Cologne Processing Element

File: cpe.kicad_sch

C_FUNCTION

C_FUNCTION=
0: (none)
1: "C_ADDF": C_ADDF, C_ADDCIN
2: "C_ADDF2": C_ADDF, C_ADDF2, C_ADDCIN
3: "C_MULT": C_MULT, C_ADDF, C_ADDF2, C_ADDCIN
4: "C_MX4": C_MX4
5: "C_EN_CIN": C_EN_CIN
6: "C_CONCAT": C_ADDF
7: "C_ADDCIN": C_ADDCIN

C_MULT
C_MX4   C_MX4
C_EN_CIN   C_EN_CIN
C_ADDF
C_ADDF2   C_ADDF2
C_ADDCIN   C_ADDCIN

COMBINATIONAL LOGIC

MUX4a
(x4)
M1 M2 M3 M4
C = L10
X = L11
C_MX4

IN1 IN2 IN3 IN4

C_I1 L00
C_I2 L01
C_I3 L02
C_I4 L03

C_EN_CIN

C_MX4
MUX4b
(x4)
M1 M2 M3 M4
C = L11
X = L10

C_HORIZ

B_1
A_0
B_1
A_−1

In multiplier mode,
the LUTs should be set up as:
L10 = (A[1] & B[0]) ^ SI[1]
L11 = (A[0] & B[0]) ^ SI[0]

C_ADDF2
C_MX4
C_ADDF2
C_ADDCIN

CADD_S
PROPAGATE
COMB2OUT
COMB1OUT
COMP_OUT
CADD_A
GENERATE
COY2_a
COY2_S
MULTO2
MULTO1

C_SN[2]
C_SN[1]
C_SN[0]

SR C_R
END C_EN C_ENSEL
PINY1 C_CLK C_CLKSEL
CLK
CINY1

C_BR
C_BR
C_BR
C_2D_IN

C_FUNCTION=0 or 6
File: cpe_comb.kicad_sch
C_FUNCTION=1 or 7
File: cpe_comb_addcin.kicad_sch
C_FUNCTION=2 or 3
File: cpe_comb_addf2.kicad_sch
C_FUNCTION=4
File: cpe_comb_mux4.kicad_sch
C_FUNCTION=5
File: cpe_comb_en_cin.kicad_sch

MUXOUT

SEQUENTIAL LOGIC

POUTX
POUTY1
COUTX
COUTY1

C_C_P
C_C_P

CLK C_CLK C_CLKSEL
C_HORIZ
C_MULT
CINY1 C_EN C_ENSEL
END
PINY1

Because of the inverted D input,
the SET/RST signals go to the
R/S ports of the flops and
latches respectively.

SR C_R
END C_EN_SR C_S
SRD

D LATCH
D FLIP FLOP
D LATCH
D FLIP FLOP

C_L_D
C_L_D

C_BR
C_2D_IN

COMB1OUT
COMB2OUT
MUXOUT

SET RST EN CLK

C_O1[1]
C_O1[0]
MUXOUT
COMB1OUTD
RAM_I1 C_RAMI1 OUT1
C_RAM_O1 RAM_O1

C_O2[1]
C_O2[0]
MUXOUT
COMB2OUTD
RAM_I2D C_RAMI2 OUT2
C_RAM_O2 RAM_O2

CARRY/PROPAGATE LINES

C_HORIZ
C_ADDF
CADD_S
C_MULT
C_CX_I
C_SEL_C
C_SELX

OUT1 OUT2 COMP_OUT
CADD_A
COUTX
CINXD
MULTO2
CINY1 C_Y12
CINY2
C_CY1_I
CADD_S

C_ADDF
C_HORIZ
C_SEL_C
C_SELY1
CINXD
CADD_A
CINY1 COUTY1

C_SEL_C
C_SELY2
COY2_S
C_MULT
C_CY2_I
CINY2 COUTY2
CINXD

C_HORIZ
C_COMP
COMP_OUT
C_MULT
C_PX_I
C_SEL_P
C_SELX
PINY1 C_Y12
PINY2
C_COMP_I
MULTO1
PINXD POUTX

C_COMP
C_HORIZ
C_SEL_P
C_SELY1
COMP_OUT
C_COMP_I
PINY1 POUTY1
PINXD

C_SEL_P
C_SELY2
C_MULT
C_PY2_I
PINY2 POUTY2
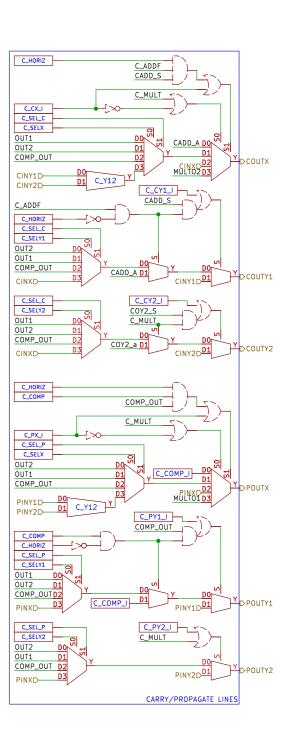PINXD

