Sequential Logic

Pat Hanrahan

CS448H: Agile Hardware Design
Winter 2017

Combinational Logic

Combinational logic is an expression involving (combination of) logic gates

1. **All inputs connected to exactly one output**
2. **No cycles**

Expression: Fixed topology and fixed data widths

***Pure* and *total* function of inputs**

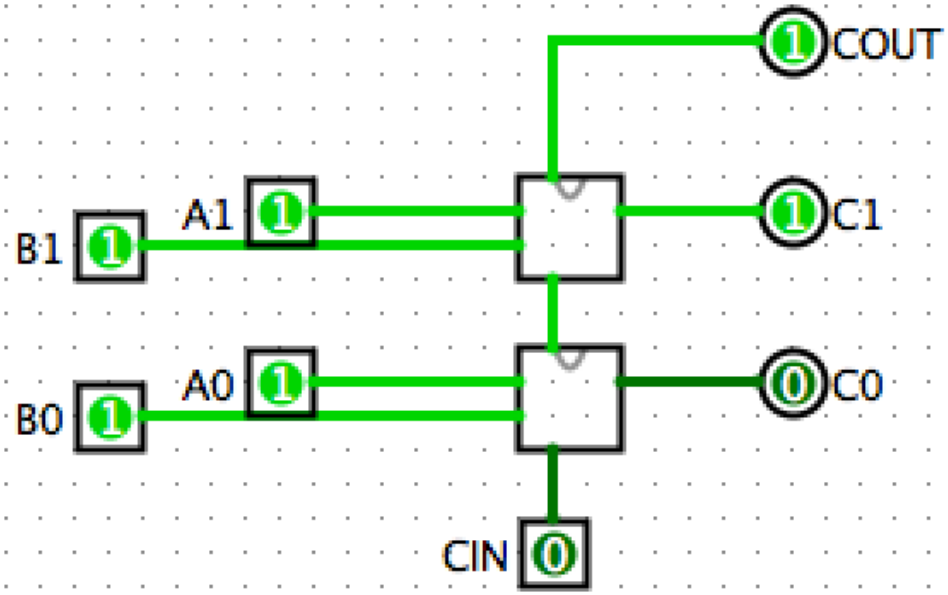
Executes continuously and in parallel

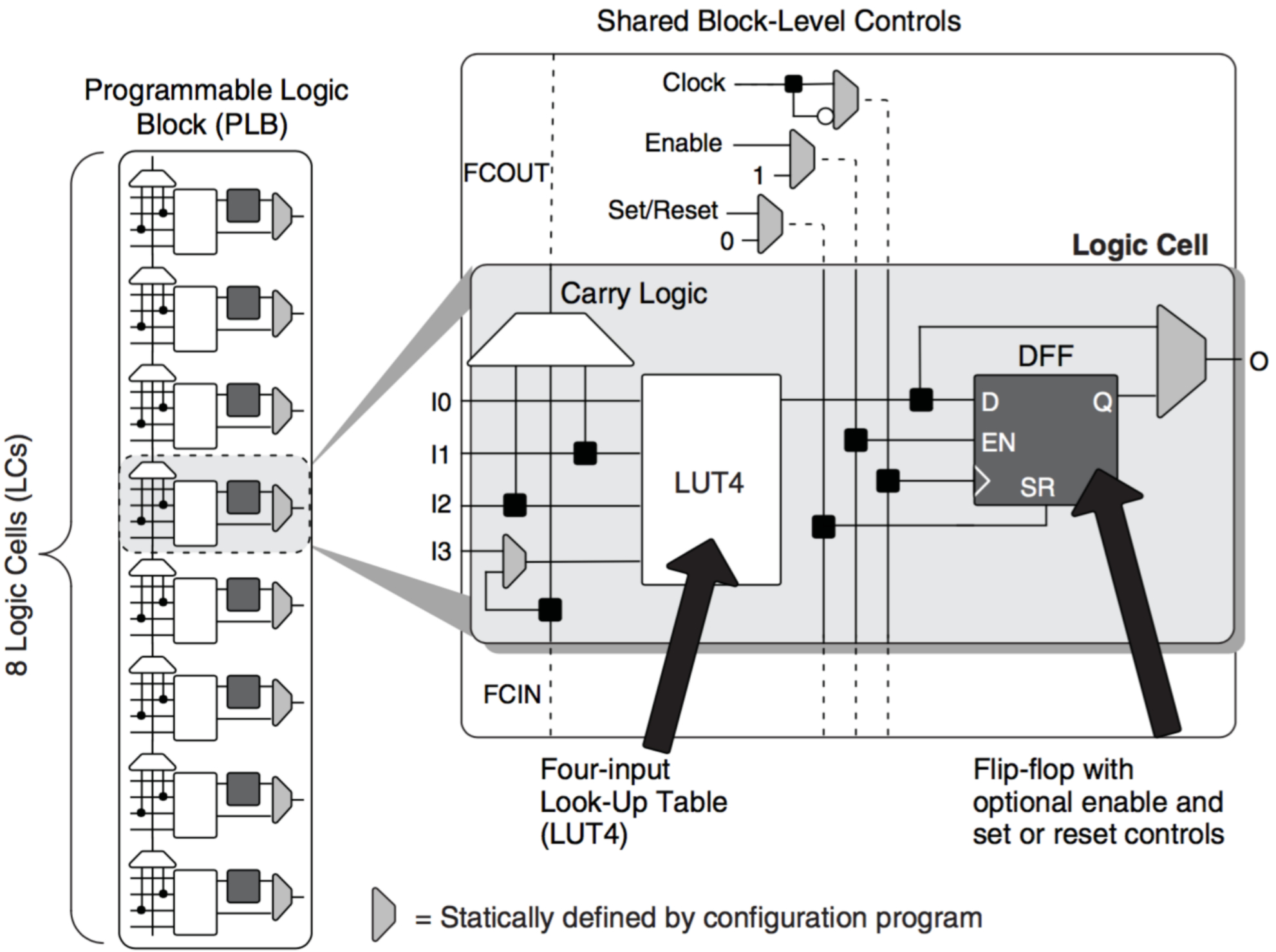
```
// Truth table for Mux2
S A B O
0 0 0 0
0 0 1 0
0 1 0 1
0 1 1 1
1 0 0 0
1 0 1 1
1 1 0 0
1 1 1 1
```

```
// Truth table for Mux2 - Don't Care
S A B O
0 0 X 0
0 0 X 0
0 1 X 1
0 1 X 1
1 X 0 0
1 X 1 1
1 X 0 0
1 X 1 1
```

```
// Truth table for Mux2 - Don't care
S A B O
0 0 X 0
0 1 X 1
1 X 0 0
1 X 1 1
```

