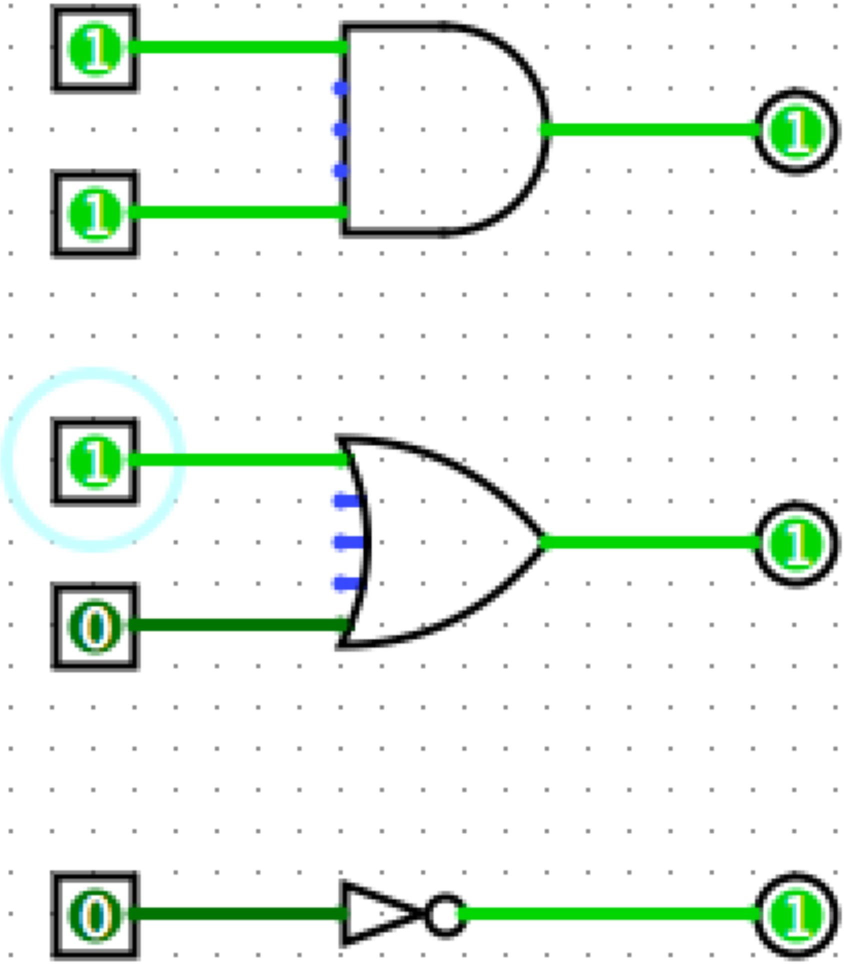
Combinational Logic

Pat Hanrahan

CS448H: Agile Hardware Design
Winter 2017

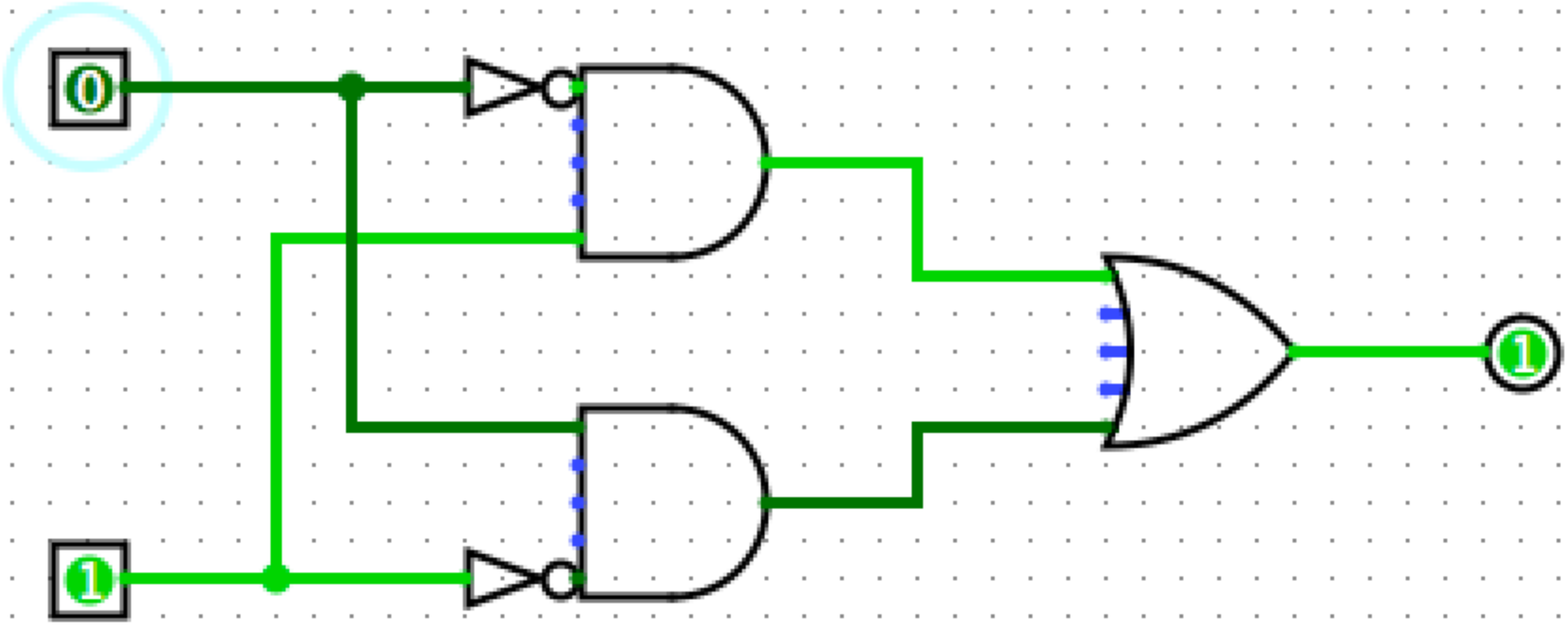
logisim

Logic.circ



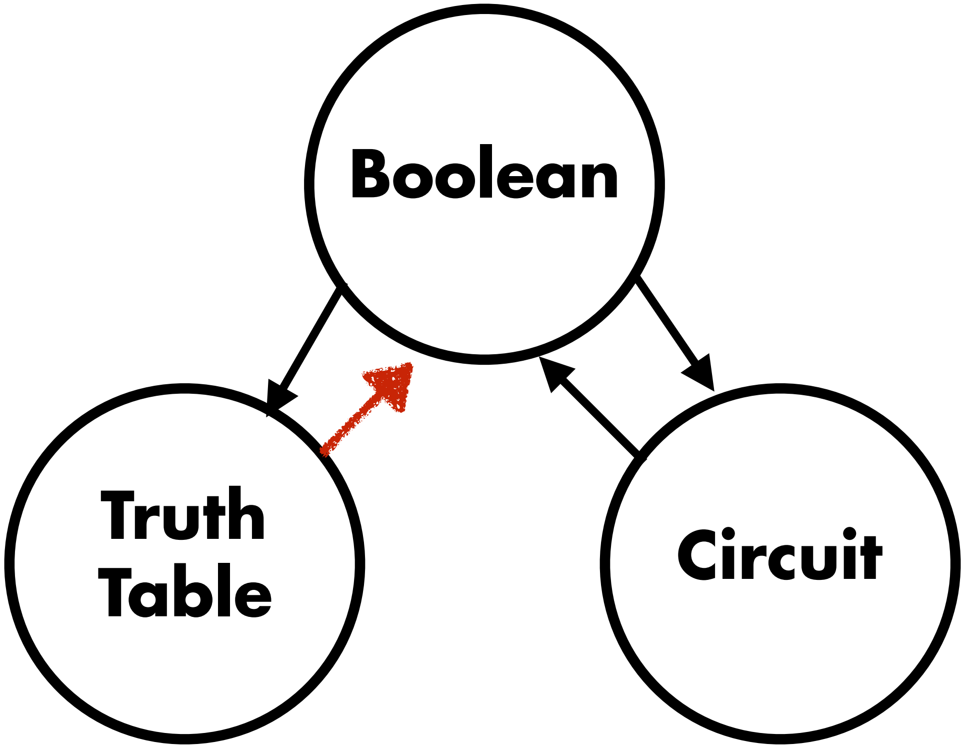