This diagram is intended to be functionally correct, but when given a choice between ease of understanding
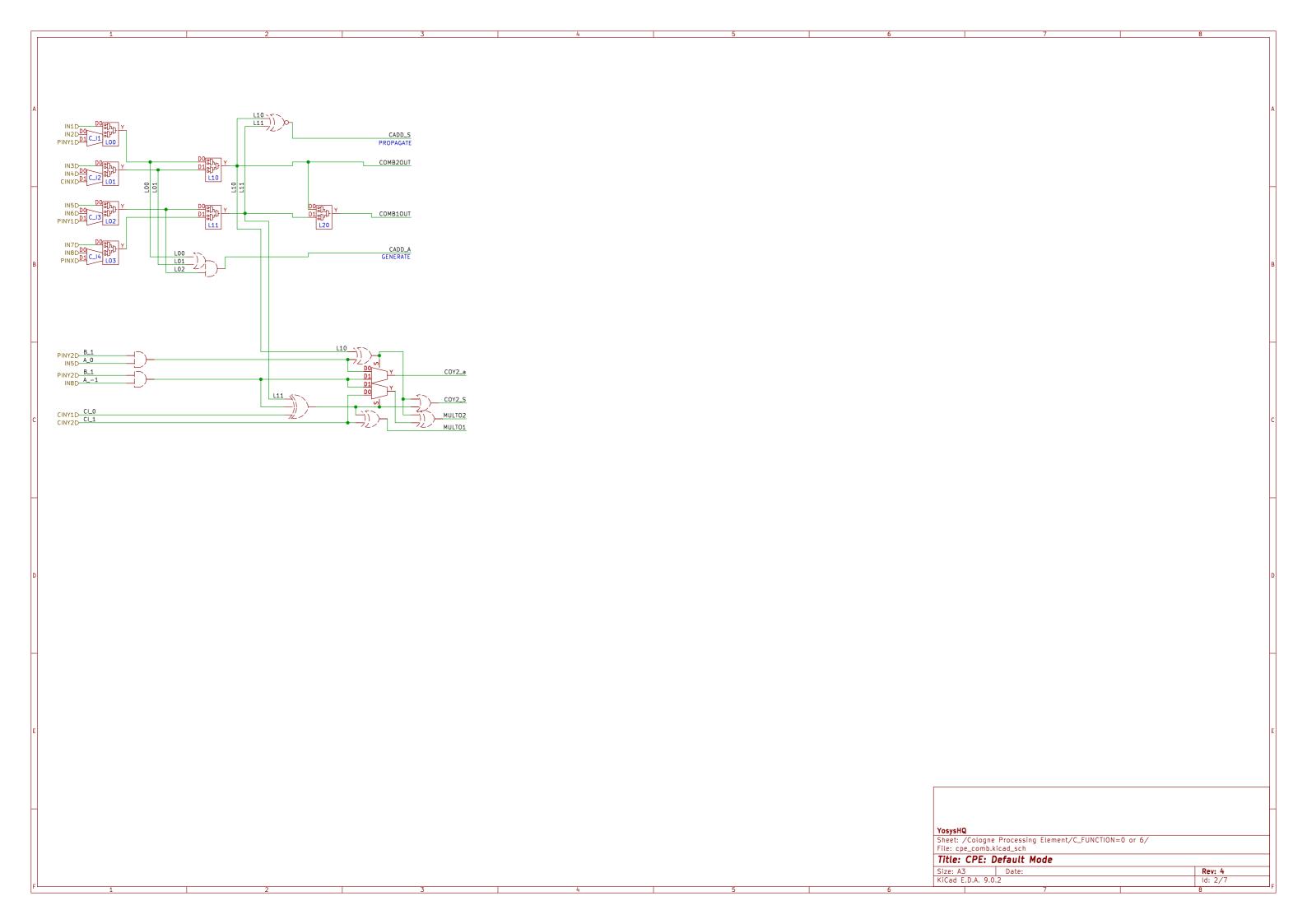and physical accuracy, I always chose the former.