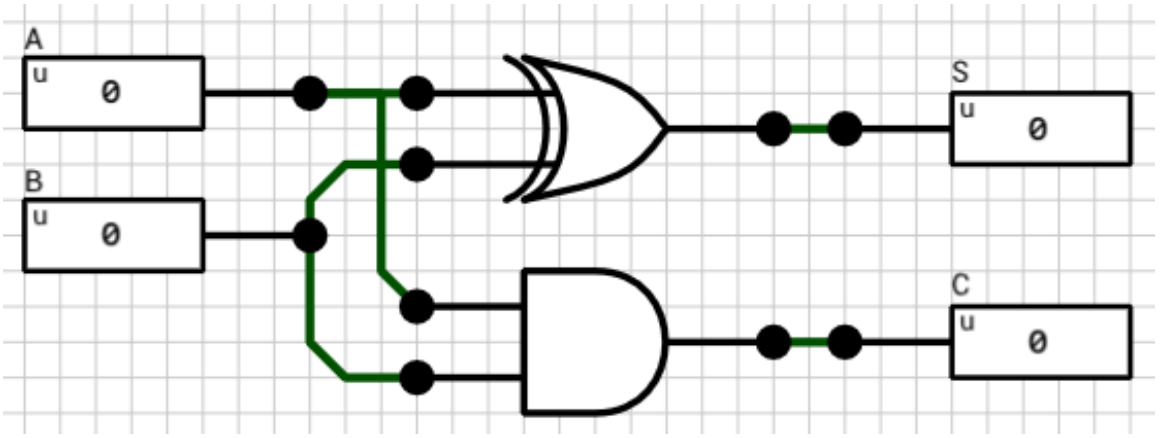
Add2.circ

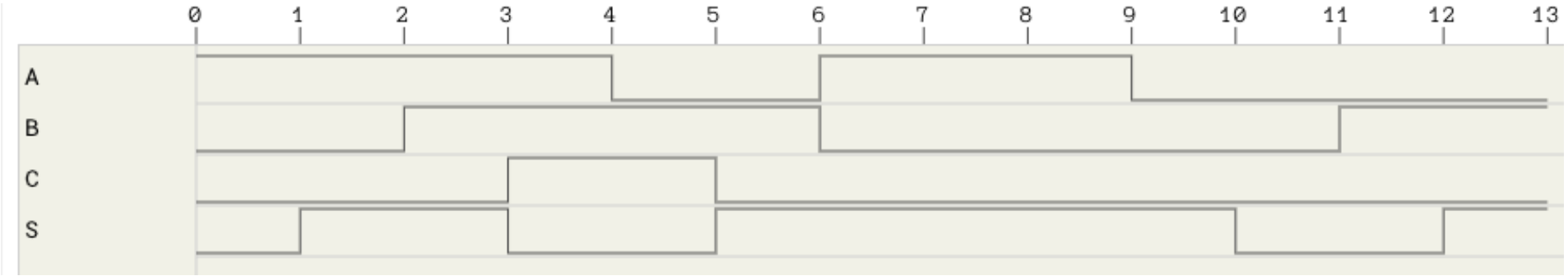



Shared Block-Level Controls
Programmable Logic Block (PLB)
Clock
Enable
1
Set/Reset
0
FCOUT
Logic Cell
Carry Logic
DFF
O
I0
I1
I2
I3
LUT4
D
Q
EN
SR
FCIN
8 Logic Cells (LCs)
Four-input Look-Up Table (LUT4)
Flip-flop with optional enable and set or reset controls
= Statically defined by configuration program

