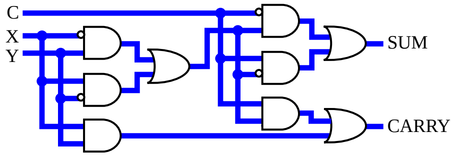
Boolean Logic is an Insufficient Logic

- The mathematical notion of the function is a stateless data mapping with no expression of coordination.

	T	F
T	T	F
F	F	F



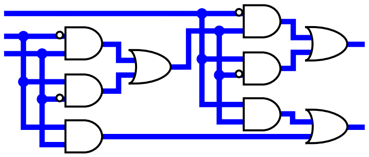
- A composition of functions forms a concurrent network.

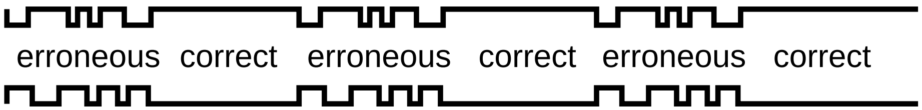


- The behavior of the network of functions is traditionally coordinated by a mathematician with a pencil who manages the flow of data and the instantiation of each function in its proper order.

- Without the coordinating mathematician Boolean logic is insufficiently expressive.

- Transitions race indiscriminately through the concurrent network causing a large number of erroneous values before the expression stabilizes to a correct resolution of the presented input





- There is no way to determine from the behavior of the network of functions when the output has stabilized to the correct resolution of presented data or even when new data has been presented.

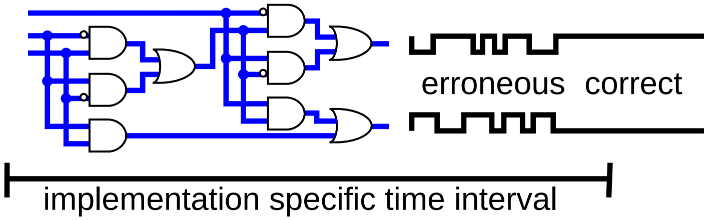
Boolean Logic Cannot Work Alone

The Crux of the Matter

The missing coordination behavior of the absent mathematician

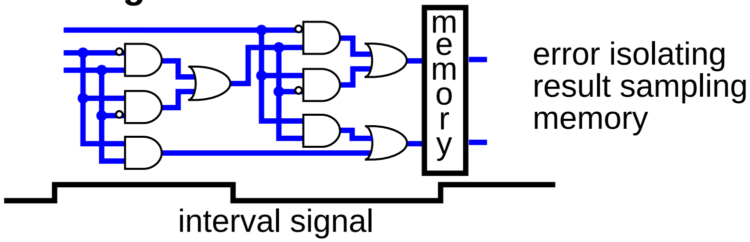
Supplement Boolean Logic

With stable input the circuit will eventually settle to a correct result.



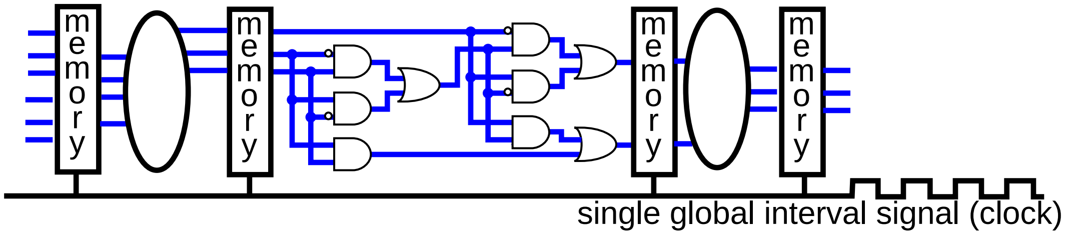
Express correct logical behavior with a time interval.

Logical behavior blurrs into a single timed behavior.



Ignore the erroneous behavior and sample the correct output at the end of the interval.

Further composition is in terms of identical time intervals and shared memories bounding the instantiation and resolution of each logic expression.



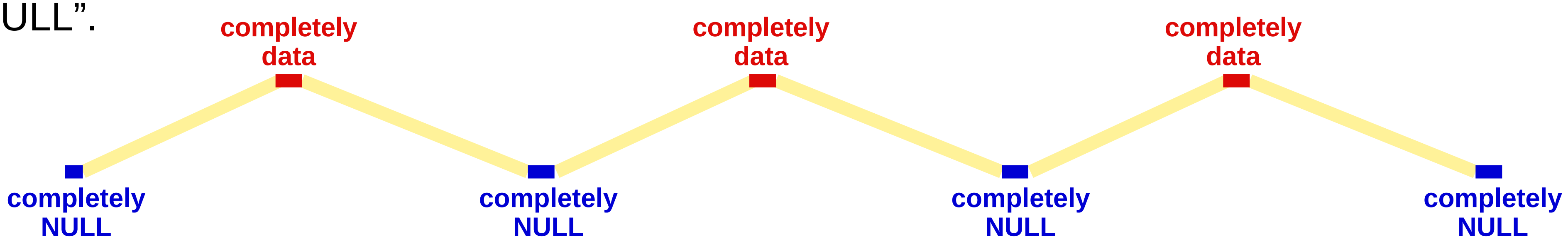
Coordination is restored with the imposition of a non-logical expression of time and memory

Not the mathematician’s coordination behavior, but the scheme can be made to work with sufficient engineering attention.

Define A Sufficient Logic

The NULL Convention

We assume a representation of data with two disjoint value domains, one expressing “data” and one explicitly expressing “not data”, which we will call NULL. A data path presents successive data sets by monotonically transitioning between “completely data” and “completely NULL”.



The Completeness Criterion

An operation;

- transitions its output to DATA only when its input is “completely DATA,
- transitions its output to NULL only when its input is “completely NULL” and
- maintains its output when its input is neither “completely DATA nor “completely NULL”

A two input operator will transition its output to D only when both inputs are D and transition its output to N only when both inputs are N. Otherwise it will not transition its output –.

