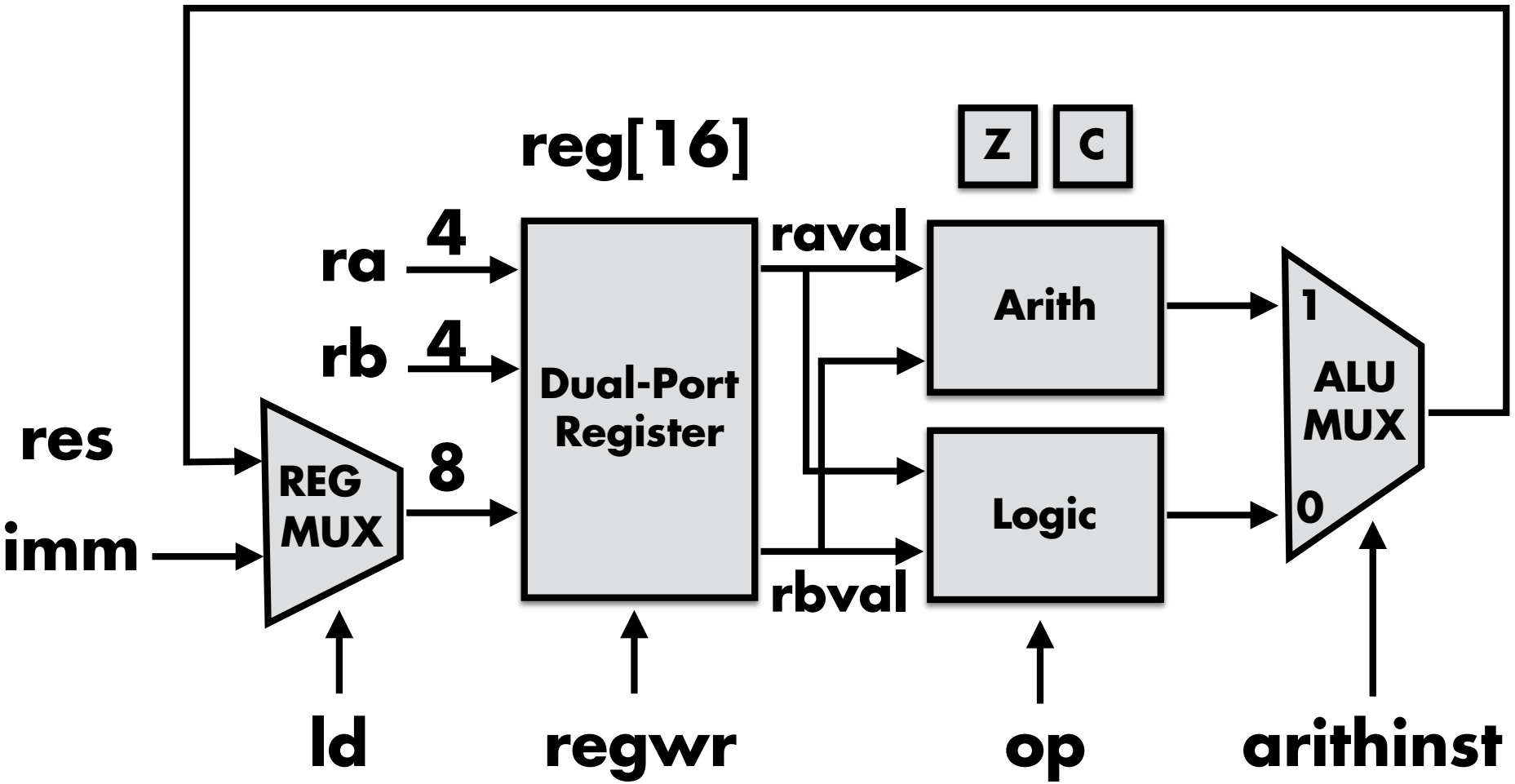
Tool Chain

Pat Hanrahan

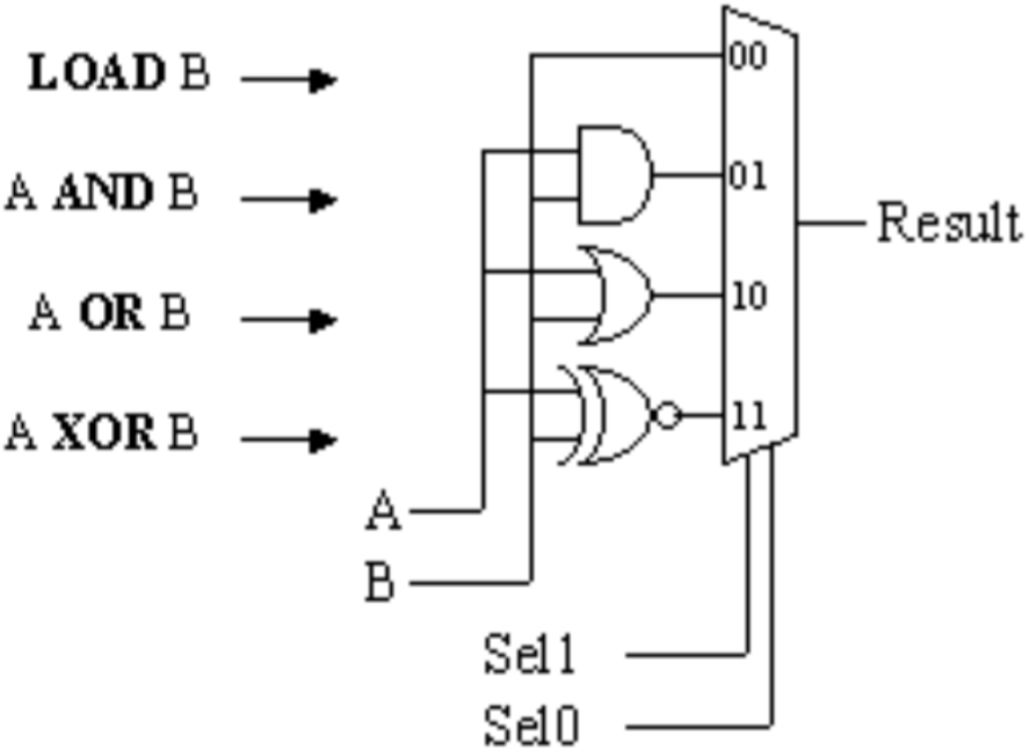
CS448H: Agile Hardware Design
Winter 2017

Pico40

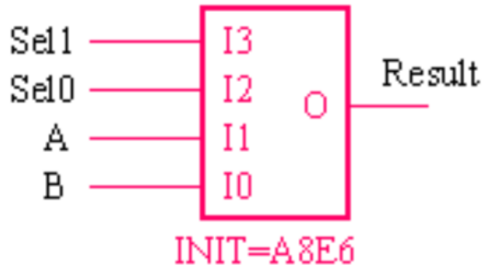

reg[16]
Z
C
ra
4
rb
4
raval
Arith
1
ALU
MUX
Dual-Port
Register
res
REG
MUX
8
Logic
0
imm
rbval
ld
regwr
op
arithinst