Logisim

Xor.circ



Logisim

Three Representations



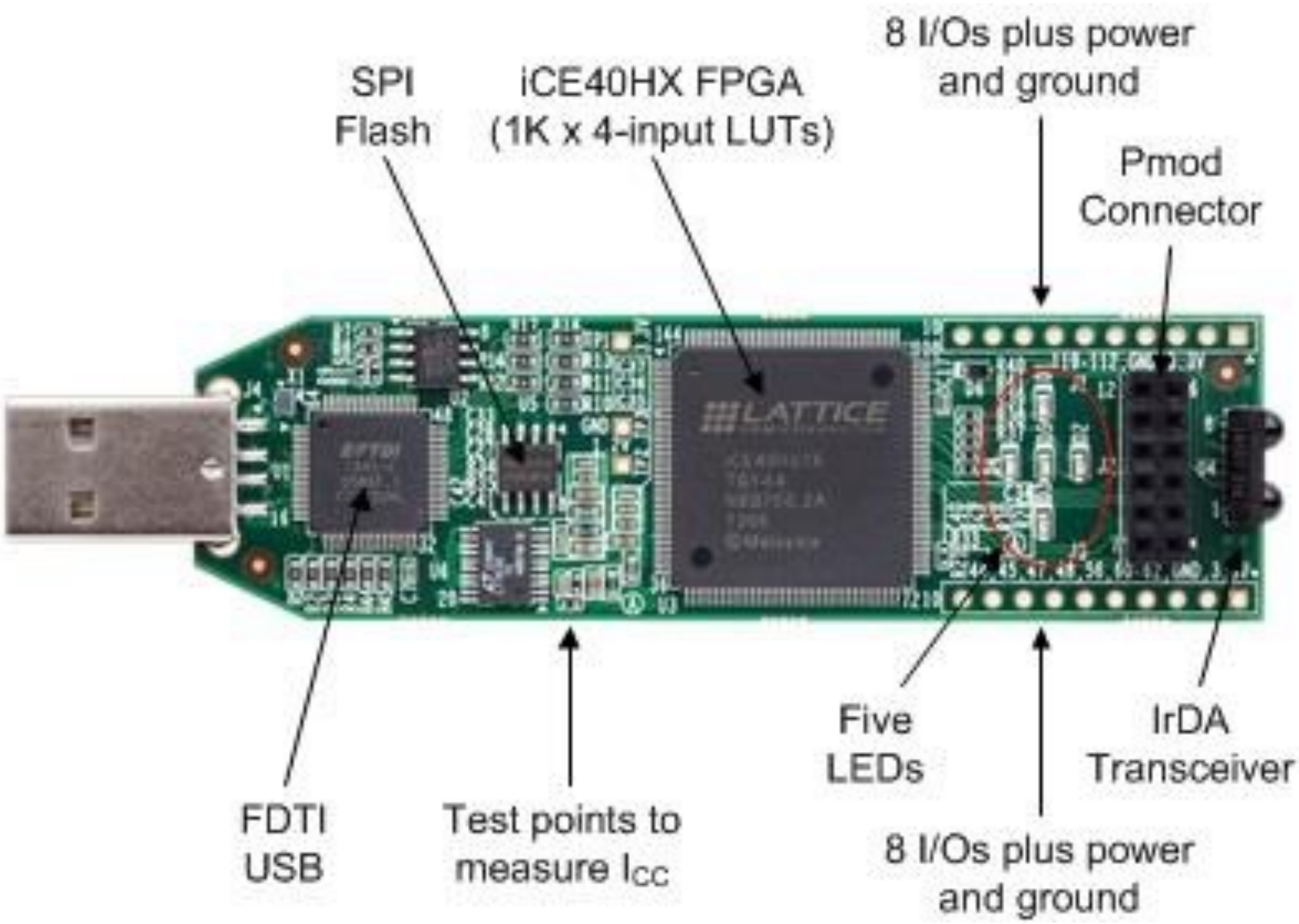
Boolean

Truth Table

Circuit

→ Unique

→ Many formulas (sum of products, logic minimization)