Dashed symbols represent where I'm describing something physically inaccurate.
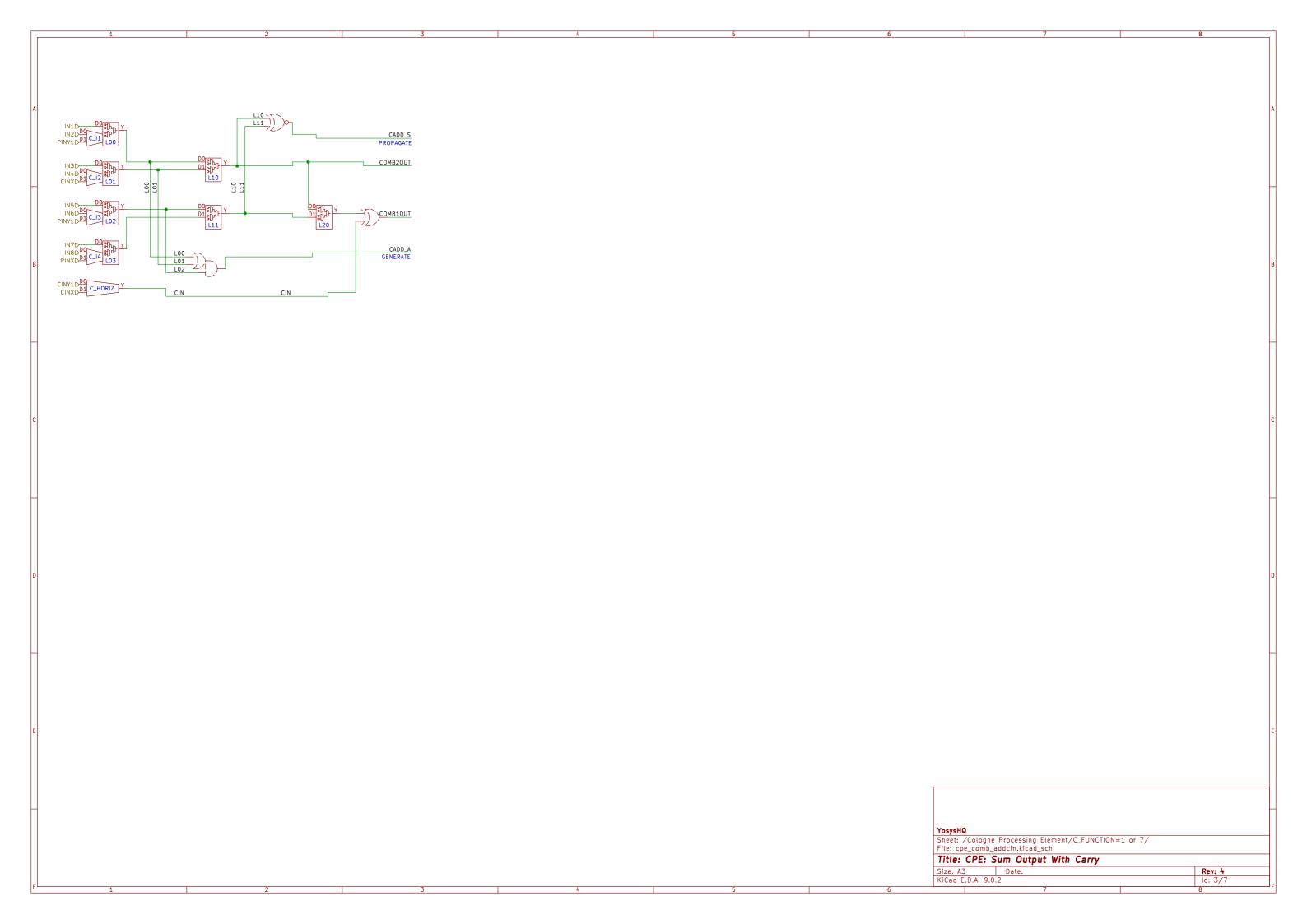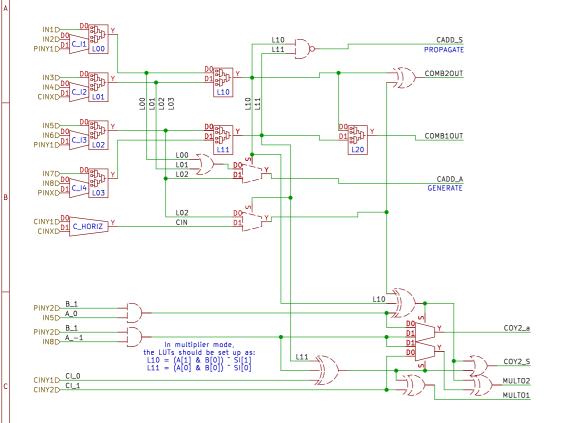— Hannah Ravensloft