Logic Unit

Logic Unit

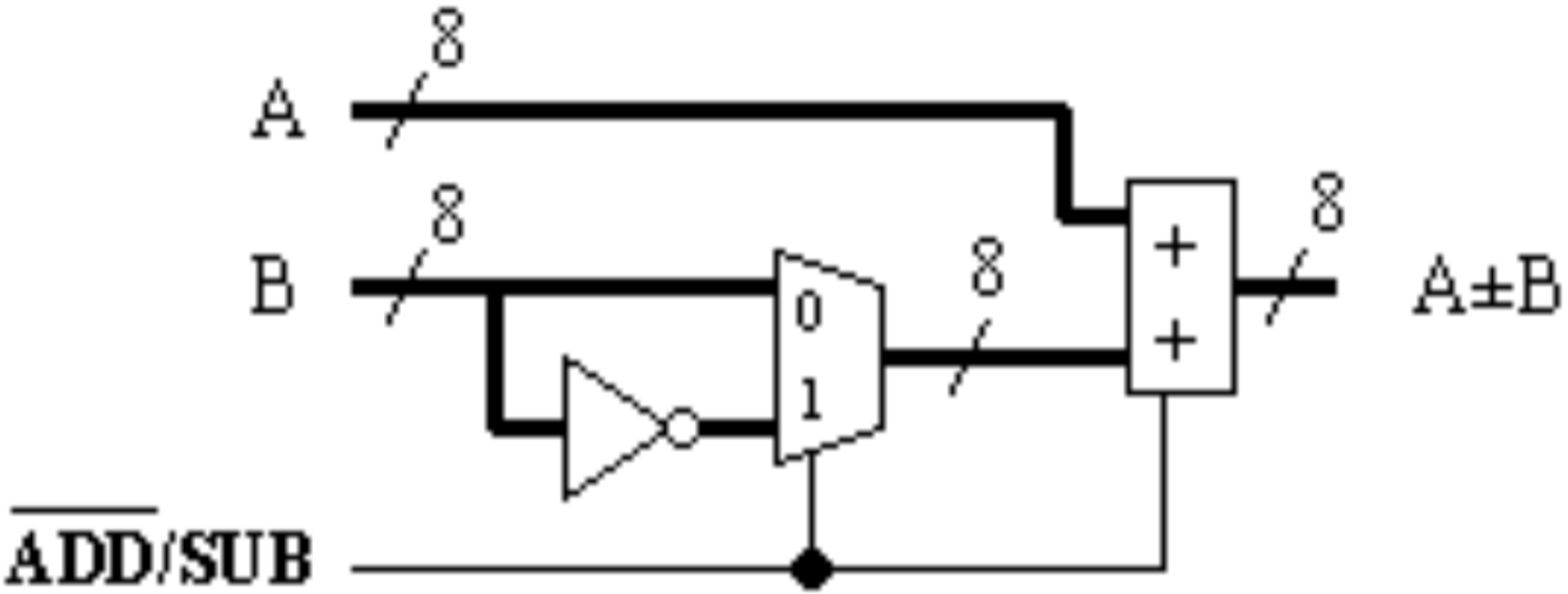


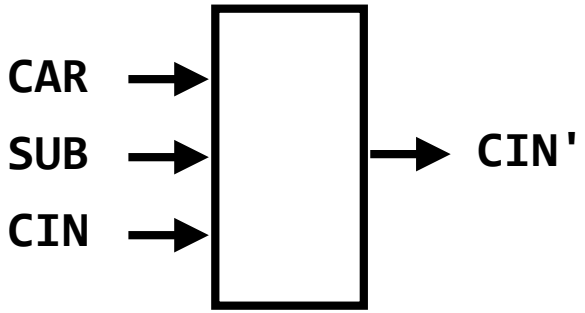
Logical Group					
I3	I2	I1	I0	O	INIT=A8E6
Sel1	Sel0	A	B	Result	
0	0	0	0	0	6
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	0	
0	1	0	0	0	E
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	8
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	0	A
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	



```
def logicfunc(A, B, S0, S1):
    if S1 == 0:
        if S0 == 0:
            return B
        else:
            return A&B
    else:
        if S0 == 0:
            return A|B
        else:
            return A^B

def Logic(n):
    def logic(y):
        return LUT4(logicfunc)
    return braid(col(logic, n),
                 joinargs=['I0', 'I1'],
                 forkargs=['I2', 'I3'])
```

Arith Unit





#		CAR	SUB	CIN	CIN'
#	ADD	0	0	0	0
#	ADD	0	0	1	0
#	SUB	0	1	0	1
#	SUB	0	1	1	1
#	ADDC	1	0	0	0
#	ADDC	1	0	1	1
#	SUBC	1	1	0	1
#	SUBC	1	1	1	0

```
def Arith(n):
    A = In(Array(n, Bit))()
    B = In(Array(n, Bit))()
    SUB = In(Bit)()
    CARRY = In(Bit)()
    CIN = In(Bit)()

    inv = Invert(n)
    wire( B, inv )

    mux = Mux(2,n)
    wire(   B, mux.I0 )
    wire( inv, mux.I1 )
    wire( SUB, mux.S )

    add = AddC(n, cin=True)
    wire(  A, add.I0 )
    wire(mux, add.I1)

    truth = [0, 0, 1, 1, 0, 1, 1, 0]
    cin = LUT3(truth)(CIN, SUB, CARRY)
    wire(cin, add.CIN)
    return AnonymousCircuit("A",A,"B",B,"SUB",SUB,"CARRY",CARRY,"CIN",CIN,
                            "O",add.O, "COUT",add.COUT)
```