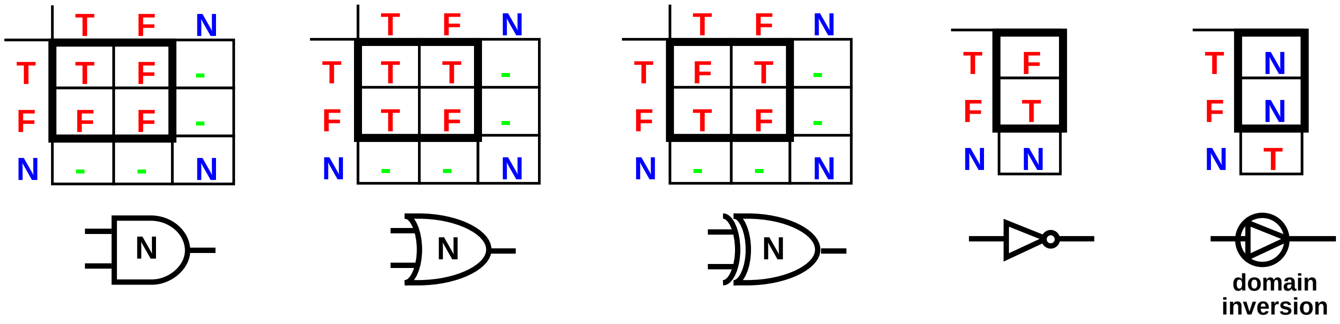
	D	N
D	D	–
N	–	N

The transition of the output implies the completeness of the input.

NULL Convention Logics

3 value NULL convention Logic: 3NCL

The completeness criterion applied to Boolean Logic results in a three value logic



	T	F	N
T	T	F	-
F	F	F	-
N	-	-	N

	T	F	N
T	T	T	-
F	T	F	-
N	-	-	N

	T	F	N
T	F	T	-
F	T	F	-
N	-	-	N

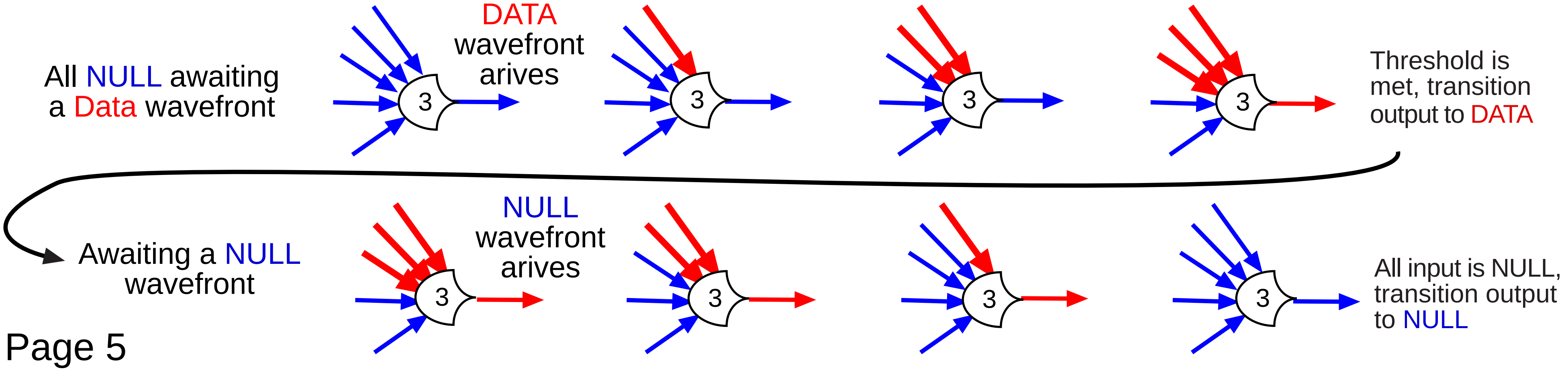
T	F
F	T
N	N

T	N
F	N
N	T

2 value NULL convention Logic: 2NCL

One data value and NULL With only one data value the only discriminable property when presenting input data is how many data values are presented so 2NCL is a threshold logic with state holding behavior.

When the number of input data values matches the threshold the threshold operator transitions its output to DATA.
When all input values become NULL the threshold operator transitions its output to NULL.