iceadd2.v

HalfAdder.circ

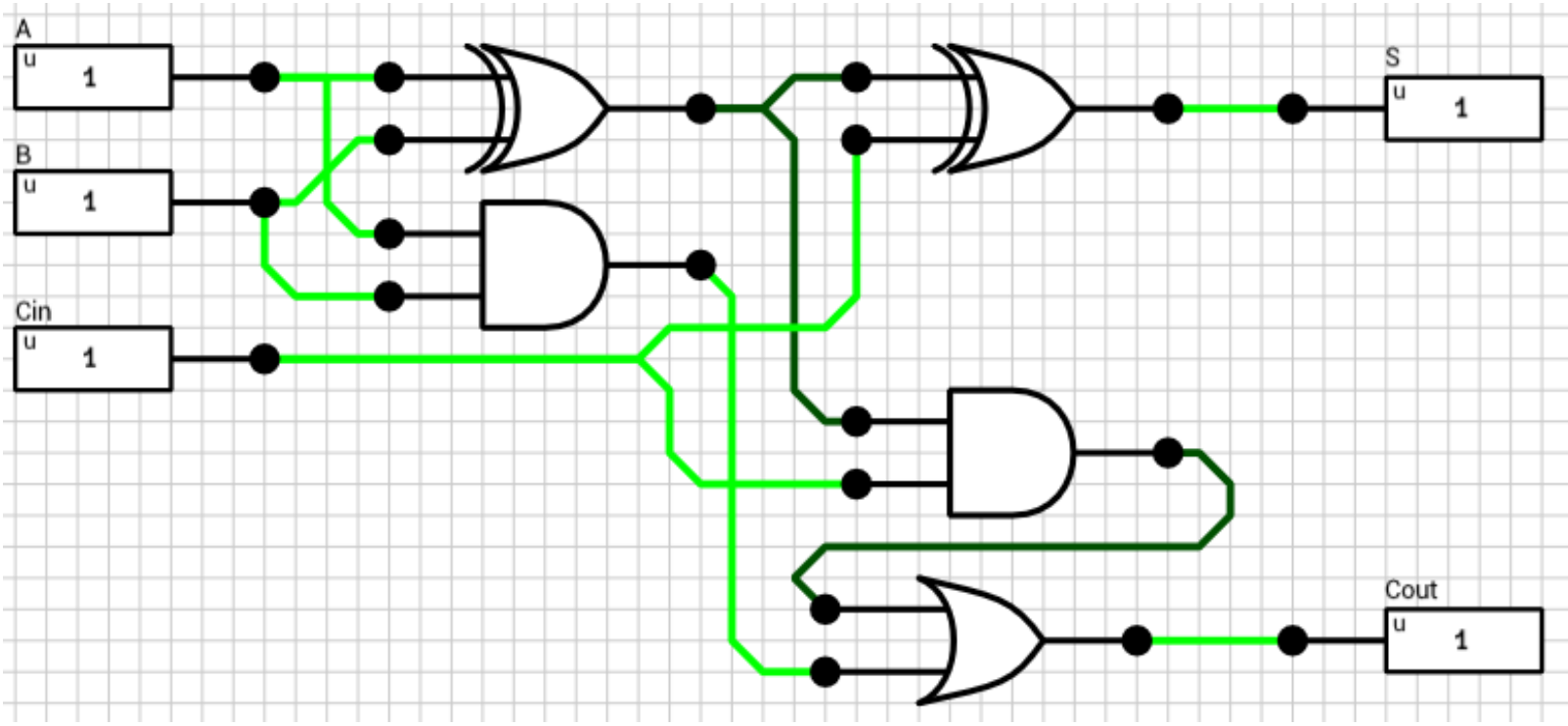


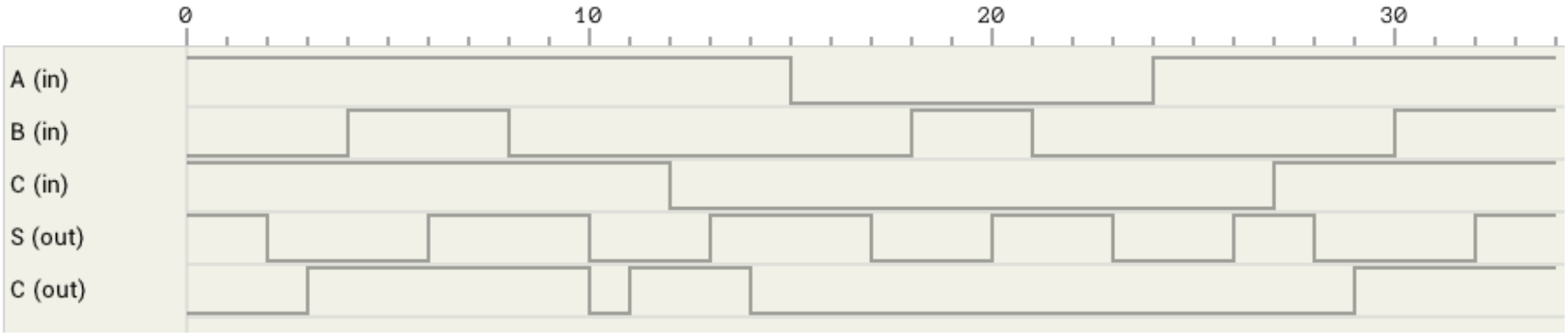


Assuming each gate has the same delay T