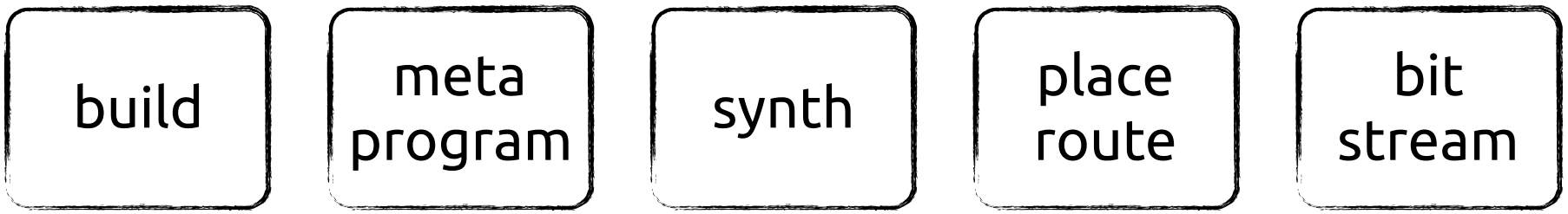
Numbers of LUTs?

```
IOs             17 / 96
GBs             0 / 8
  GB_IOs        0 / 8
LCs             533 / 1280
  DFF           146
  CARRY         15
  CARRY, DFF    0
  DFF PASS      136
  CARRY PASS    2
BRAMs           1 / 16
WARMBOOTs       0 / 1
PLLs            0 / 1
```

```
Resolvable net names on path:
     2.246 ns ..  4.399 ns inst0_RDATA[4]
     4.778 ns ..  5.907 ns inst58.inst0_O
     6.356 ns ..  6.945 ns inst65.inst0_O
     7.345 ns ..  8.825 ns inst68.inst0_O
     9.274 ns .. 10.234 ns inst70.inst0_O
    10.683 ns .. 11.848 ns inst79.inst0_O
    12.247 ns .. 12.836 ns inst80.inst0_O
    13.068 ns .. 13.068 ns inst81.inst1_CO
    13.194 ns .. 13.454 ns inst81.inst3_CO
    13.769 ns .. 15.347 ns inst81.inst4_O
    15.747 ns .. 17.641 ns inst83.inst2_O
    18.019 ns .. 18.980 ns inst96_O
    19.380 ns .. 20.544 ns inst97_O
    20.944 ns .. 21.533 ns inst98_O
    21.933 ns .. 23.511 ns inst99_O
                  lcout -> inst10_Q

Total number of logic levels: 15
Total path delay: 23.91 ns (41.82 MHz)
```