IN1 — D0
IN2 — D0
PINY1 — D1 C_I1 Y   L00

IN3 — D0
IN4 — D0
CINX — D1 C_I2 Y   L01

IN5 — D0
IN6 — D0
PINY1 — D1 C_I3 Y   L02

IN7 — D0
IN8 — D0
PINX — D1 C_I4 Y   L03

L10
L11

CADD_S
PROPAGATE

D0
D1 Y   L10

COMB2OUT

D0
D1 Y   L11

D0
D1 Y   L20

COMB1OUT

L00
L01
L02

CADD_A
GENERATE

L10
L11

PINY2 — B_1
IN5 — A_0

PINY2 — B_1
IN8 — A_-1

L10
S

D0
D1 Y   COY2_a

D1
D0 Y   COY2_S

S

CINY1 — CI_0
CINY2 — CI_1

L11

MULTO2

MULTO1

CADD_S
PROPAGATE

COMB2OUT

COMB1OUT

CADD_A
GENERATE

CIN

CIN

IN1 D0
IN2 D0
PINY1 D1 C_I1 L00 Y

IN3 D0
IN4 D0
CINX D1 C_I2 L01 Y

IN5 D0
IN6 D0
PINY1 D1 C_I3 L02 Y

IN7 D0
IN8 D0
PINX D1 C_I4 L03 Y

CINY1 D0
CINX D1 C_HORIZ Y

D0 D1 L10 Y

D0 D1 L11 Y

D0 D1 L20 Y

L00
L01

L10
L11

L10
L11

L00
L01
L02

IN1
IN2
PINY1

IN3
IN4
CINX

IN5
IN6
PINY1

IN7
IN8
PINX

CINY1
CINX

D0
D0
D1
C_I1
L00

D0
D0
D1
C_I2
L01

D0
D0
D1
C_I3
L02

D0
D0
D1
C_I4
L03

D0
D1
C_HORIZ

Y

Y

Y

Y

Y

L00
L01
L02
L03

L00
L01
L02

L02
CIN

D0
D1
L10

D0
D1
L11

D0
D1

D0
D1

Y

Y

Y

Y

L10
L11

L10
L11

D0
D1
L20

Y

Y

CADD_S
PROPAGATE

COMB2OUT

COMB1OUT

CADD_A
GENERATE

PINY2
IN5

PINY2
IN8

B_1
A_0

B_1
A_-1

L10

In multiplier mode,
the LUTs should be set up as:
L10 = (A[1] & B[0]) ^ SI[1]
L11 = (A[0] & B[0]) ^ SI[0]

L11

CINY1
CINY2

CI_0
CI_1

D0
D1
D1
D0

Y

Y

COY2_a

COY2_S

MULTO2

MULTO1

C_FUNCTION = 2: ADDF2

This provides a 2-bit full adder.

The addition is performed in propagate/generate style.

Setting C_HORIZ configures the adder to propagate carries
horizontally, rather than vertically. This feature isn't
implemented in p_r or nextpnr because horizontal carries
have twice the propagation delay of vertical carries.

C_FUNCTION = 3: MULT

This provides a 2x2-bit Booth multiplier.

The dedicated logic performs multiplication by B[1], while
multiplication by B[0] is performed within the soft logic.

(These functions differ by how they set up the downstream
carry and propagation lines; from this part of the logic
they function identically, thus have been grouped together)

C_FUNCTION = 4: MUX4

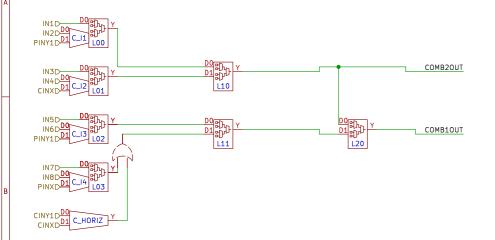This provides a four-input multiplexer.

Input inversion is performed by setting the relevant config bit of the C input, while constant inputs are masked by clearing the relevant config bit of the X input.

C_FUNCTION = 5: EN_CIN

This is one of the more obscure CPE configurations.

The intended usecase of this is to source a signal
from the carry lines to reduce routing delay.

COMB2OUT

COMB1OUT

IN1
IN2
PINY1
C_I1
L00

IN3
IN4
CINX
C_I2
L01

IN5
IN6
PINY1
C_I3
L02

IN7
IN8
PINX
C_I4
L03

CINY1
CINX
C_HORIZ

L10

L11

L20