http://dls.makingartstudios.com/tags/adders/index.html

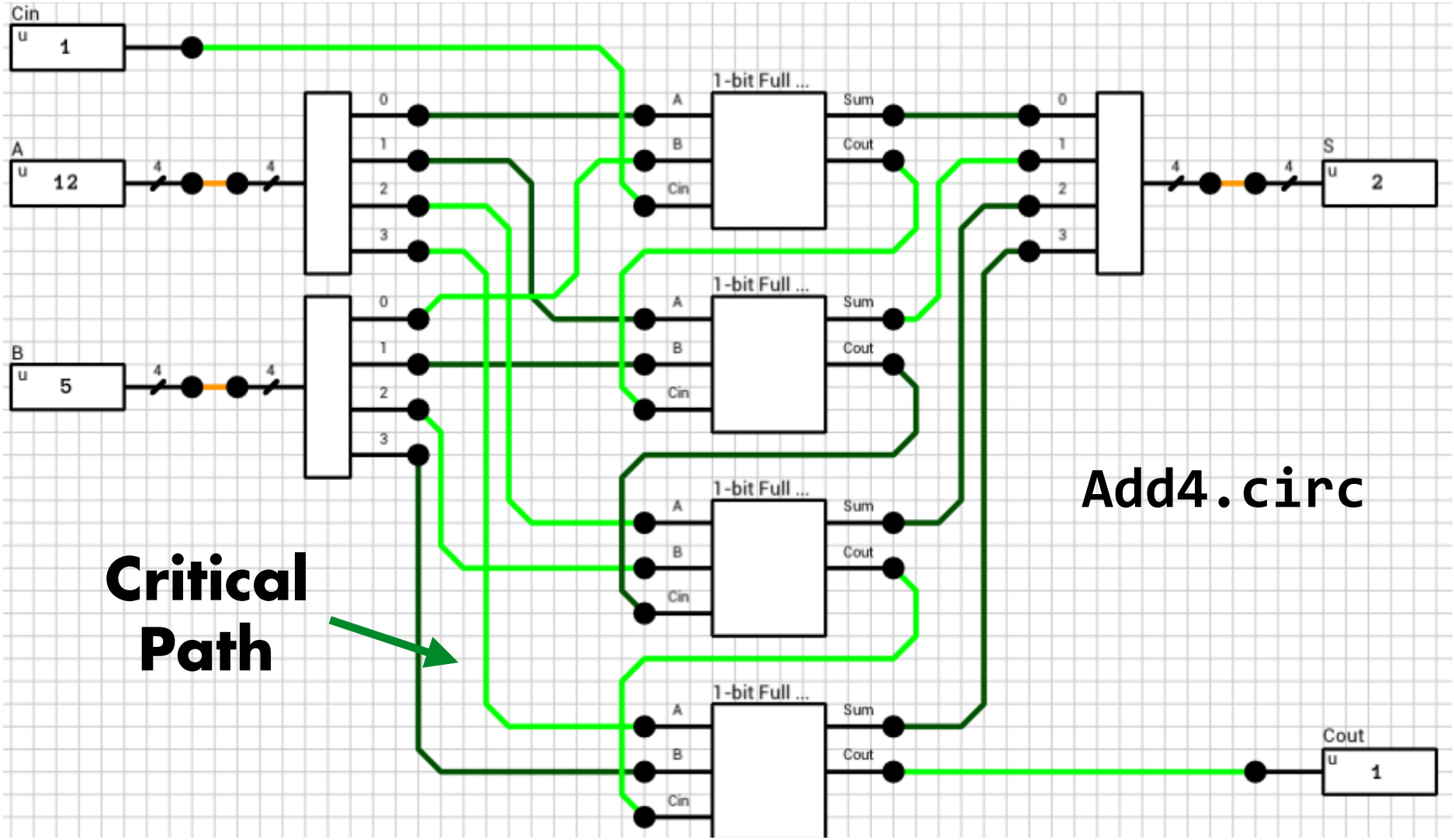
FullAdder.circ

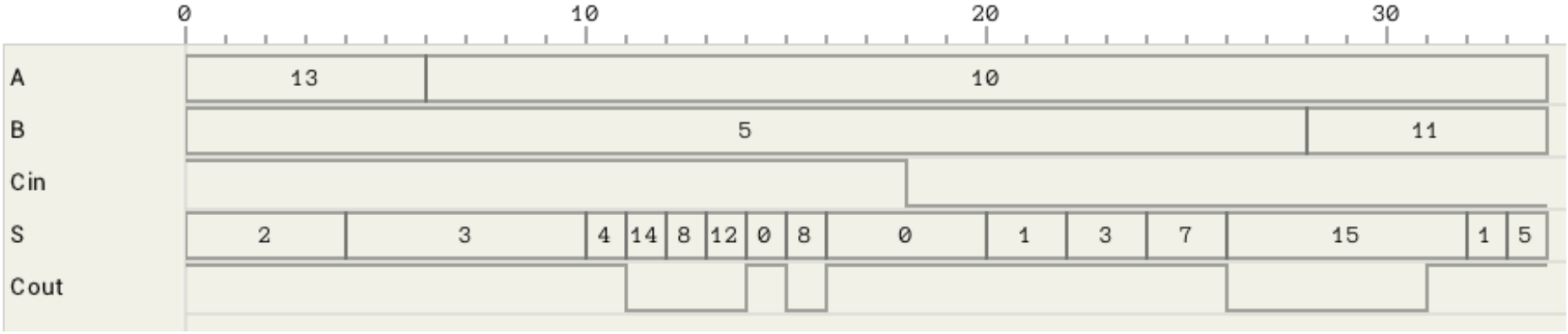




Glitches!