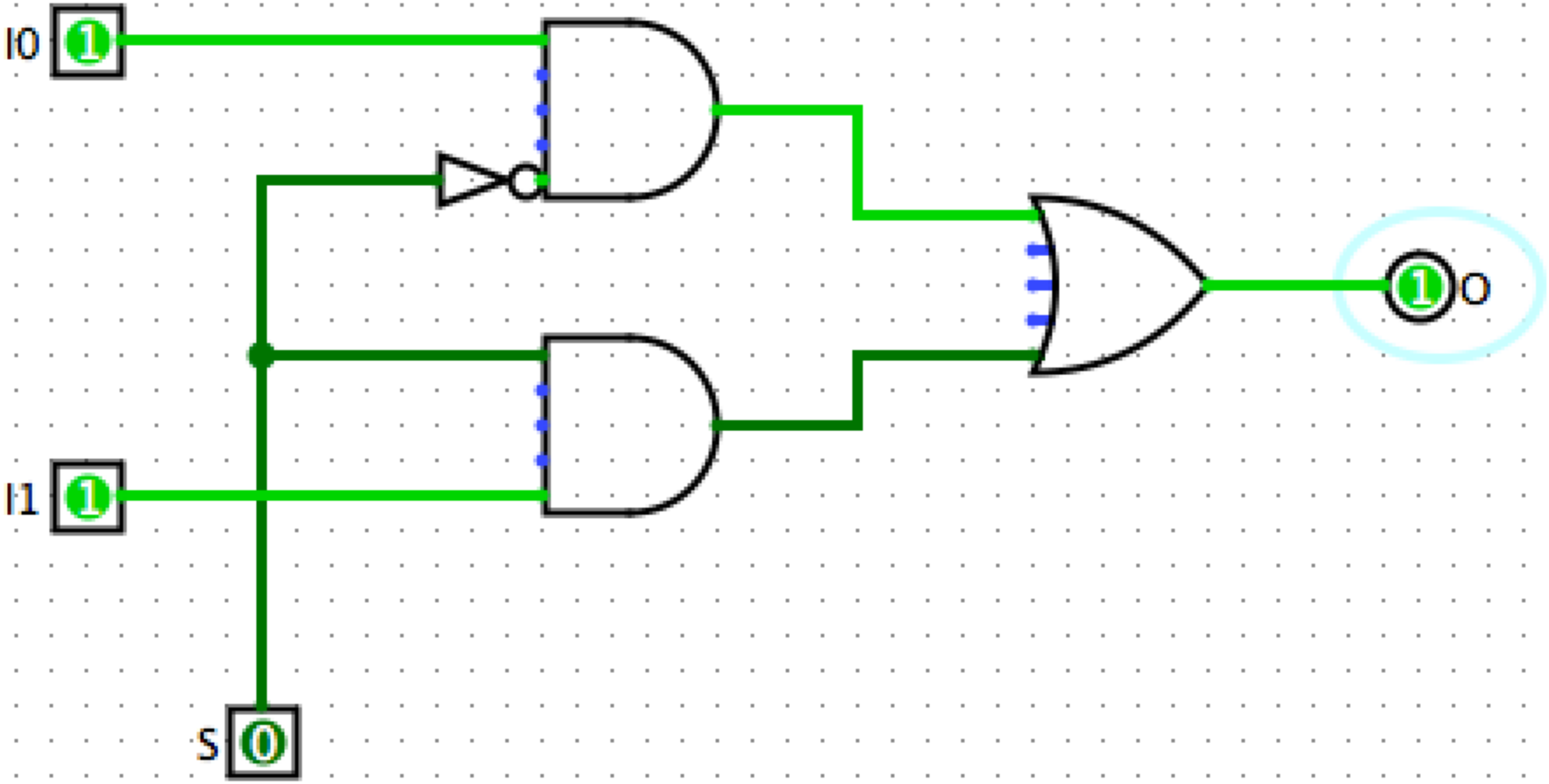

8 I/Os plus power and ground
SPI Flash
iCE40HX FPGA (1K x 4-input LUTs)
Pmod Connector
FDTI USB
Test points to measure I_{CC}
Five LEDs
8 I/Os plus power and ground
IrDA Transceiver