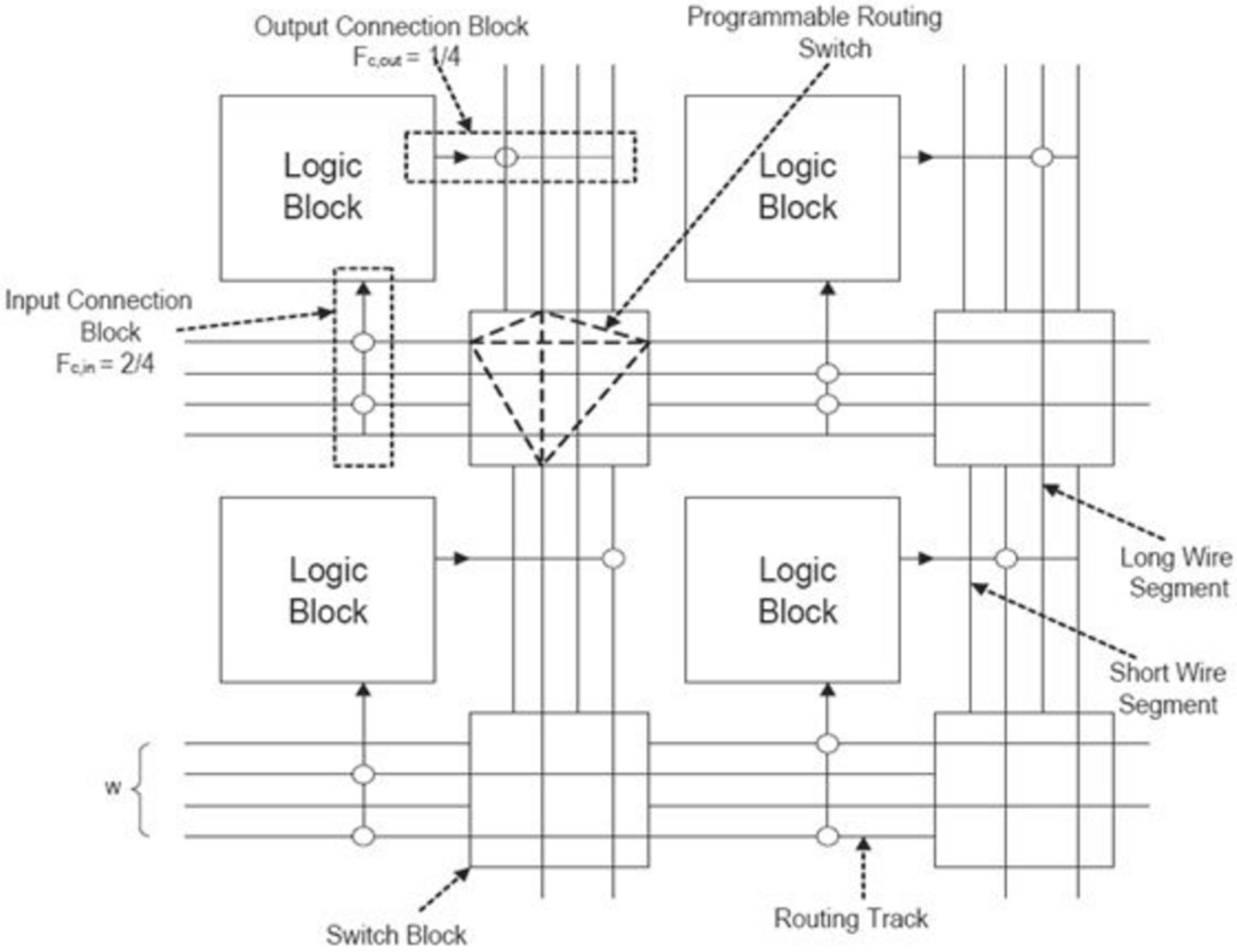
Tools / Flow

Tool Chain



bake **magma** **yosis** **arachne** **icepack**

ice storm



"Island" Architecture

ice40 architecture