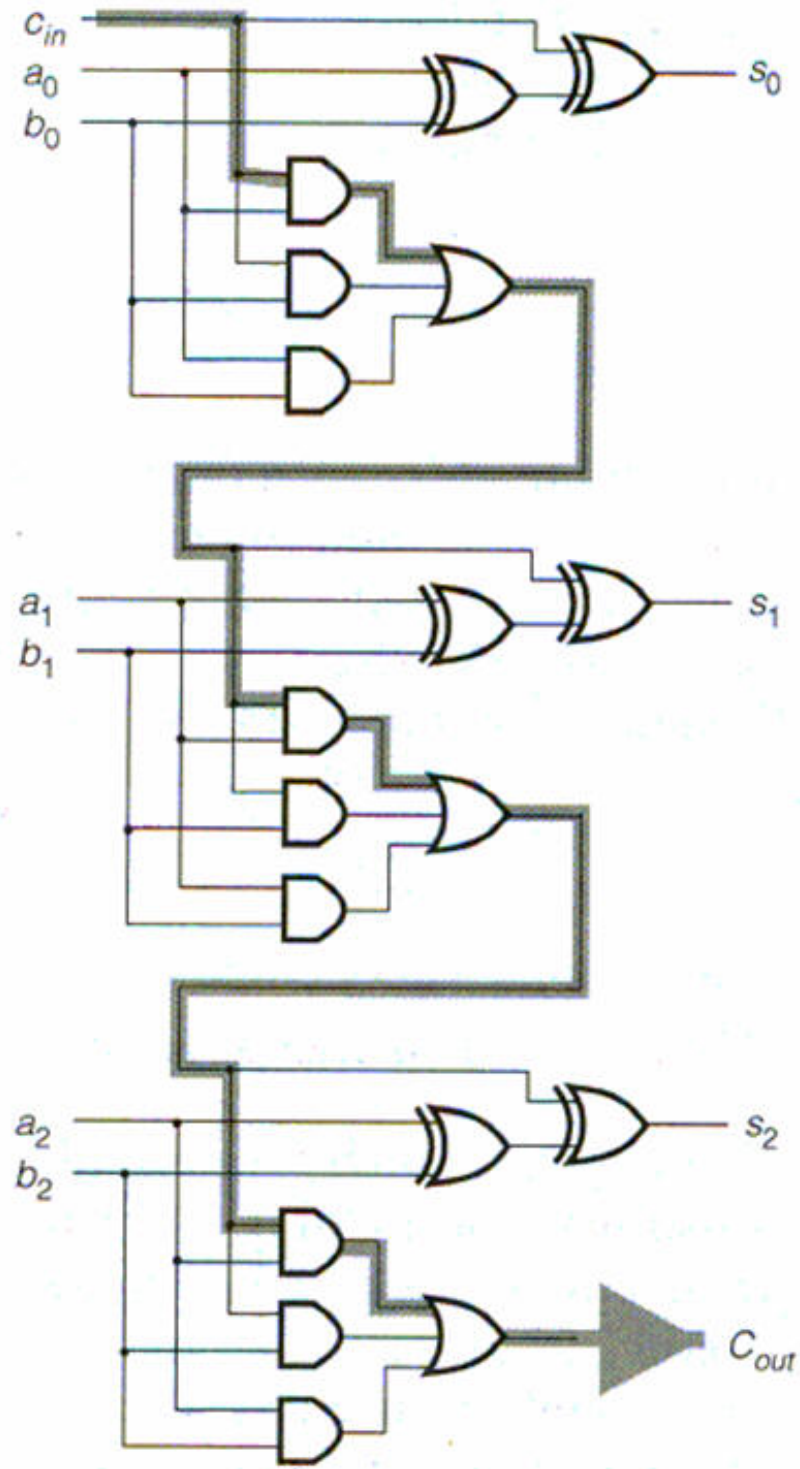
http://dls.makingartstudios.com/tags/adders/index.html





http://dls.makingartstudios.com/tags/adders/index.html

Glitches!