All NULL awaiting a Data wavefront

DATA wavefront arives

Threshold is met, transition output to DATA

Awaiting a NULL wavefront

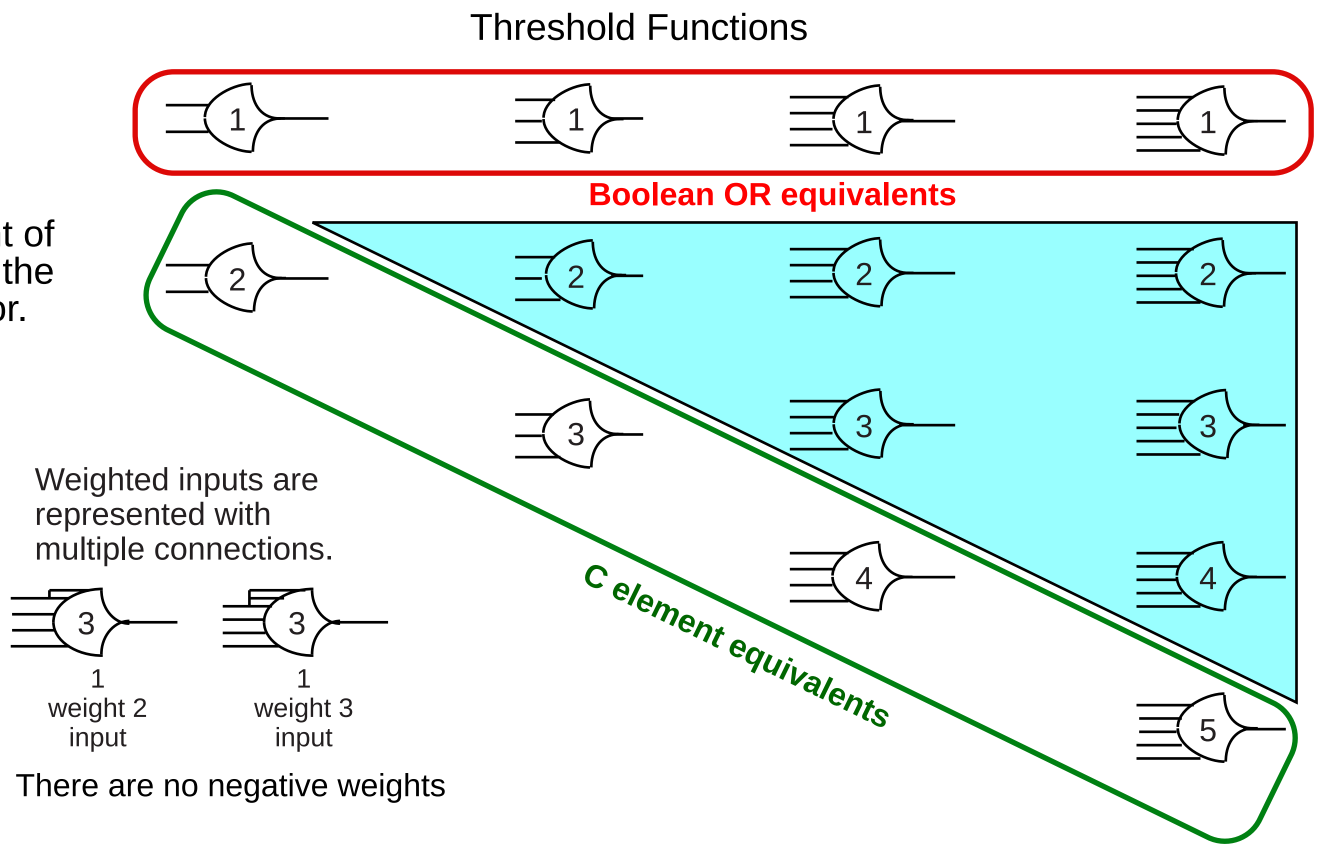
NULL wavefront arives

All input is NULL, transition output to NULL

NULL Convention Logic

The number inside the operator is the threshold for the number of indicated inputs.

There is no equivalent of the logical inverter or the Boolean AND operator.



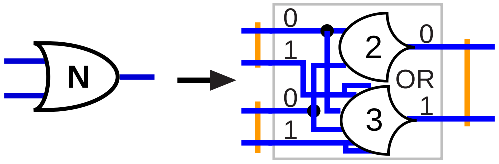
Weighted inputs are represented with multiple connections.

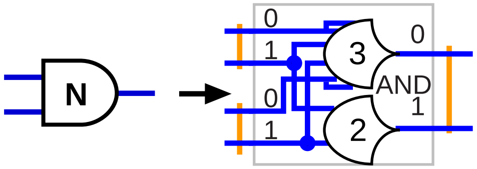
There are no negative weights

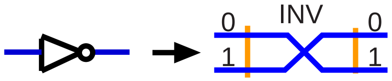
NCL Example Circuits

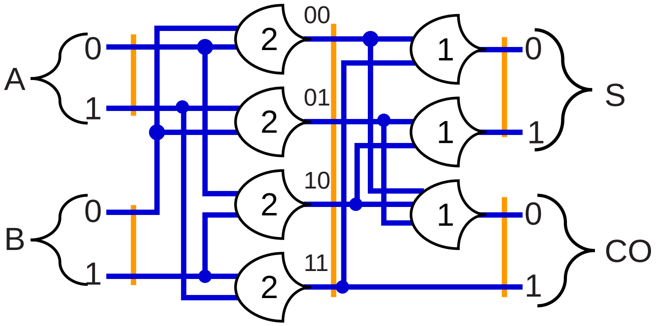
2NCL is a subvariable logic operating directly
on values between variable boundaries.

Variable boundary |

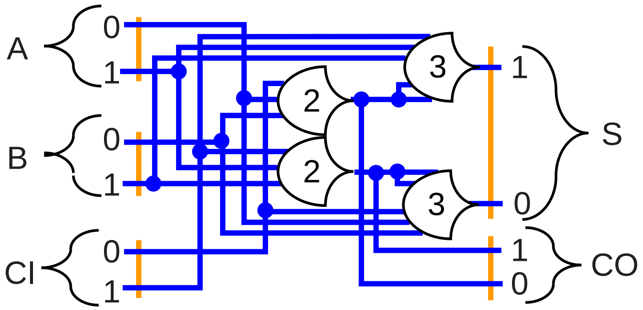




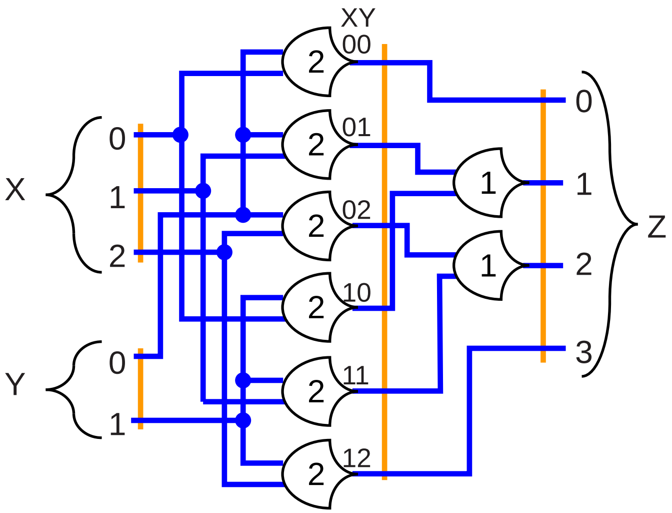




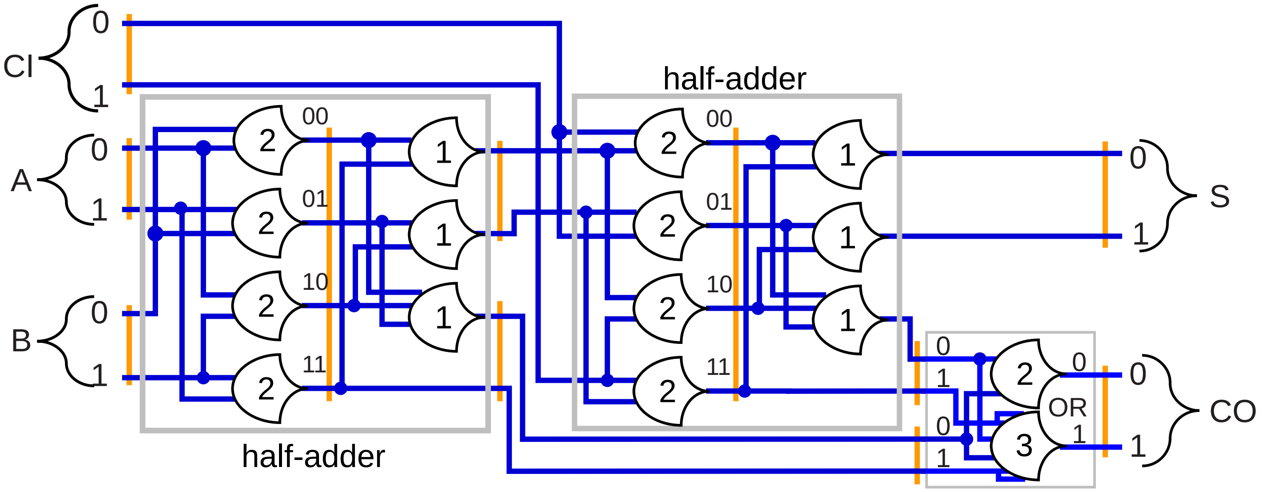
Binary half-adder



Binary full-adder



Binary-trinary-quaternary adder



Binary full-adder as two half-adders and OR

Multi-Rail/Multi-Value Variables

The most primitive element of a data path, a single wire, can assert only two values. One value must represent NULL so there can only be one data value which we will call DATA. With only one data value, an M value variable is expressed with M wires only one of which will express its DATA value at a time.