http://www.clifford.at/icestorm/

yosys

http://www.clifford.at/yosys/

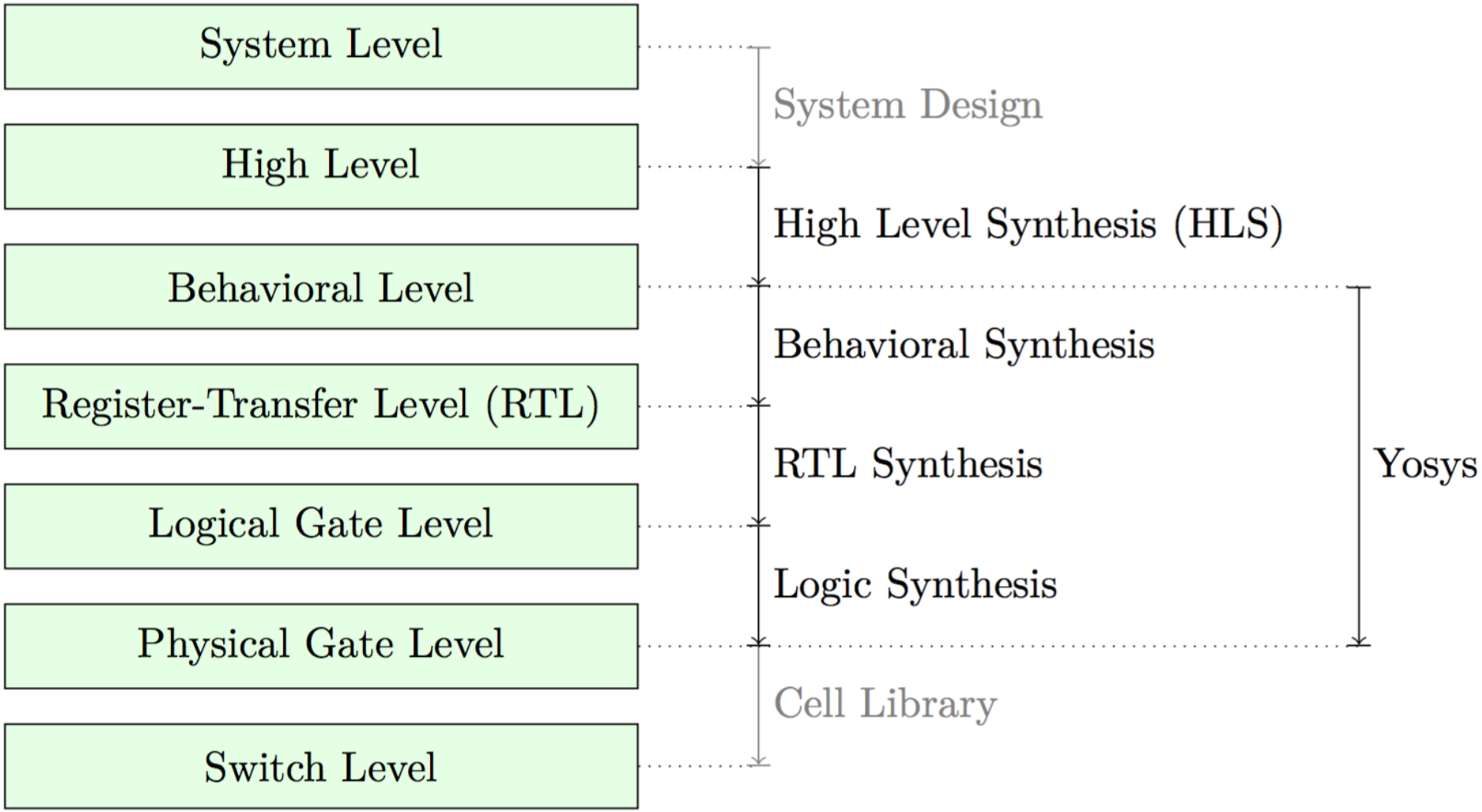


Figure 2.1: Different levels of abstraction and synthesis.