Propagation Delay

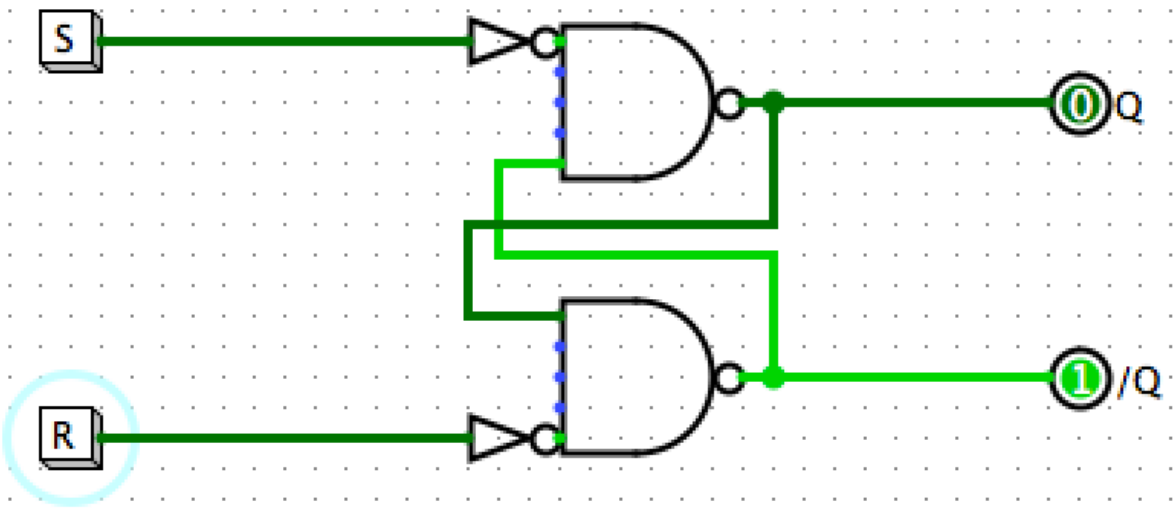
Need to wait for circuit to stabilize

Critical path determines how long

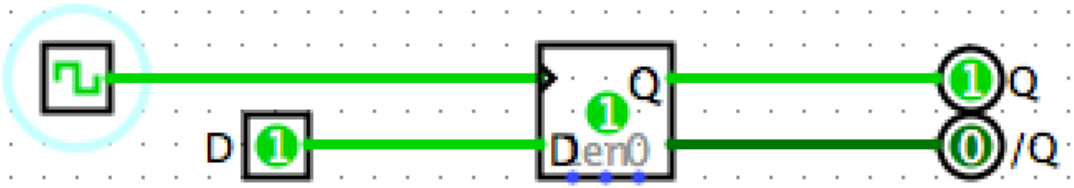
Timing analysis

iceaddn.v timing analysis

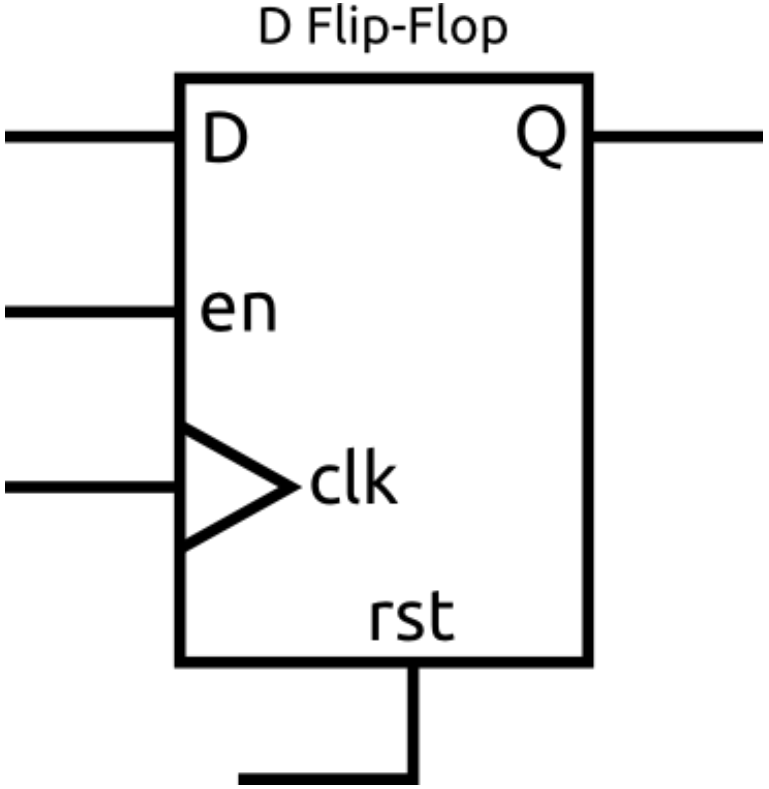
SR Flip Flop



D Flip Flop



D Flip Flop (DFF)



$\overline{SET}$	$\overline{RESET}$	D	CK	Q	$\overline{Q}$
0	1	-	-	1	0
1	0	-	-	0	1
0	0	-	-	1	1
1	1	1	↓	1	0
1	1	0	↓	0	1

- **Has an input D, outputs Q and /Q, and internal state**
- **CLK (rising) and CE (clock enable)**
- **RESET and SET (synchronous/asynchronous)**

Sequential Logic

All flip flops share a global, synchronous clock

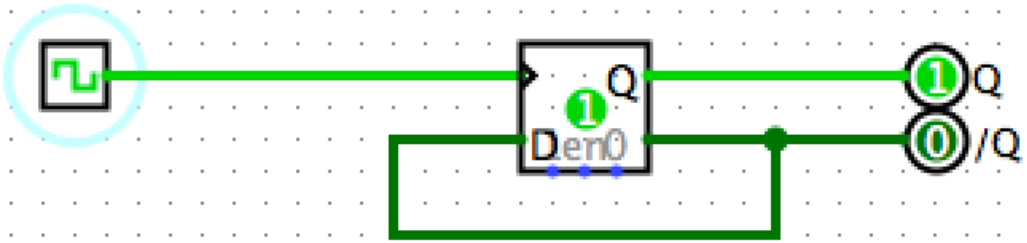
All Flip flops change state simultaneously on the rising edge of the clock