IceStick

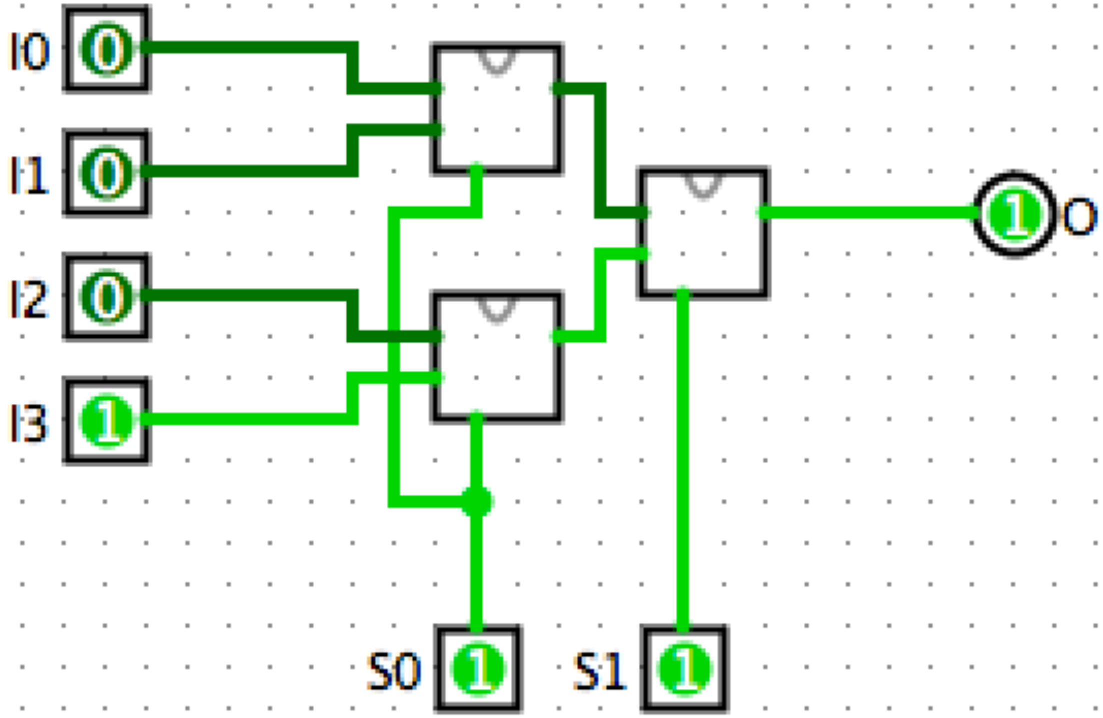
and2.v

xor2.v

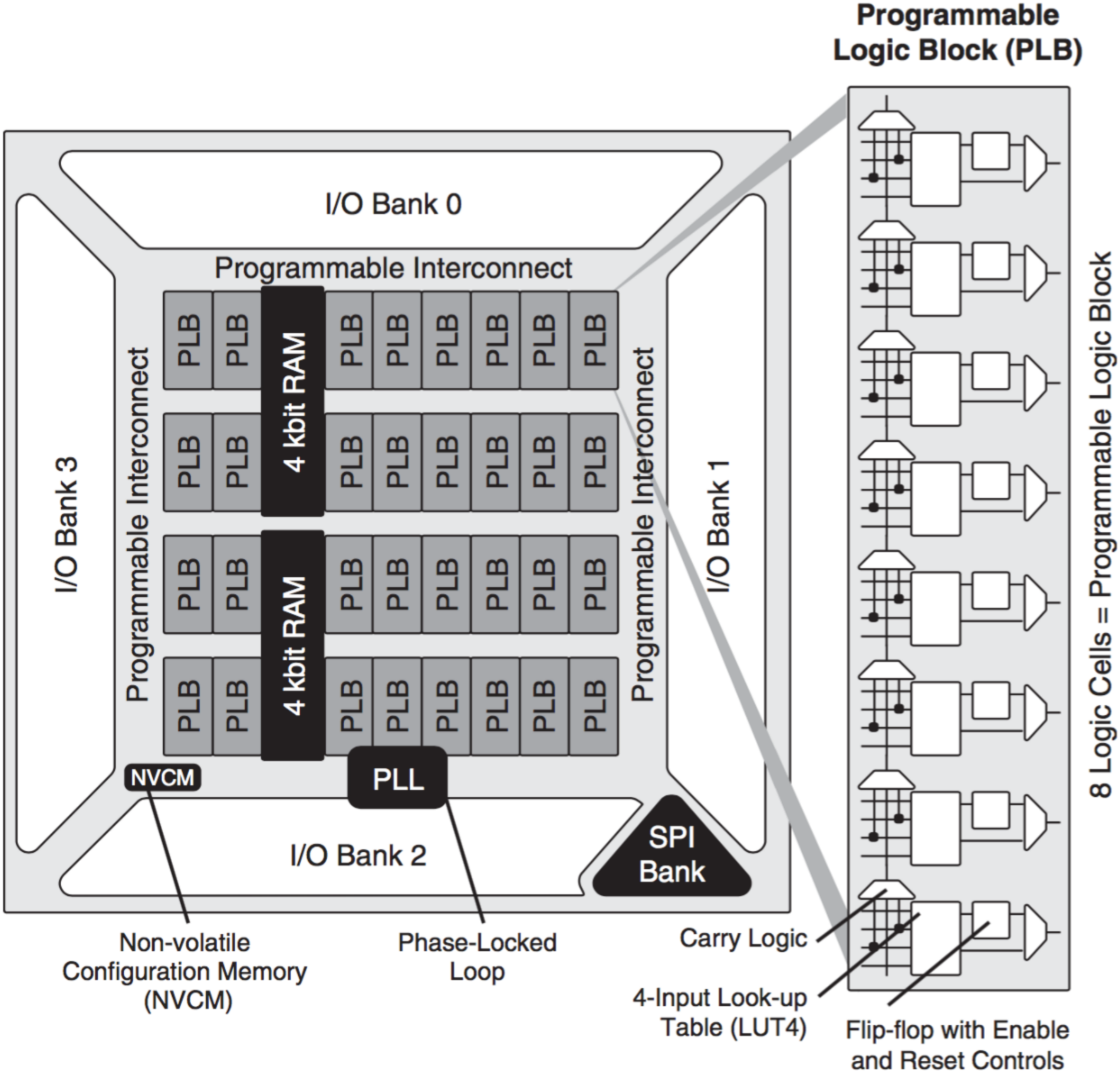
Xor2 -> Mux2



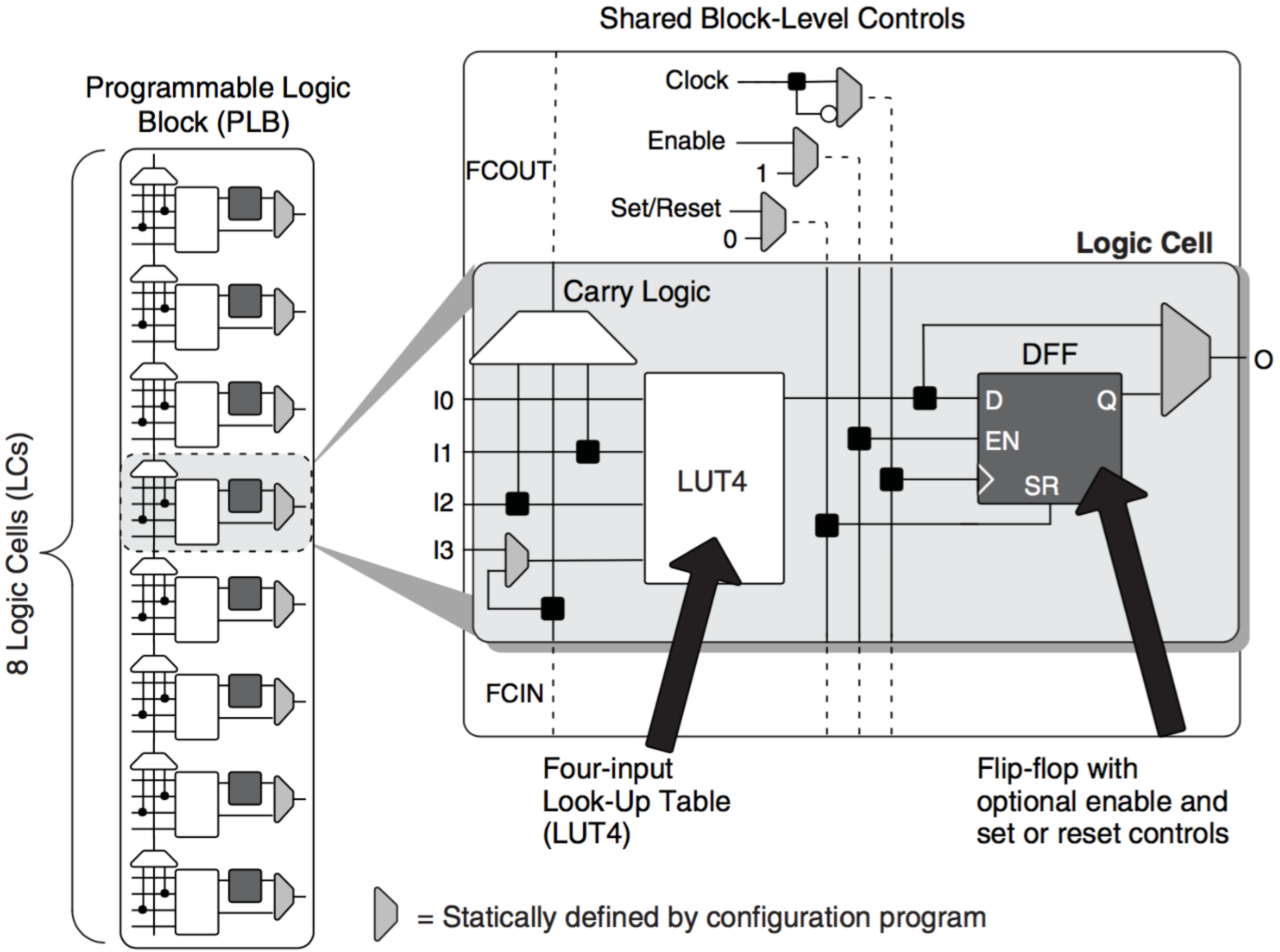
Mux4

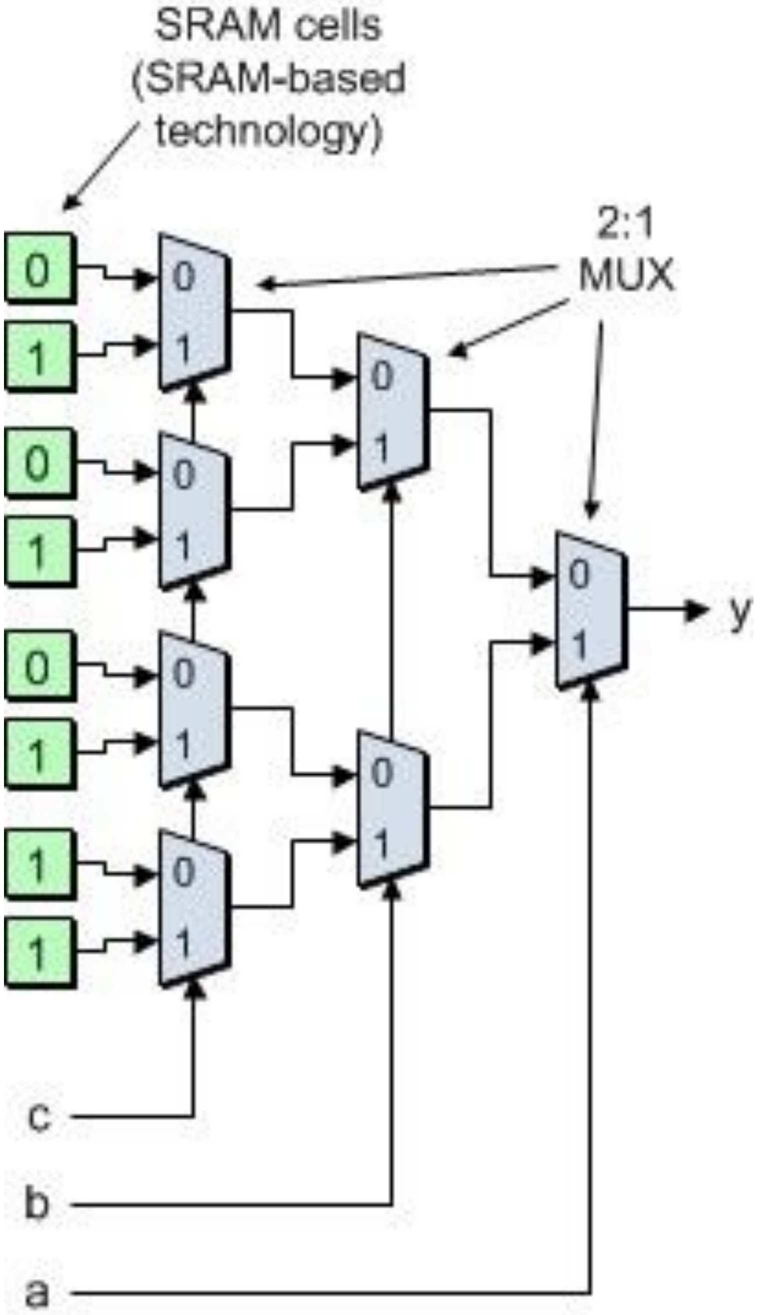


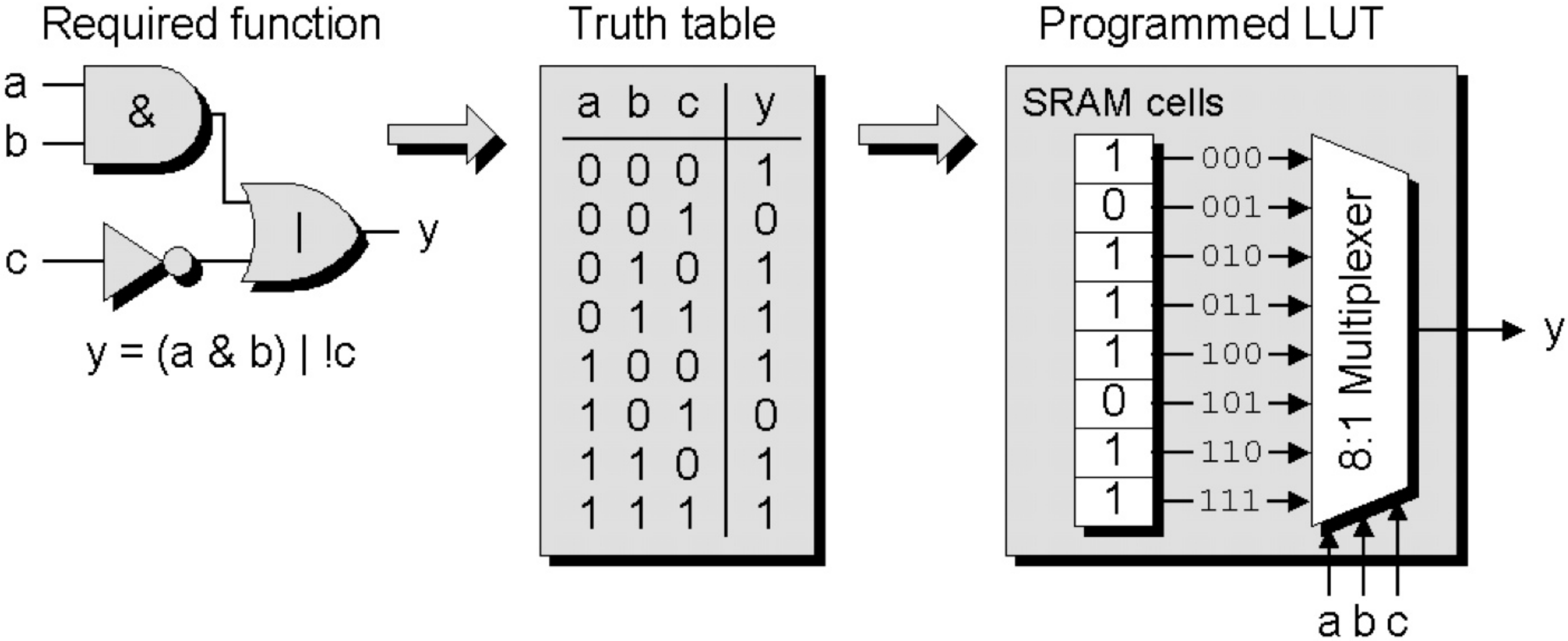
ice40



Programmable Logic Block (PLB)
I/O Bank 0
Programmable Interconnect
PLB
4 kbit RAM
I/O Bank 3
Programmable Interconnect
Programmable Interconnect
I/O Bank 1
NVCM
PLL
I/O Bank 2
SPI Bank
8 Logic Cells = Programmable Logic Block
Non-volatile Configuration Memory (NVCM)
Phase-Locked Loop
Carry Logic
4-Input Look-up Table (LUT4)
Flip-flop with Enable and Reset Controls


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


SRAM cells
(SRAM-based
technology)
2:1
MUX
0
1
0
1
0
1
1
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
y
c
b
a



Required function

Truth table

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Programmed LUT

LUT4 has 16 entries

lut4.v
and2.v, or2.v, xor2.v

```
// Represent functions as bit patterns
//  see lut.py
ZERO = 0b0000000000000000
ONE  = 0b1111111111111111
I0   = 0b1010101010101010
I1   = 0b1100110011001100
I2   = 0b1111000011110000
I3   = 0b1111111100000000

LUT4(ZERO)(i0,i1,i2,i3)==0
LUT4(ONE)(i0,i1,i2,i3)==1
LUT4(I0)(i0,i1,i2,i3)==i0
LUT4(I0^I1^I2^I3)(i0,i1,i2,i3)==i0^i1^i2^i3
LUT4((~I2&I0)|(I2&I1)(i0,i1,i2,i3)==i2?i1:i0
```