From yosys manual

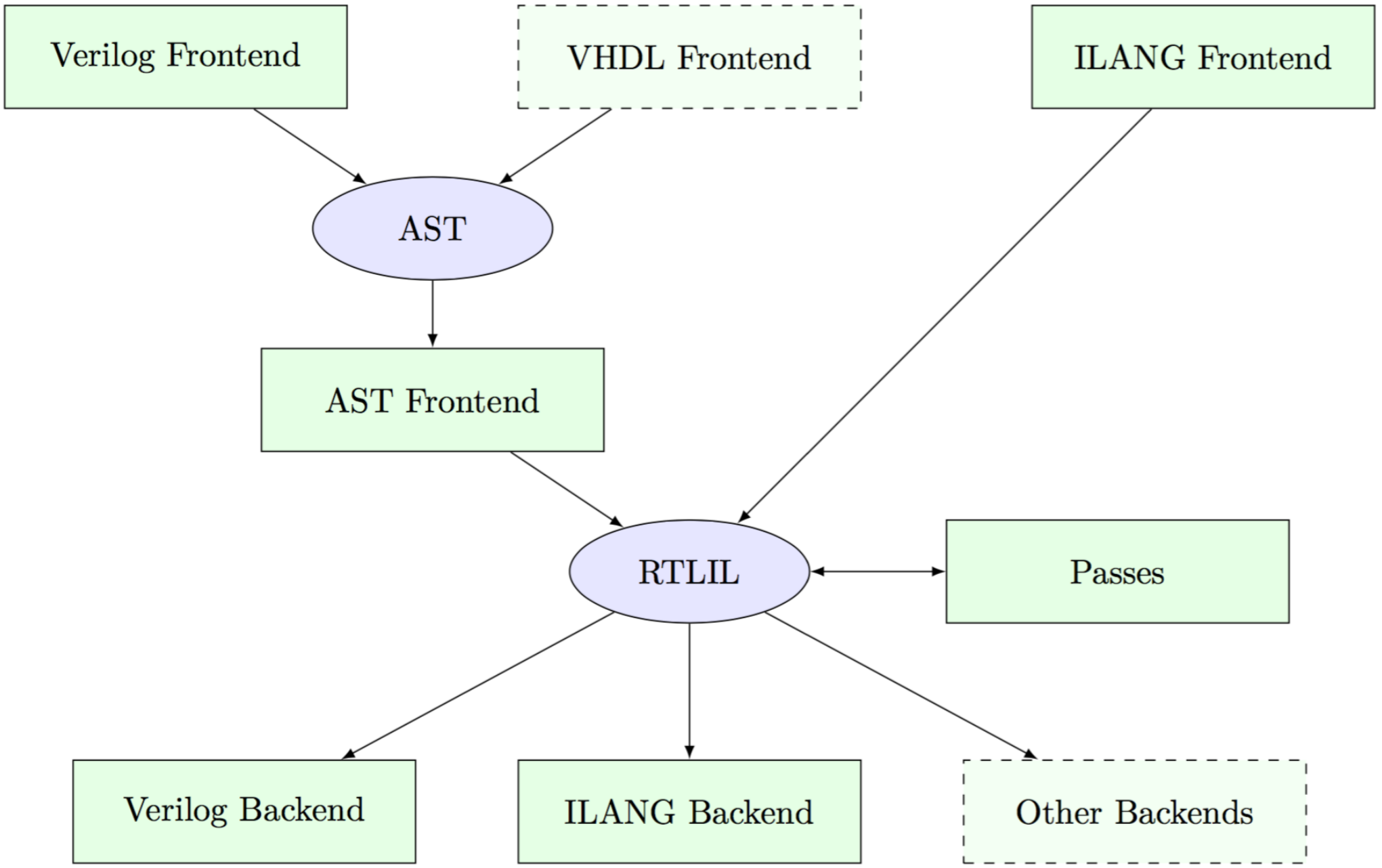