Flip flops can be conditionally updated based on the clock enable

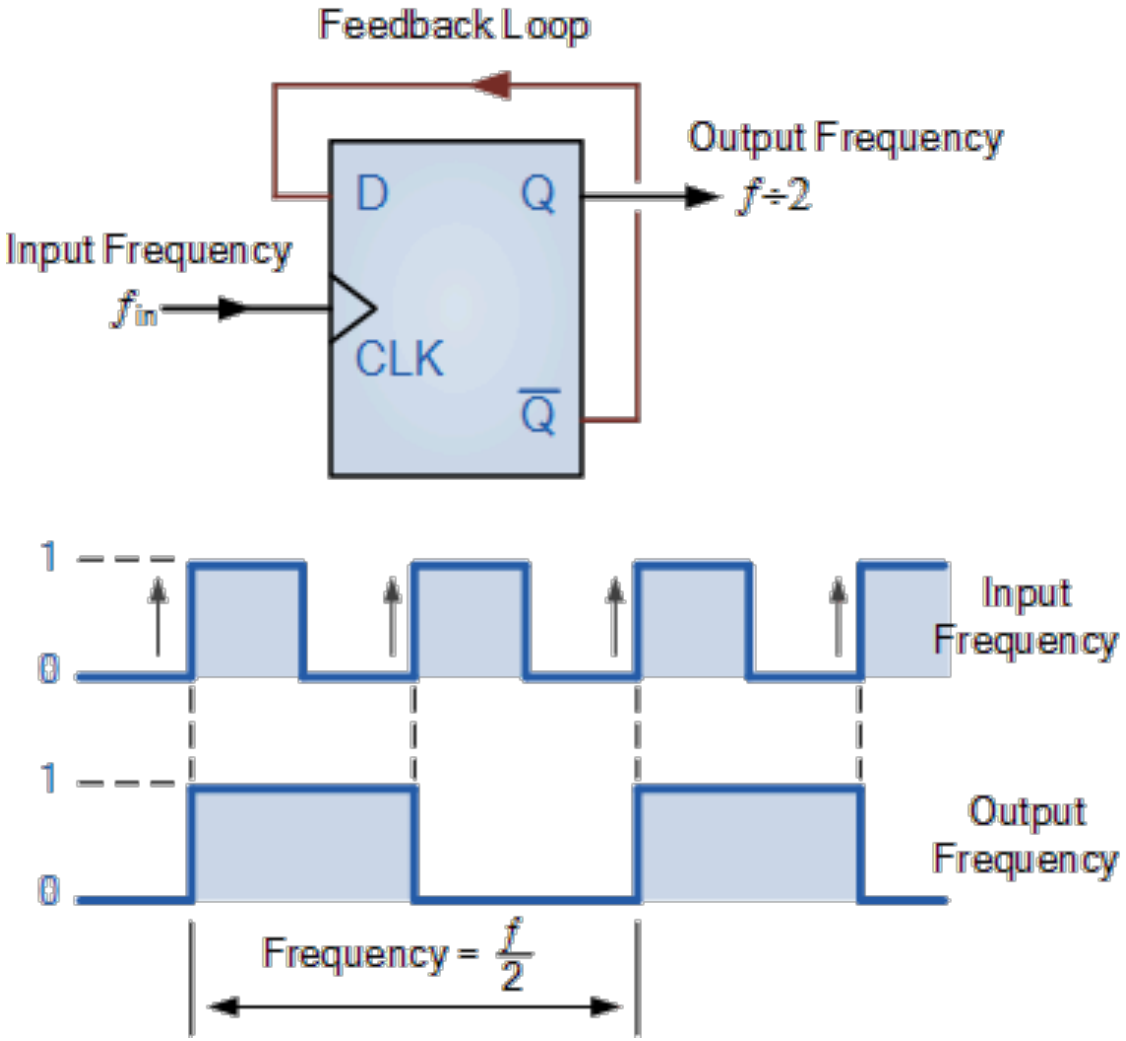
The new value is a combination function of the inputs and the values currently stored in the flip flop

Outputs are combinational functions of the flip flops and the inputs

T (toggle) Flip Flop



TFF


Feedback Loop
Output Frequency
$f \div 2$
D
Q
Input Frequency
f_{in}
CLK
$\overline{Q}$
1
0
Input Frequency
1
0
Output Frequency
Frequency = $\frac{f}{2}$

tff.v

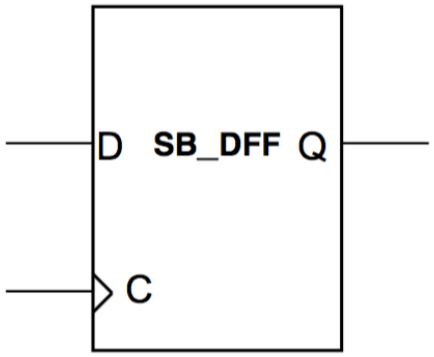
Table of Contents

Register Primitives 6

- SB_DFF 6
- SB_DFFE 8
- SB_DFFSR 10
- SB_DFFR 12
- SB_DFFSS 14
- SB_DFFS 16
- SB_DFFESR 18
- SB_DFFER 20
- SB_DFFESS 22
- SB_DFFES 24
- SB_DFFN 26
- SB_DFFNE 28
- SB_DFFNSR 30
- SB_DFFNR 32
- SB_DFFNSS 34
- SB_DFFNS 36
- SB_DFFNESR 38
- SB_DFFNER 40
- SB_DFFNESS 42
- SB_DFFNES 44

D Flip-Flop

Data: D is loaded into the flip-flop during a rising clock edge transition.