Combinational Functions

Logic

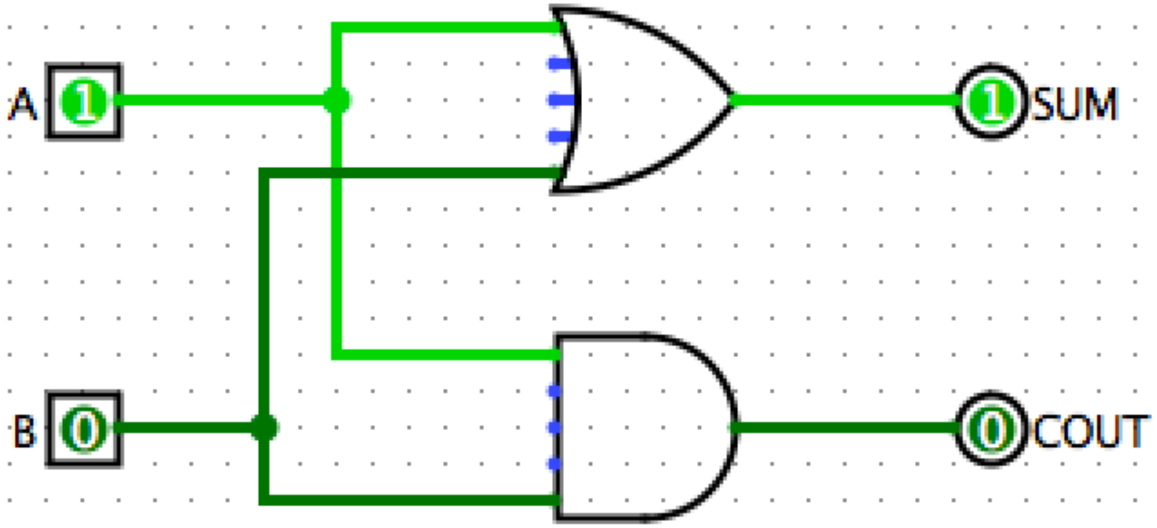
Arithmetic

Comparisons

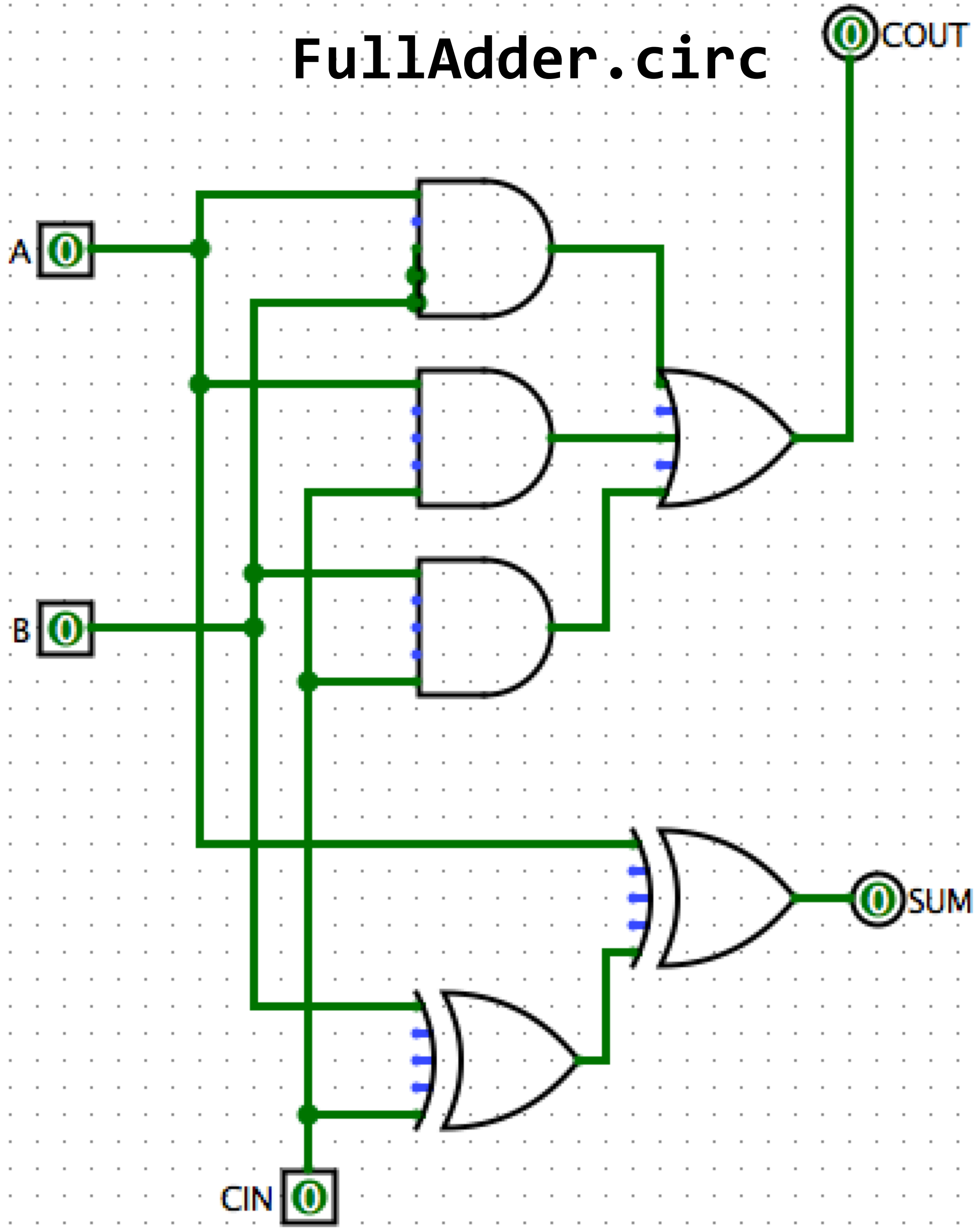
Multiplexers

Decoders and Encoders

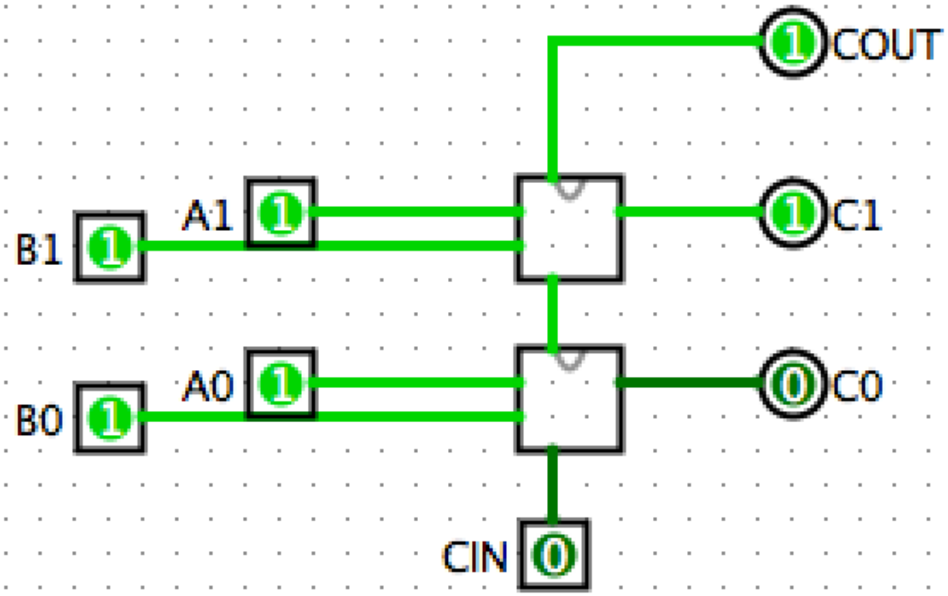
HalfAdder.circ



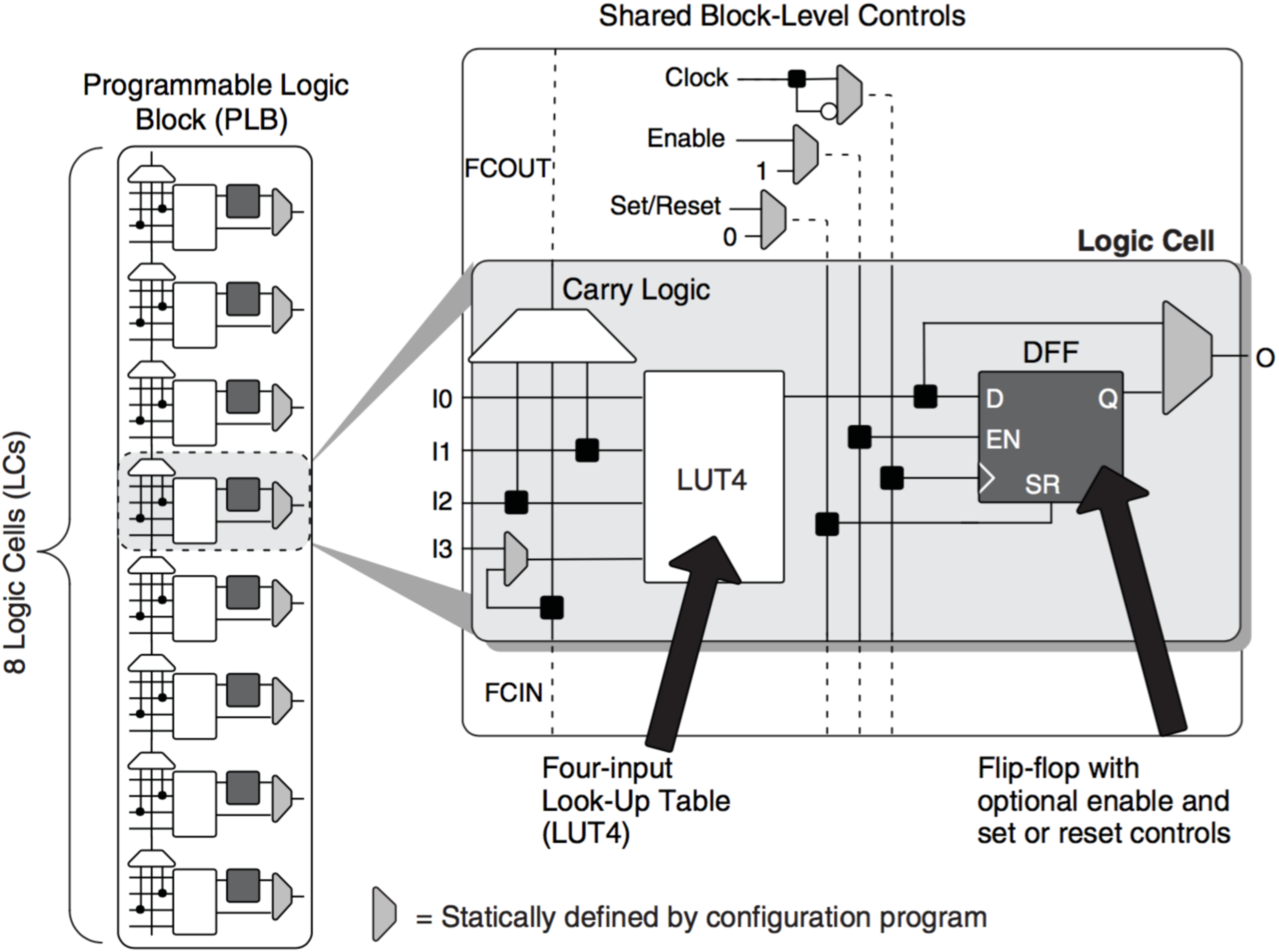
FullAdder.circ



Add2.circ



fa.v, add4.v


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

iceadd4.v

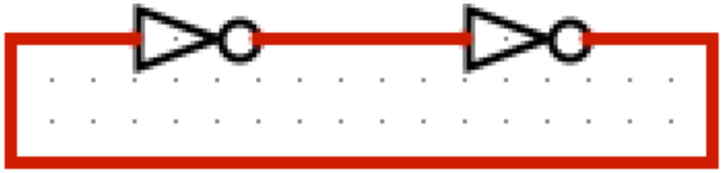
Combinational Logic

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

1. All gates should have valid inputs
2. No cycles

Pure and *total* function of inputs

Executes continuously and in parallel



SR Flip Flop