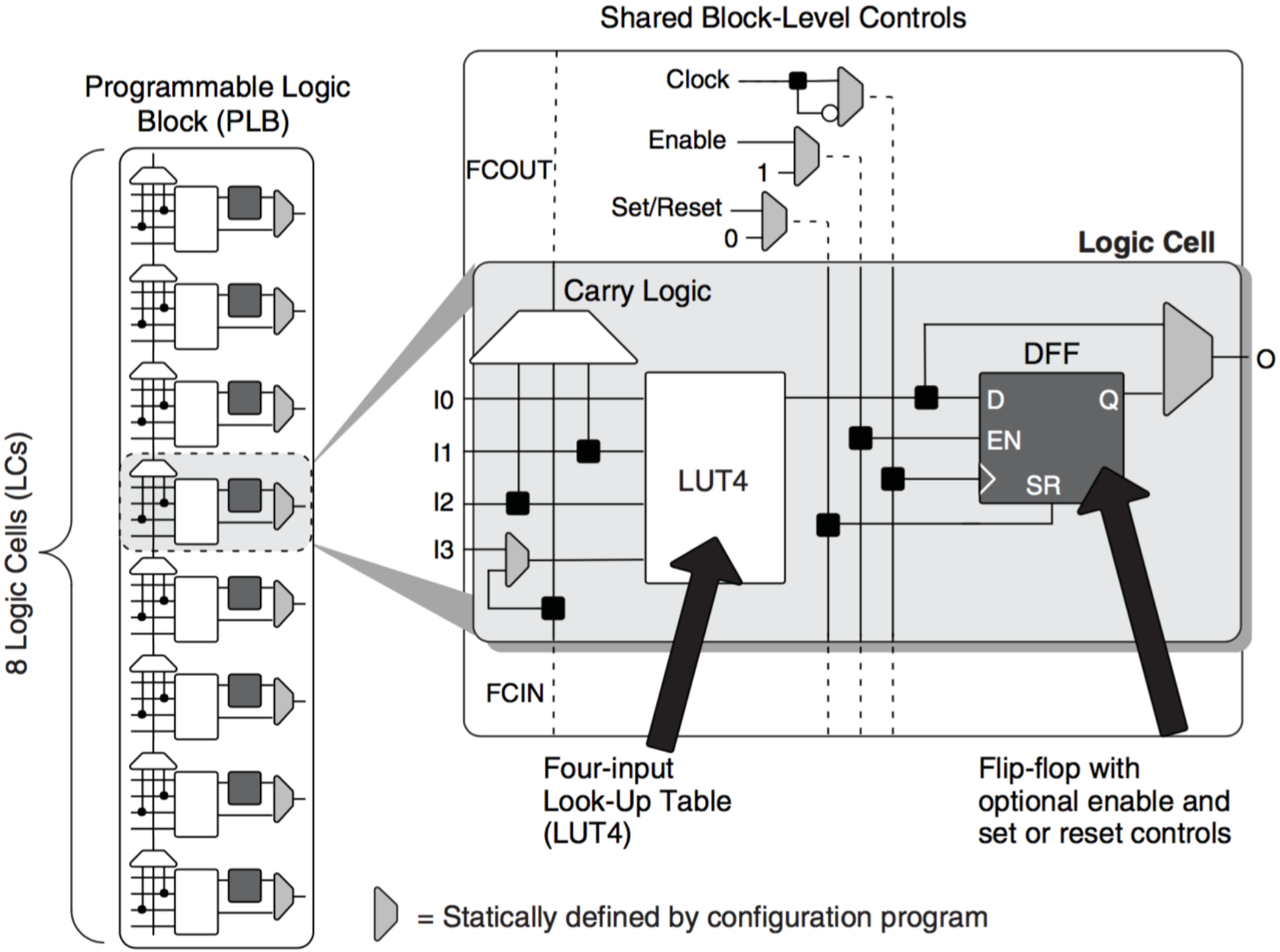