Inputs			Output
	D	C	Q
	0	↗	0
	1	↗	1
Power on State	X	X	0

Key

↗ Rising Edge
1 High logic level
0 Low logic level
X Don't care
? Unknown

HDL use

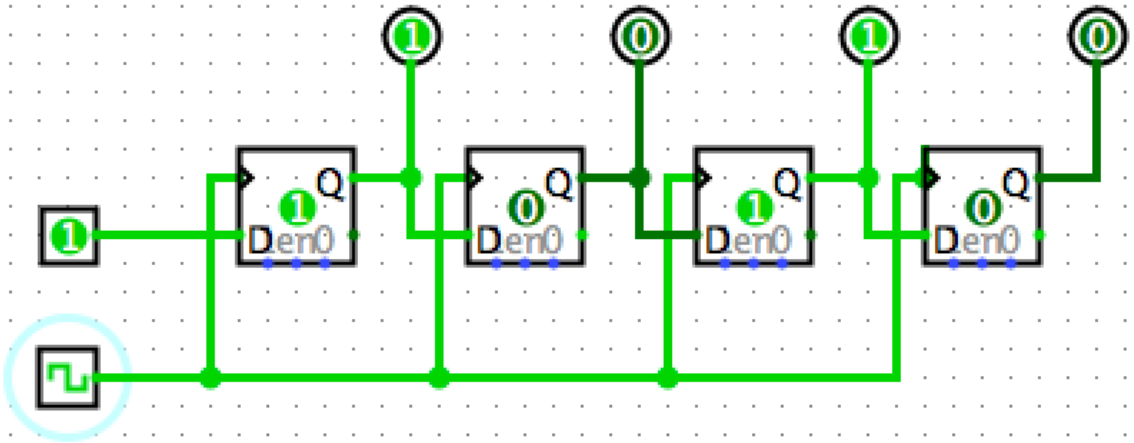
This register is inferred during synthesis and can also be explicitly instantiated.

Verilog Instantiation

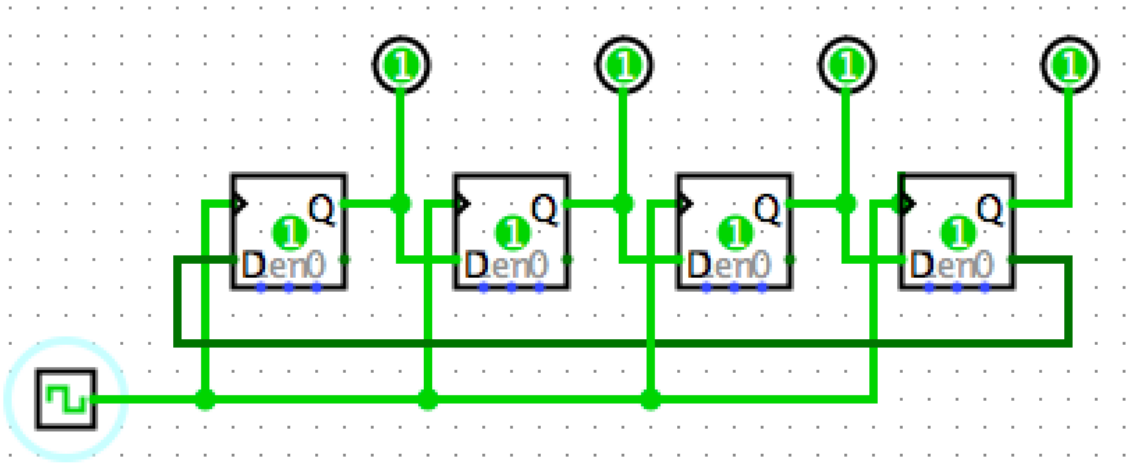
```
// SB_DFF - D Flip-Flop.

SB_DFF SB_DFF_inst (
      .Q(Q),              // Registered Output
      .C(C),              // Clock
      .D(D),              // Data
      );
```

counter.v

Shift Register



Johnson or Twisted Ring Counter



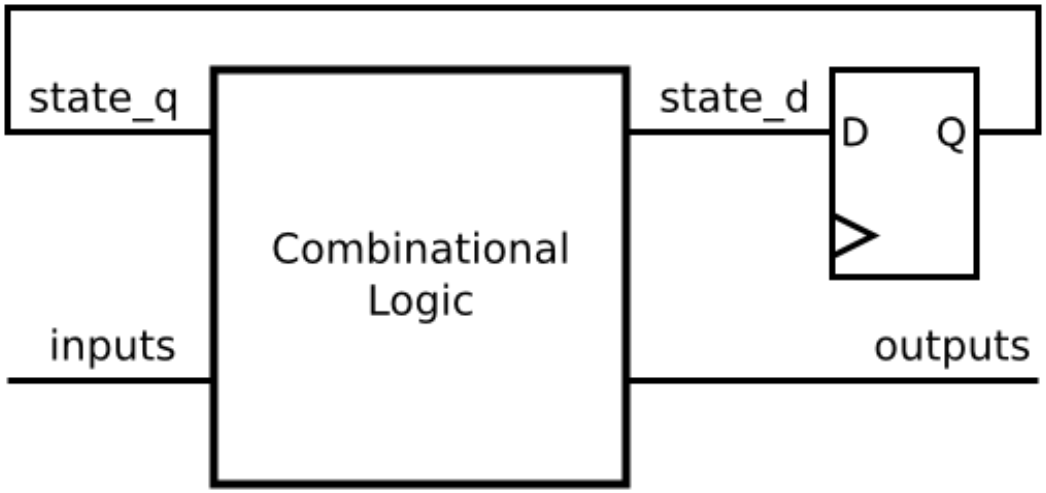
Logisim: Log main of Johnson

Selection | Table | File

Output(340,190)	Output(420,190)	Output(500,190)	Output(580,190)
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1

Close Window

Sequential Logic



Finite State Machine

Assignment 1

Hardware

- Icestick

Software

- `yosys`
- `arachne`
- `icestorm`

Due Fri Jan 27th at 12 midnight