Wire numeric base 2 meanings

Wire	NULL	0	1
#1	N	D	N
#2	N	N	D

Wire numeric base 4 meanings

Wire	NULL	0	1	2	3
#1	N	D	N	N	N
#2	N	N	D	N	N
#3	N	N	N	D	N
#4	N	N	N	N	D

Wire numeric base 10 meanings

Wire	N	1	2	3	4	5	6	7	8	9	0
#1	N	D	N	N	N	N	N	N	N	N	N
#2	N	N	D	N	N	N	N	N	N	N	N
#3	N	N	N	D	N	N	N	N	N	N	N
#4	N	N	N	N	D	N	N	N	N	N	N
#5	N	N	N	N	N	D	N	N	N	N	N
#6	N	N	N	N	N	N	D	N	N	N	N
#7	N	N	N	N	N	N	N	D	N	N	N
#8	N	N	N	N	N	N	N	N	D	N	N
#9	N	N	N	N	N	N	N	N	N	D	N
#10	N	N	N	N	N	N	N	N	N	N	D

Wire Logical meanings

Wire	NULL	TRUE	FALSE
#1	N	D	N
#2	N	N	D

Wire general meanings

Wire	NULL	Animal	Vegetable	Mineral
#1	N	D	N	N
#2	N	N	D	N
#3	N	N	N	D

Wire control meanings

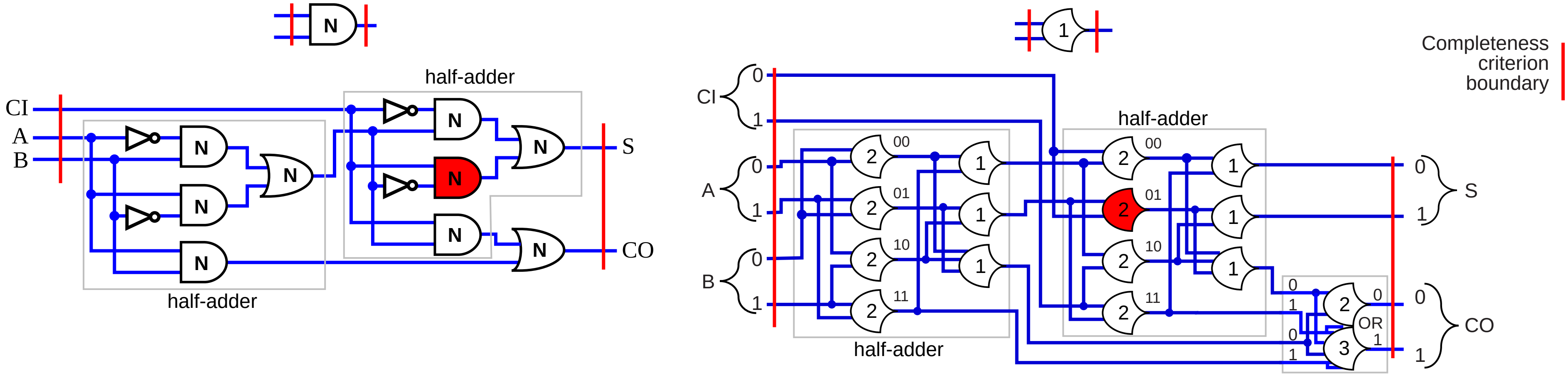
Wire	NULL	Select A	Select B	Select C
#1	N	D	N	N
#2	N	N	D	N
#3	N	N	N	D

Wire other meanings

Wire	NULL	First	Second	Third	Fourth
#1	N	D	N	N	N
#2	N	N	D	N	N
#3	N	N	N	D	N
#4	N	N	N	N	D

Completeness Behavior Composes

An NCL expression as a whole exhibits the completeness criterion.



Orderly collective behavior follows from orderly individual behavior.

When the output monotonically transitions to “completely data”, it means that the input is “completely data” and the data output is the correct result of the presented input.

When the output monotonically transitions to “completely NULL”, it means that the input is “completely NULL” and the NULL has propagated through the circuit.

It does not matter in what order the values transition at the input of the expression. Nor does it matter how long transitions take to propagate within the expression.

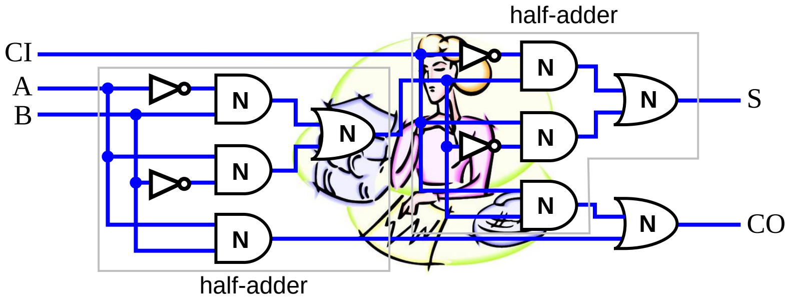
The expression as a whole:

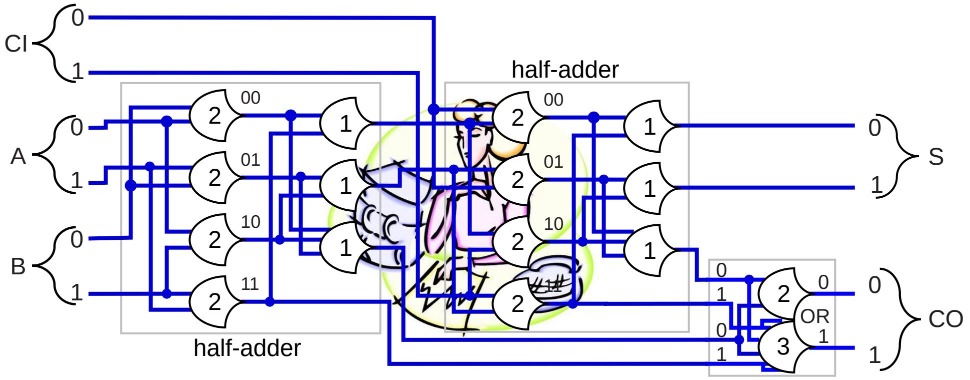
- recognizes when new input is presented,
- announces when it is done with the resolution of presented input and
- announces when it is ready to receive a new data presentation.