Figure 4.1: Yosys simplified data flow (ellipses: data structures, rectangles: program modules)

From yosys manual

```
begin:
    read_verilog -lib +/ice40/cells_sim.v
    hierarchy -check -top <top>

flatten:    (unless -noflatten)
    proc
    flatten
    tribuf -logic
    deminout

coarse:
    synth -run coarse

fine:
    opt -fast -mux_undef -undriven -fine
    memory_map
    opt -undriven -fine
    techmap -map +/techmap.v -map +/ice40/arith_map.v
    abc -dff    (only if -retime)
    ice40_opt

map_ffs:
    dffsr2dff
    dff2dffe -direct-match $_DFF_*
    techmap -map +/ice40/cells_map.v
    opt_expr -mux_undef
    simplemap
    ice40_ffinit
    ice40_ffssr
    ice40_opt -full

map_luts:
```

top.v
top.blif
top.vv


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

```
// See yosys/techlibs/ice40/cells_sim.v

// Packed IceStorm Logic Cells

module ICESTORM_LC (
    input I0, I1, I2, I3, CIN, CLK, CEN, SR,
    output LO, O, COUT
);
    parameter [15:0] LUT_INIT = 0;

    parameter [0:0] NEG_CLK       = 0;
    parameter [0:0] CARRY_ENABLE = 0;
    parameter [0:0] DFF_ENABLE    = 0;
    parameter [0:0] SET_NORESET  = 0;
    parameter [0:0] ASYNC_SR      = 0;
```

arachne-pnr

Pack

`top.pack.vv`

`pack.cc`

Place

top.place.vv

place.cc

icedrom floorplan