NCL is a Fully Sufficient Logic

An NCL expression is complete and sufficient in itself: Purely in terms of logical relationships

No supplementary expression such as a time interval, memory, controller or mathematician is required.





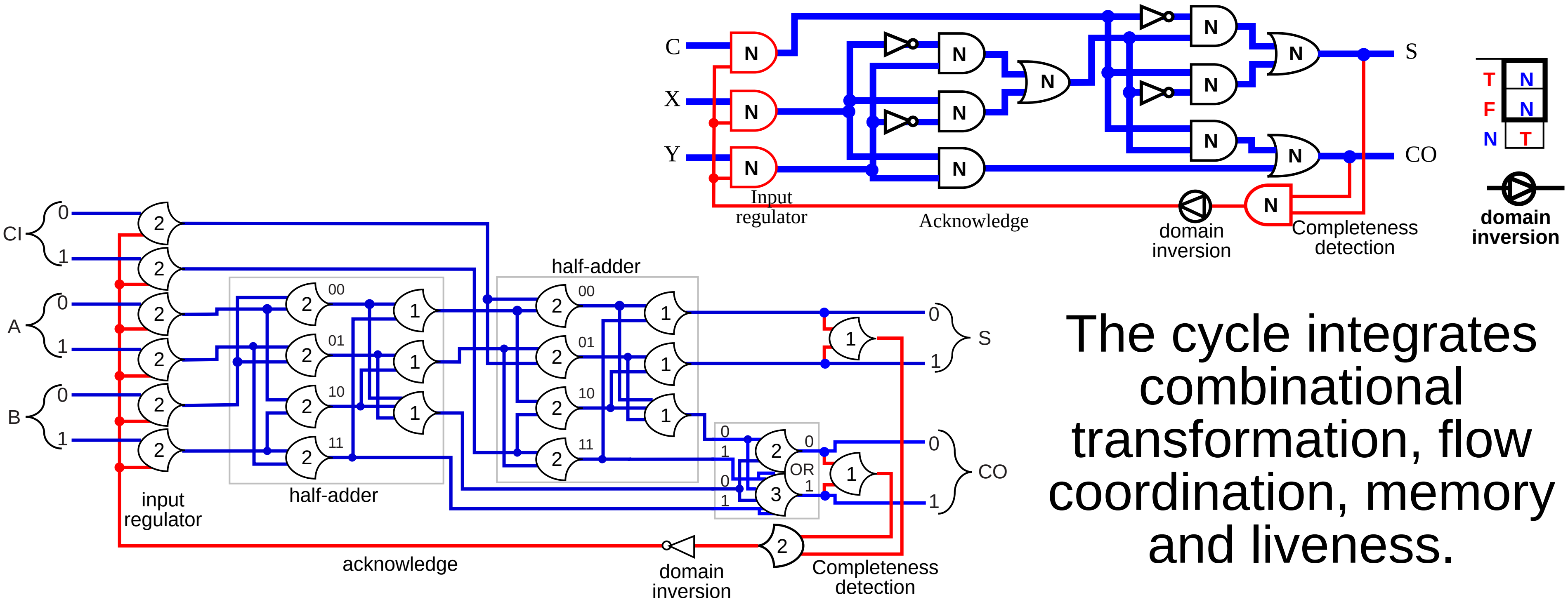
The coordinating behavior of the mathematician and her pencil are fully restored. The expression behaves exactly as if a mathematician was coordinating it.

No races, no hazards, no glitches, no indeterminate behavior.

The behavior of the expression, fully determined by the logic, is reliable, repeatable, testable and trustable.

Self Coordination: The Cycle

Any expression expressing the completeness criterion can coordinate its own input with a feedback acknowledge signal generated from its own output completeness. The inversion in the feedback creates an oscillator continually striving to transition between complete data and complete NULL.



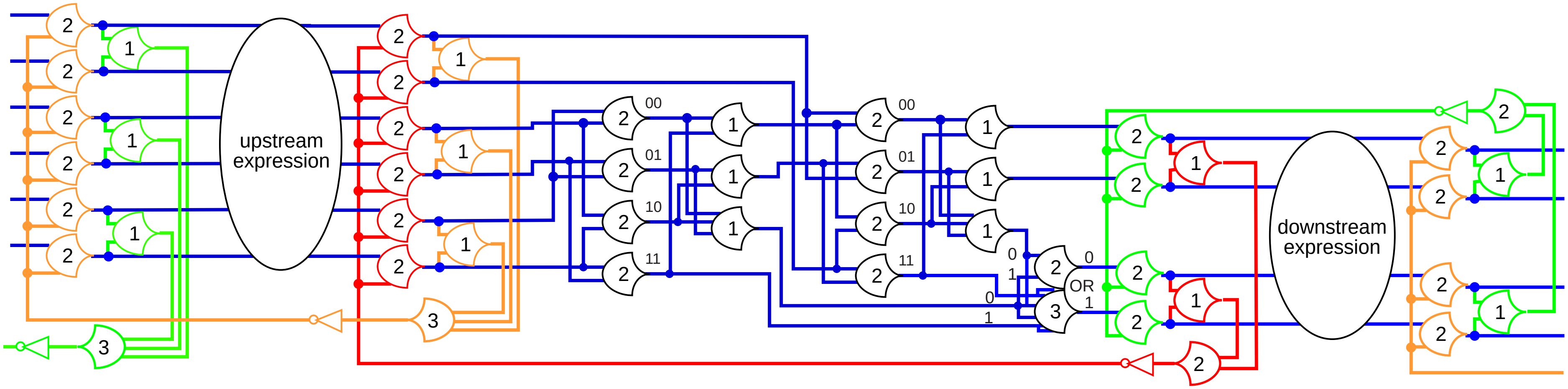
The cycle integrates combinational transformation, flow coordination, memory and liveness.

When acknowledge is DATA, the input regulator will pass and maintain data values.
When acknowledge is NULL, the input regulator will pass and maintain NULL values.

No new concepts or behaviors have been introduced.
The expressions are still purely in terms of logical relationships

Composing With Coupled Cycles

Cycles can be coupled such that the completeness detection of a presenting cycle is after the input regulation of a receiving cycle.



The presenting cycle will maintain its output wavefront until it detects output completeness which will occur strictly after the receiving cycle has accepted the wavefront and is stably maintaining it.

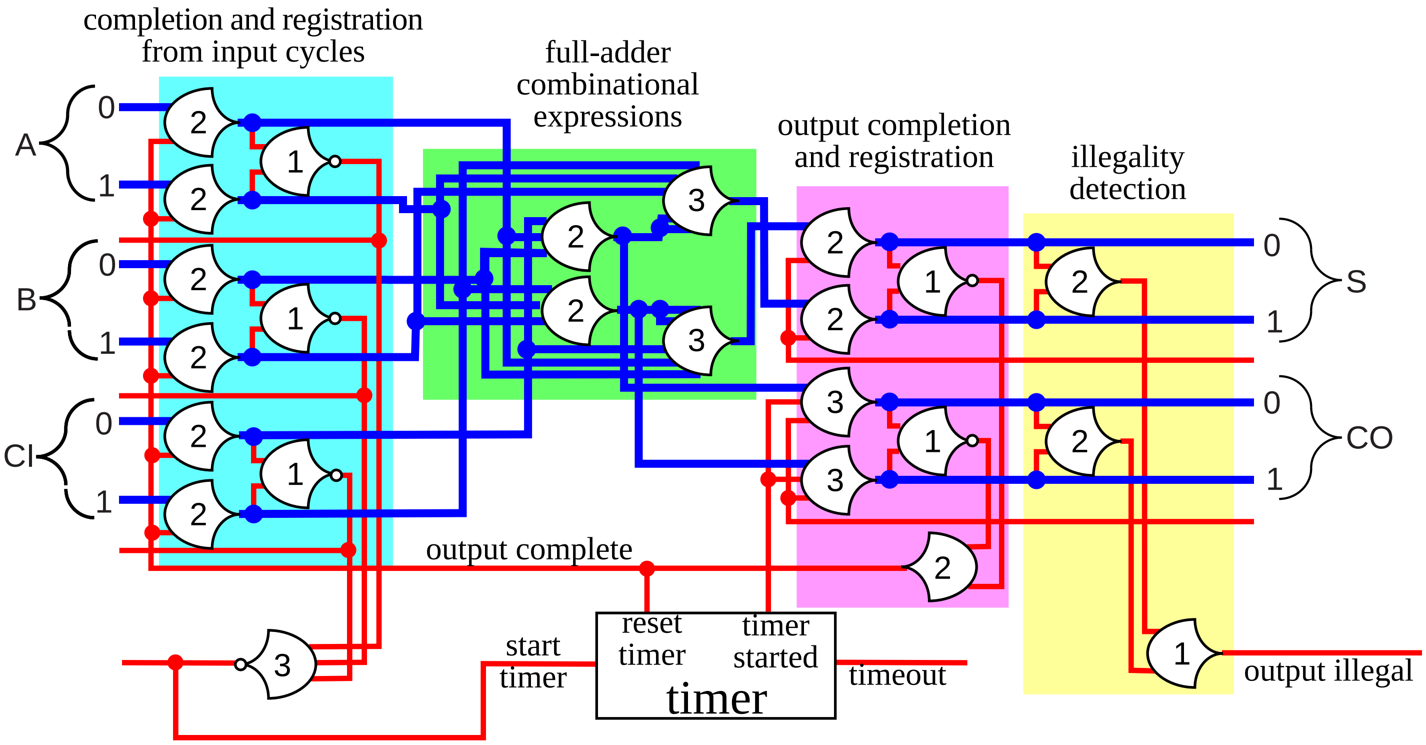
Wavefronts are handed stably from cycle to cycle.

Complete systems can be composed as a structure of coupled cycles expressed **purely in terms of local logical relationships.**

An NCL circuit cannot tell a lie

At any variable completeness boundary an NCL expression will

- present a correct result
- indicate that it is presenting an incorrect result by presenting more than one DATA value per variable which is logically detectable
- fail to achieve completeness



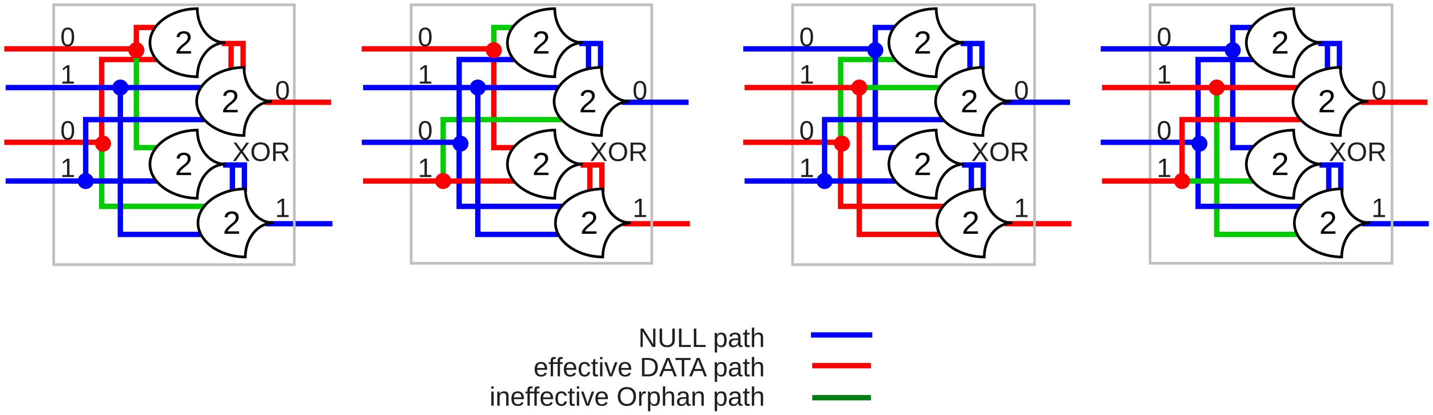
The 2NCL Orphan

An effective data path participates in generating the next variable and is always fully delay insensitive

An orphan is an ineffective data path that branches off an effective data path internal to the circuit and does not participate in generating the result variable. It is called an orphan because it loses all its relationships with other signals.

It cannot be logically determined that the current wavefront has propagated over an orphan before the next wavefront arrives. Therefore, the expression must include an assumption that all orphan paths propagate before a next wavefront can arrive.

Because an orphan does not participate in generating the next variable an orphan does not cross a variable boundary consequently it is always isolated between variable (completeness) boundaries.

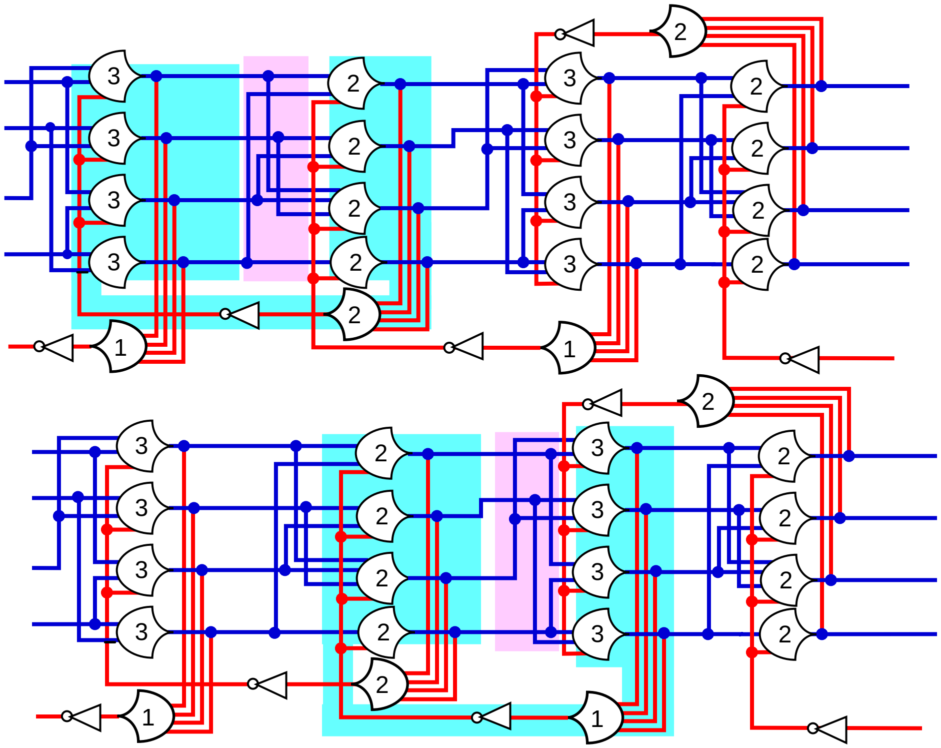


The NCL Orphan in the Cycle

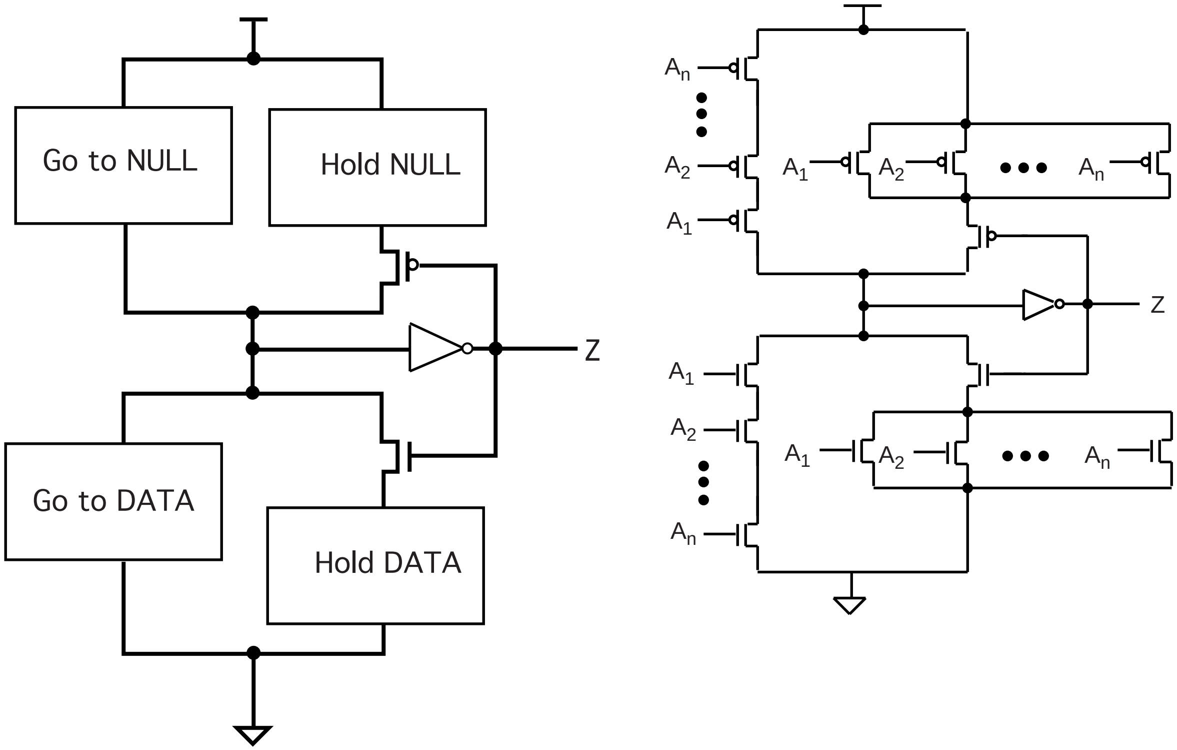
Every NCL circuit resides in a cycle and the cycle period determines when a next wavefront can arrive.

An oprhan path must propagate strictly faster than the period of its cycle

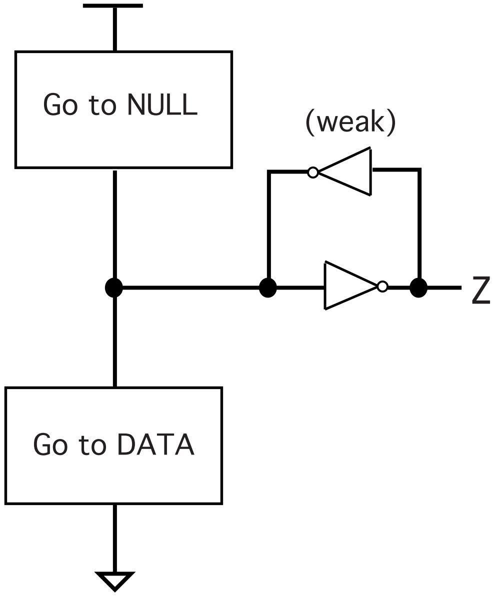
The branches within the pink must propagate strictly faster that the cycle paths under the blue.



Generic Transistor Design Templates

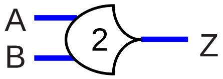


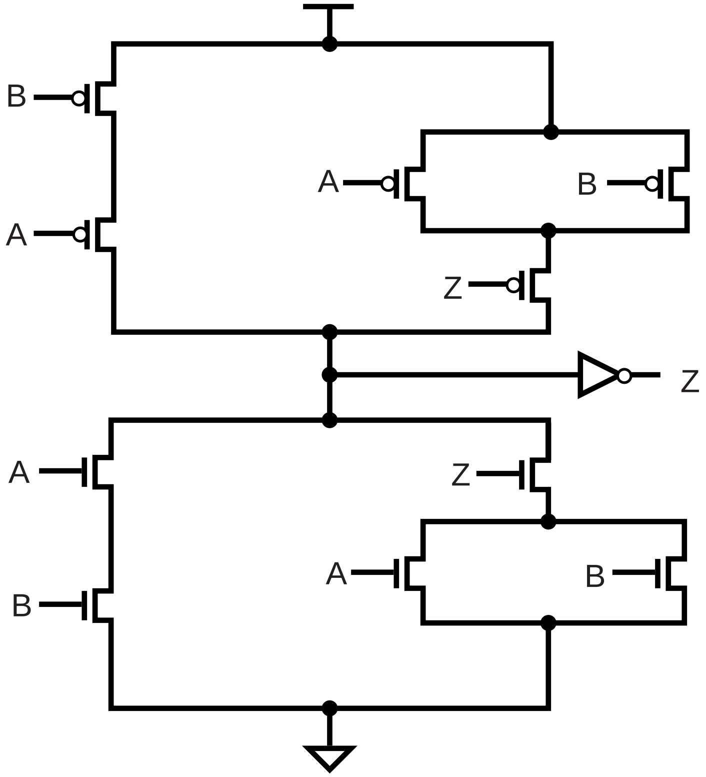
Generic static CMOS implementation

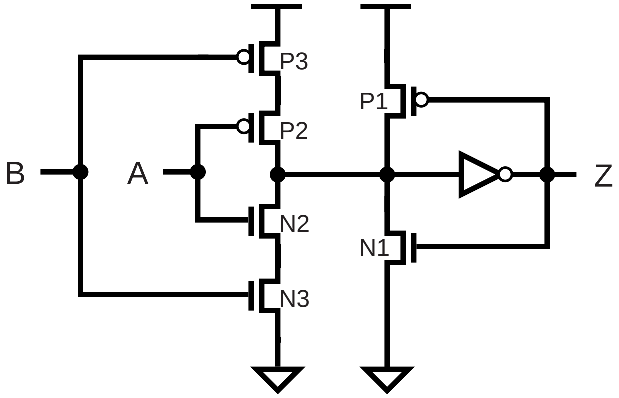


Generic semi-static CMOS implementation

2 of 2 Operator Transistor Design



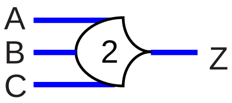


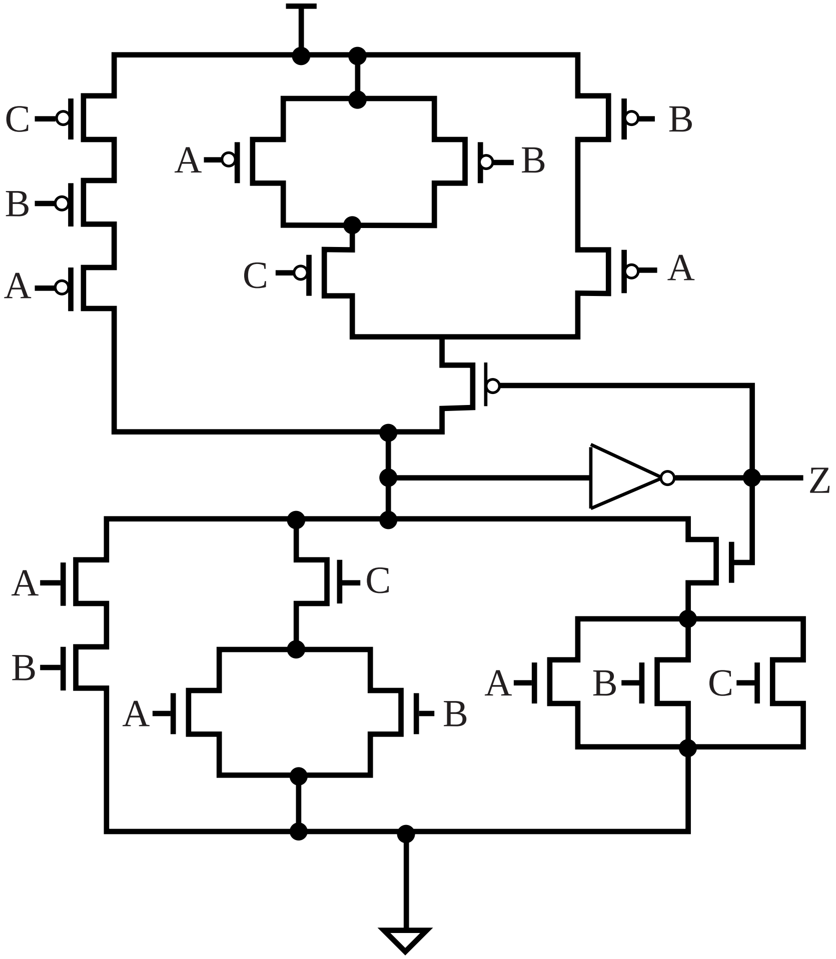


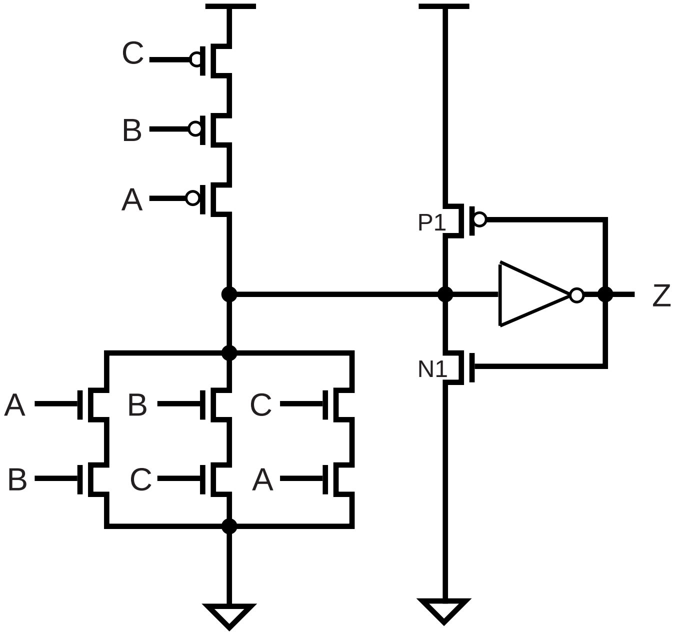
Static CMOS implementation
Faster, lower power, easier to design and fab.

Semi-static CMOS implementation
Fewer transistors

2 of 3 Operator Transistor Design







Static CMOS implementation

Semi-static CMOS implementation

This page intentionally blank

This page intentionally blank